

Informed Urban Transport Systems

Classic and Emerging Mobility
Methods toward Smart Cities



JOSEPH Y. J. CHOW

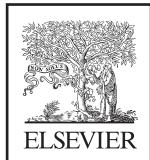
Foreword by Will Recker

INFORMED URBAN TRANSPORT SYSTEMS

INFORMED URBAN TRANSPORT SYSTEMS

**Classic and Emerging
Mobility Methods
toward Smart Cities**

JOSEPH Y.J. CHOW



Elsevier
Radarweg 29, PO Box 211, 1000 AE Amsterdam, Netherlands
The Boulevard, Langford Lane, Kidlington, Oxford OX5 1GB, United Kingdom
50 Hampshire Street, 5th Floor, Cambridge, MA 02139, United States

© 2018 Elsevier Inc. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or any information storage and retrieval system, without permission in writing from the publisher. Details on how to seek permission, further information about the Publisher's permissions policies and our arrangements with organizations such as the Copyright Clearance Center and the Copyright Licensing Agency, can be found at our website: www.elsevier.com/permissions.

This book and the individual contributions contained in it are protected under copyright by the Publisher (other than as may be noted herein).

Notices

Knowledge and best practice in this field are constantly changing. As new research and experience broaden our understanding, changes in research methods, professional practices, or medical treatment may become necessary.

Practitioners and researchers must always rely on their own experience and knowledge in evaluating and using any information, methods, compounds, or experiments described herein. In using such information or methods they should be mindful of their own safety and the safety of others, including parties for whom they have a professional responsibility.

To the fullest extent of the law, neither the Publisher nor the authors, contributors, or editors, assume any liability for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions, or ideas contained in the material herein.

Library of Congress Cataloging-in-Publication Data

A catalog record for this book is available from the Library of Congress

British Library Cataloguing-in-Publication Data

A catalogue record for this book is available from the British Library

ISBN: 978-0-12-813613-3

For information on all Elsevier publications visit our website at <https://www.elsevier.com/books-and-journals>



Working together
to grow libraries in
developing countries

www.elsevier.com • www.bookaid.org

Publisher: Joe Hayton

Acquisition Editor: Tom Stover

Editorial Project Manager: Thomas Van der Ploeg

Production Project Manager: Anusha Sambamoorthy

Cover Designer: Victoria Pearson

Typeset by SPi Global, India

LIST OF FIGURES

Fig. 1.1	Overview of Canadian ITS Architecture version 2.0	6
Fig. 1.2	Demonstration of use of MATSim to visualize public transport boardings and alightings in Singapore	7
Fig. 1.3	Comparison of personal computer, smartphone, and tablet sales	8
Fig. 1.4	A vision of Mobility-as-a-Service	9
Fig. 1.5	Distribution of BIXI bike share in Toronto in 2012	14
Fig. 1.6	Manheim-Florian-Gaudry framework for transport systems analysis, with new elements in <i>gray</i>	19
Fig. 1.7	MATSim functionality	26
Fig. 2.1	Demonstrations of data opportunities due to advances in Big Data and smart cities: (A) using Twitter data to track urban mobility during Hurricane Sandy; (B) real-time California freeway traffic monitoring from Caltrans PEMS; (C) repository of real-time and scheduled GTFS feeds from around the world including Paris; (D) call detail records used to model travel patterns; (E) Taxi GPS data	32
Fig. 2.2	A typology of smart city functions	33
Fig. 2.3	Columbus smart city challenge implementation vision	35
Fig. 2.4	Illustration of connecting IoT infrastructure with cities, users, operators, and software developers	36
Fig. 2.5	Illustration of beats that form pulses unique to different locations in a city depending on the type of activity data used	39
Fig. 2.6	Using a (A) 4-region representation of NYC and (B) illustration of a standardized pace vector progressing over time before, during, and after Hurricane Sandy	41
Fig. 2.7	(A) Original illustration of activity prisms and (B) demonstration of space-time aquarium of space-time paths of African and Asian Americans in Portland	44
Fig. 2.8	(A) Minimum paths and (B) velocity field isochrones for a point 2.5 miles east of Manchester city center in 1965	45
Fig. 2.9	Constructing a least-cost path from a cost vector field. (A) Initialization, (B) least cost path with accumulated cost to j , (C) temporally referenced least cost path, and (D) extracted path as a space-time polyline	47
Fig. 2.10	Illustration of a field-based space-time prism from Miller and Bridwell (2009)	48
Fig. 2.11	Illustration of two different individual velocity vectors	48
Fig. 2.12	(A) Household travel survey data from Greater Toronto Area; (B) kernel density map of the trajectories at 8 a.m.; (C) travel momentum field at 8 a.m.; (D) isometric views of travel momentum field in (C)	51
Fig. 2.13	Example for Exercise 2.2	53

Fig. 2.14	(left) Beijing, China, and (right) total daily real-time GPS locations from 12,000 taxis on November 2, 2012	54
Fig. 2.15	Snapshots of kernel density estimated taxi population TMF in 24 h	55
Fig. 2.16	Illustration of vector projection	56
Fig. 2.17	TMFs for Exercise 2.3	57
Fig. 2.18	Projected values and temporal profile	58
Fig. 2.19	Bus route no. 506 from Toronto Transit Commission	59
Fig. 2.20	Temporal profile of route projections of TMF on TTC bus no. 506	59
Fig. 2.21	(A) TMFs for Exercise 2.4 and (B) project values along with temporal profile	60
Fig. 2.22	Magnitude of travel impulse field between 2011 and 2006 TMFs at 8 a.m	61
Fig. 3.1	Supply-demand curves for a transport system	70
Fig. 3.2	Equilibrium allocation of travelers on two parallel routes	72
Fig. 3.3	Network for Exercise 3.3	74
Fig. 3.4	Before and after scenarios of network for Braess' Paradox example	85
Fig. 3.5	Illustration of freight assignment of (A) path-based commodity flows, and simultaneously, (B) link-based cyclic truck and rail vehicular flows	94
Fig. 3.6	Sioux Falls network with link IDs and link flows (ID/flow) under UE	96
Fig. 3.7	Illustration of network assignment in Greater Toronto Area using TransCAD	98
Fig. 3.8	Illustration of transit lines operating over a 4-node network	99
Fig. 3.9	Sample network to illustrate common lines problem	100
Fig. 3.10	Sample network for Exercise 3.9	102
Fig. 3.11	Sample network with updated link cost functions for Exercise 3.10	106
Fig. 3.12	Illustration of time-expanded network representation of a transit network with three lines	108
Fig. 3.13	Equilibrium link flows for each of three passenger groups	109
Fig. 3.14	MILATRAS model framework	110
Fig. 3.15	Queue represented by cumulative departures from home and arrivals at work	113
Fig. 3.16	Bottleneck queue with optimal toll applied across optimal time interval	115
Fig. 3.17	Plot of queues formed from example in Exercise 3.11	116
Fig. 3.18	(A) Assumed downtown city block setting and (B) dynamic flows under saturated condition	118
Fig. 3.19	Study area in downtown Toronto	121
Fig. 3.20	Dynamics of search friction	124
Fig. 3.21	Algorithms to (A) run and (B) calibrate the taxi equilibrium model	125
Fig. 3.22	Relative spatial changes in consumer surplus from single-ride taxi market to shared taxi via (A) matching by waiting and (B) matching by detour	127

Fig. 3.23	Smart cities mobility provision represented as a two-sided market	129
Fig. 3.24	Illustration of the sensitivity of two-sided market to platform pricing	130
Fig. 3.25	Components of agent-based stochastic user equilibrium obtained by day-to-day adjustment of user and operator decisions as a two-sided market	134
Fig. 4.1	A (A) 4-node network on which (B) a user conducts activities	140
Fig. 4.2	Additional trip generated without any additional economic activity (Kang et al., 2013)	141
Fig. 4.3	Disutility of schedule delay depending on the flexibility of late arrival	143
Fig. 4.4	Activity system to illustrate flexibility of model to capture different scheduling preferences	153
Fig. 4.5	Three solutions of mHAPP based on different parameters for user 1 (A, B) and user 2 (C)	155
Fig. 4.6	Solutions of example in Chow (2014) for a single household using Algorithm 4.2, for (A) a baseline scenario and (B) a scenario where discretionary activity utility is doubled	161
Fig. 4.7	Overview of decomposition approach to obtaining market schedule equilibrium	163
Fig. 4.8	Cumulative diagram of arrivals and departures onto link (u, w) for Exercise 4.4	165
Fig. 4.9	Activity and transport system for Exercise 4.5 (driving: $\{(0, 7), (0, 6), (6, 7)\}$, transit: $(8, 9)$, walking: $\{(0, 2), (0, 8), (2, 8), (2, 7), (1, 6), (1, 9)\}$)	168
Fig. 4.10	Algorithm 4.3 convergence for Exercise 4.5	169
Fig. 4.11	Cumulative diagram (A) without link capacity and (B) with link capacity	170
Fig. 4.12	MATSim input data: (A) transit line schedule from GTFS overlaid on a road network from OpenStreetMaps and (B) activity zones corresponding to NYMTC household travel survey	172
Fig. 4.13	Schedule file that needs to be modified for scenario analysis	173
Fig. 4.14	Comparison of distribution of welfare measure for users of the system before and after headway reduction	174
Fig. 4.15	Comparison of distribution of 7 a.m.–10 p.m. ridership on the A line with Regular (before) and Reduced (after) headway	174
Fig. 4.16	Comparison of distribution of time of day arrivals onto the A line at the W. 4th Station (Station 13) before and after headway reduction	175
Fig. 4.17	Freight activity system	179
Fig. 5.1	Illustration of a state change in the Nguyen-Dupuis network that leads to a change in observed flows	186
Fig. 5.2	Bayesian network model of Luxembourg commuter mode choice	191
Fig. 5.3	Sample network for Exercise 5.1	196

Fig. 5.4	Example dial-a-ride problem for Exercise 5.2	200
Fig. 5.5	OCTAM Network and activity locations of households in Orange County, CA, from the 2001 California Household Travel Survey	203
Fig. 5.6	Comparison of HAPP outputs based on different parameters	206
Fig. 5.7	Sample GPS data from drayage trucks out of the San Pedro Bay Ports	207
Fig. 5.8	Comparison of inverse VRP results for individual and fleet of two vehicles	208
Fig. 5.9	Sample network for Exercise 5.3	209
Fig. 5.10	Thirteen major airports in California	211
Fig. 5.11	Multiagent inverse optimization as a fixed-point problem	214
Fig. 5.12	Illustration of Algorithm 5.2	215
Fig. 5.13	Test network with node and link IDs	219
Fig. 5.14	Convergence of Algorithm 5.2 on test network	220
Fig. 5.15	Output distribution of posterior link costs across the population of 500 simulated agents for link 1 (top) to link 5 (bottom)	221
Fig. 5.16	Toy network used for illustrating methodology	227
Fig. 5.17	Nguyen-Dupuis network	228
Fig. 5.18	Convergence of dual prices using Algorithm 5.2A	230
Fig. 5.19	Trajectory of simulated route observations in an online learning setting	231
Fig. 5.20	Dual price trajectories based on 300 simulated agent arrivals	231
Fig. 5.21	Queens freeway network	232
Fig. 5.22	Link dual price trajectories obtained from online network learning	233
Fig. 5.23	Half-hour interval screenshots of Google Maps real-time shortest path queries (Xu et al., 2017)	234
Fig. 5.24	Snapshots of multiagent IO output dual prices at every half hour with nonzero prices represented by <i>heavier arrows</i>	235
Fig. 6.1	Laplace distributed noise added to functions of values from S and S'	242
Fig. 6.2	Illustration of filtered income data according to $\epsilon = \{1000, 100\}$	246
Fig. 6.3	Spatial distribution and travel times between zone centroids, and 20 sample OD locations	249
Fig. 6.4	Distributions of location noise by origin zone for $\epsilon = 5$	249
Fig. 6.5	Comparison of OD distributions for original and ϵ -differentially private locations	251
Fig. 6.6	Framework for k -anonymous diffusion models	253
Fig. 6.7	(A) Original flows, (B) outcome assignment to minimize repeating the original flows, and (C) the outcome assignment to minimize repeating the combined flows on the left	254
Fig. 6.8	Shift of probabilities from maximum probability for origin flows to unconstrained maximum entropy	258
Fig. 6.9	(A) Example tour, with (B) true vehicle arrival times and passenger travel times, and (C) other known parameters	262
Fig. 6.10	Diffusion of $(0, 2, 1, 3, 5, 4, 6, 0)$ into 90 tours with $\Delta = 0.1$	263
Fig. 6.11	Ten-anonymous diffusion of $(0, 2, 1, 3, 5, 4, 6, 0)$ using Algorithm 6.2 with $\Delta = 0.10$	264

Fig. 6.12	Comparison of entropy value for number of tours under different tour generation methods	264
Fig. 6.13	Two-path diffused data object set	267
Fig. 6.14	Two-anonymous distribution of posterior link 7 dual price as a function of prior	267
Fig. 7.1	Types of problems by complexity	276
Fig. 7.2	Network used for Exercise 7.1	279
Fig. 7.3	Iterations of Prim's Algorithm for Exercise 7.1	280
Fig. 7.4	Illustration of the Steiner Tree Problem	281
Fig. 7.5	Seven bridges of Königsberg	282
Fig. 7.6	Illustration of worst-case matched odd-degree node lengths	284
Fig. 7.7	Optimal solution to TSP using integer programming	284
Fig. 7.8	Eulerian tour obtained using Algorithm 7.2	285
Fig. 7.9	Several iterations of an nearest neighbor heuristic on example from Fig. 7.2	286
Fig. 7.10	Instance for Exercise 7.3	290
Fig. 7.11	Optimal solution obtained by integer programming	291
Fig. 7.12	Optimal solution to DARP example	293
Fig. 7.13	Instance for Exercise 7.5	297
Fig. 7.14	Optimal solution via integer programming for two P values	298
Fig. 7.15	Optimal solutions to set covering problem with $s=1, s=2$	301
Fig. 7.16	Facility location based on (A) nodes, (B) flow, and (C) itinerary intercept	305
Fig. 7.17	Solutions to (A) relocation ignoring queue delay and (B) relocation with queue delay	308
Fig. 7.18	Illustration of line planning problem	309
Fig. 7.19	Input data for Exercise 7.11	314
Fig. 7.20	Illustration of a bilevel problem in Eq. (7.20) with no solution	318
Fig. 7.21	Four steps to surrogate-based NDP: (A) generate sample solutions, (B) fit a surrogate model to the samples, (C) ensure feasibility of sample solutions, (D) select new solution using surrogate model	319
Fig. 7.22	Network used for Exercise 7.12	320
Fig. 7.23	Two layers of Algorithm 7.6 on Exercise 7.13	321
Fig. 7.24	Example for Exercise 7.15	328
Fig. 7.25	Lower $(0,0,0)$ and upper bound $(1,1,1)$ solutions at the root node	329
Fig. 7.26	(A) Regions for classifying symbiotic network design strategies and (B) impact of network expansion and contraction	333
Fig. 7.27	Example network for Exercise 7.16	334
Fig. 7.28	Objective space of the alternative solutions	334
Fig. 7.29	Example with (A) substitutable toll roads and (B) complementary toll roads	336
Fig. 8.1	Examples of stochastic variables: (A) a probability density function of a random variable with a triangular distribution and (B) corresponding cumulative distribution function of triangular distribution	343

Fig. 8.2	Examples of stochastic processes: (A) first and (B) second instances of a simulation ($t=[0, 5]$) and projection ($t=(5, 6]$) of a geometric Brownian motion with $\mu=0.1$, $\sigma=0.4$	350
Fig. 8.3	Expanding the NPV decision space (top region) to four other regions as opportunities	353
Fig. 8.4	Distributions of profit under commitment (<i>solid</i>) and option (<i>dashed</i>) investments	354
Fig. 8.5	Plot of project value R and the corresponding option value V for two projects	357
Fig. 8.6	(A) Sioux Falls network with candidate links and (B) comparison of deferral with redesign (Network Investment Deferral Option (NIDO)) to deferral only as function of σ	364
Fig. 8.7	Illustration of market switching	365
Fig. 8.8	Comparison of two trajectories (Q1, Q2) from an initial point $Q_0=\$0.75$	368
Fig. 8.9	(A) Single OD, three-link example with (B) a plot of the base link cost functions and (C) simulated trajectories of Q over a 2-year horizon	375
Fig. 8.10	Comparison of the CR-Weibull distributions for City A (F_A) and City B (F_B)	382
Fig. 8.11	Simulation of Hyttiä et al. (2012) dynamic dispatch policy	383
Fig. 8.12	Evaluation of myopic and nonmyopic policies relative to CR-Weibull distributions. (A) Scenario A, (B) Scenario B, and (C) Scenario C	384
Fig. B.1	Systems engineering process	416
Fig. B.2	Use case diagrams for (A) on-bus fare ticketing and (B) roadside fare ticketing	419
Fig. B.3	A conceptual diagram for the roadside bus ticketing system	420
Fig. B.4	An activity diagram with swimlanes for Pay Fare use case in roadside ticketing	421
Fig. B.5	A deployment diagram of the PT04 Transit Fare Collection Management use case	422
Fig. C.1	Example state transition diagram	427
Fig. C.2	State transition diagram of a birth-and-death process, of which the M/M/1 model is one	430
Fig. C.3	Comparison of the state probabilities for M/M/1 and M/M/1/1 queues	434
Fig. C.4	Region for a spatial queue analysis with a single shuttle based at a central depot	435
Fig. C.5	Example of a queue network	437
Fig. D.1	Illustration of isouility curves and resource expenditure budget (straight line)	440
Fig. D.2	Derivation of individual demand function from utility curves	441
Fig. D.3	Comparison of CDFs of ε_n	444
Fig. D.4	Nested logit example	448
Fig. D.5	Illustration of nonconvexity of revenue maximization on two segments with $p_2=\$2$	454

LIST OF TABLES

Table 1.1	Elements of a transport system	11
Table 1.2	Additional elements for an <i>urban</i> transport system	13
Table 3.1	Costs of coffee shop queue	71
Table 3.2	Path-link incidence matrix for Exercise 3.4	79
Table 3.3	Test networks	97
Table 3.4	Commercial software and open source tools for traffic assignment	97
Table 3.5	Comparison of equilibrium and social optimal scenarios for downtown Toronto	122
Table 3.6	Different types of mobility as a service	128
Table 4.1	Modes used in scheduling portion of CEMDAP	145
Table 4.2	Cumulative arrivals and departures on link (u,w) for Exercise 4.4	165
Table 4.3	Activity data for Exercise 4.5	168
Table 4.4	mHAPP network data for Exercise 4.5	168
Table 4.5	Market schedule equilibrium assignment	170
Table 4.6	Stations on the A-Lefferts Blvd line	173
Table 5.1	Machine learning methods by similarity compiled by Brownlee (2013)	188
Table 5.2	Machine learning applications in urban transport	190
Table 5.3	Inverse optimization advances and applications	193
Table 5.4	Median arrival times by activity type from 2001 California Household Travel Survey	204
Table 5.5	Prior and calibrated capacity parameters for freight flows in the airports in California in 2007	212
Table 5.6	Summary of model calibration and validation	212
Table 5.7	Illustration of parameter estimation using Algorithm 5.2 for example from Section 4.5	217
Table 5.8	Scenarios evaluated in example	218
Table 5.9	Estimated parameters and significance tests for multinomial and mixed multinomial logit model	221
Table 5.10	Estimated shares (MNL) vs actual shares of route choices when link 3 is closed	222
Table 5.11	Comparison of uninformed prior vs optimal invariant common prior	223
Table 5.12	Comparison of performance measures	224
Table 6.1	Illustration of the differential privacy criterion evaluation	243
Table 6.2	Sample of 20 individual incomes to illustrate effect of differential privacy on consumer surplus	245
Table 6.3	Summary of 10 simulated queries of income data with differential privacy for $\epsilon = \{1000, 100\}$	247
Table 6.4	Simulated origins and destinations	250
Table 6.5	Travel times and demand	257

Table 6.6	Travel times for Exercise 6.5	262
Table 7.1	Network design problems covered in this chapter	276
Table 7.2	Popular software packages for solving network design problems	277
Table 7.3	Sorted savings for all node pairs in descending order	291
Table 7.4	Solutions to MCLP for Exercise 7.7	303
Table 7.5	Server locations at time t and $t+1$ (without and with relocation costs)	304
Table 7.6	Solution to Exercise 7.11	314
Table 7.7	Example equilibrium solutions	337
Table 8.1	Summary of solution for Exercise 8.2	348
Table 8.2	Twelve “in-the-money” simulated path values for $t=\{9, 10\}$	362
Table 8.3	Summary of sensitivity analysis for Exercise 8.6	368
Table 8.4	Summary of fixed-flexible switching example for Fig. 8.7 in Guo et al. (2017)	370
Table 8.5	Simulation of 20 sample paths	376
Table 8.6	Exercise values for each option in sequence	377
Table 8.7	Exercise decisions	378
Table 8.8	Summary of exercise, deferral, option values, and decisions for each sequence in whole	379
Table 8.9	Summary of CR policy decisions as a function of σ	380
Table A.1	Transportation research h-index rankings by institution for (left) all years up to 2016, and (right) for papers published in 2007–16	414
Table B.1	UML diagrams (Ambler, 2017)	418
Table B.2	Selection of ITS performance measures	423
Table C.1	Example transition matrix	427
Table D.1	Input data for assortment problem example	455

FOREWORD

I love “aha” moments—sudden, sharp moments that bring clarity to otherwise obscured subjects. I have been fortunate enough to have had a few such moments spread over a, perhaps, too-long career; many other would-be aha moments fizzled and died on the vine. This book is filled with aha moments—moments that bring unexpected clarity to the underlying issues that define so-called smart cities and the complex networks that will make them possible. Oft said that “everything is connected to everything”—in the sense of the butterfly effect—but we now enter into a data-rich, information-dominated era where “everything *can be* connected to everything” in the sense of real-time system management. Joe has captured the imagination of the possibilities portended by these newfound capabilities and given a box of tools to build and choreograph the dance of an interconnected urban society characterized by physical, economic, and social mobility.

Okay, so by now we have all heard about Big Data and how it is going to revolutionize our approaches to understanding how urban systems work, in general, and how transport networks behave, in particular. But, other than recognizing the possibility of drowning in the vast amounts of disparate data that might offer clues to understanding, or even obscure them, there has been only scant progress in identifying approaches to harness the wealth of information provided by new technologies into meaningful analysis tools. Although the era of Big Data and such new concepts as autonomous/connected vehicles and shared use/ownership are bound to give rise to unimagined mobility systems and their attendant modeling capabilities that heretofore simply were impossible to achieve, it does not mean that the fundamental properties of urban systems derived from basic economic principles need to be abandoned. So, it is pretty fitting that Joe starts this journey by first couching its destination within the context of the classical Manheim-Florian-Gaudry (MFG) framework that characterizes urban systems as an interaction between activity and transport systems. The prominent role of activities as the driving force of the need for transport, and the networks that support it, in this paradigm serves its extension to the design of informed urban transport systems nicely throughout the remaining chapters.

In [Chapter 2](#), Joe systematically identifies the components and characteristics that position smart cities as a time-geographic subset of the internet of things. But, then comes an aha moment—what if we treated these real-time “big mobility data” as comprising travel momentum vector fields. Transport analysis has a long history of borrowing ideas from other fields—gravity model of trip distribution, fluid dynamics of traffic flow, assay analysis as the roots of travel choice behavior—as analogies to transport concepts that stem from no physical laws. But, for the most part, these analogies have been addressed to explaining certain observable outcomes of mobility decisions, for example, traffic flows on networks, travel speeds, transit patronage, and not as vehicles to expose and define as yet unobserved phenomena. With the introduction of the vector field interpretation of activity data, Joe has opened the door to a completely new foundation for analyzing mobility patterns that form the heartbeat of urban existence—one that promises to lead to models that uncover new concepts built on Big Data, rather than simply using Big Data to address conventional aspects of mobility. Maybe it is just my old training in applied mechanics showing through, but I find this train of thought pretty exciting stuff.

[Chapter 3](#) begins the “heavy lifting” in developing smart city design and operation capabilities, and is not for the faint hearted, as the mathematical formulations underpinning classical network problems require some due diligence (as does most of the content of the remaining chapters). But, there is a reward at the end of this necessary positioning in the form of the exposition of a wealth of approaches to the general MaaS (Mobility as a Service) systems that will characterize smart cities of the future—a future in which public agencies are the platform linking mobility providers and travelers, connecting everything to everything.

During a period when I was fortunate to have Joe as a post doc, he became a bit intrigued by an activity scheduling model (HAPP) that I had developed some years previously, and we played around with some extensions to the model that left me pretty satisfied that some incremental research had been accomplished, but which left Joe with a vision (which I, admittedly, lacked) of the potential of the model as a framework for general application in urban systems analysis. I remember jokingly sending Joe off with a charge to take the model and apply it in ways that would make me famous as its originator. Well, with his mHAPP, which is largely the focal point of [Chapter 4](#), just maybe he will. When I first proposed HAPP, what has now become known as activity-based modeling was at its infancy. It struck me that, as in the words of Hawthorne, we were “looking at

the wrong side of the tapestry” in our focus on modeling travel as trips, rather than as their being just the threads that weave, as Allan Pred called it, the activities that define the “choreography of human existence.” I developed the framework for the original HAPP as a mathematical programming model, in the hope that “real” or researchers would use it as a kernel for all sorts of urban mobility problems. So, some 20 years later, Joe has managed to do just that with his generalizations of the approach, as applied to the complex interactions of the various actors in mobility systems that are fundamental to smart cities. This is must reading for anyone who buys into mathematical modeling approaches to activity systems.

As theoretically satisfying mHAPP and its derivatives are as mathematical representations of complex systems, they certainly are not without their challenges. Not only do they generally apply only to the individual actor, they also possess a myriad of parameters that, while easily definable, are ridiculously hard to come by in an urban setting. In the original work that Joe and I did in extending HAPP, the parameters were the vaguely defined time windows within which people had to complete activities comprising a set agenda and the respective value or utility that they derive from participating in an activity depending on where it is positioned within the time window. Now, you might think that uncovering these values is just another calibration/estimation problem—like maximum likelihood estimation in discrete choice theory—but it is not. Not only is the choice set unbounded, but more importantly it is constrained by a hard set of relationships that cannot be violated. Joe cleverly addresses this conundrum using what is known as inverse optimization. The basic idea here is, absent definitive values for the parameters that define the constraints and objective in a mathematical programming application, reformulate the problem as a dual formulation in which the objective is to pick the parameter values that best match the real-world observations, subject to minimizing the difference between prior- and posterior-probability distributions—sort of reverse engineering. In [Chapter 5](#), he expands on and generalizes this approach in applications to a wide variety of urban network examples. These examples tie in neatly to newfound opportunities for real-time management afforded by Big Data and machine learning. And, with extensions of the approach to multiagent systems, we can get a glimpse into the tapestry woven by many travelers on networks operated by public and private agencies—bringing us about as close to what smart cities are all about as we can imagine at this point in time.

In [Chapter 6](#), Joe tackles the thorny problem of data privacy. The inverse optimization models described in [Chapter 5](#) are “designed” to expose

unobserved operating policies and behavioral preferences not captured by foundational, *a priori* assumptions of the modeling formulations. In the former case, this has the potential to reduce or even eliminate competitive advantages that an operator may have over its rivals; in the latter case, it can be used to reveal the identity, and personal information, of travelers comprising the data used to estimate model parameters. The sharing of travel-related information pertaining to individual travelers and both public and private operators in a dynamic MaaS environment is an essential requirement for the development of smart cities. The challenge is to preserve the integrity of the fundamental properties of the information needed from the shared pool of data while protecting the anonymity of the providers of the information and the specific policies of the operators of the system. Through a series of illustrative examples and exercises, Joe walks through a number of algorithms for applications of the notion of “differential privacy” guarantees to individuals providing real-time location data, and k -anonymous diffusion models to disguise guarded policies underlying operator-provided system management, ending with a tie to network learning systems introduced in [Chapter 5](#)—all setting the stage for his exposition in the succeeding chapters of the design of networked systems envisioned for smart cities.

All of which brings us to what Joe calls the “meat” of the book—network design in the era of smart cities. To be sure, network design problems are not exactly new kids on the block; they have been studied by many over the past decades and there have been numerous formulations of, and proposed solutions to, what are principally variations of vehicle routing problems (themselves being variations of the famous traveling salesman problem). What is different here is that, with few exceptions, previous study in this area has been focused on the operational characteristics of distinctly separate networks, for example, network of roadways, series of transit routes, taxi/dial-a-ride dispatch, freight delivery systems, communications systems—with ascribable user demand. If smart cities are to fulfill their promise, all of these systems must be capable of interacting, in real time, with each other, and with a population of travelers and goods that flow seamlessly across the boundaries of disparate networks to achieve a common goal—a dynamic network of networks, so to speak. So, in the final section of the book, aptly labeled “Design of Informed Systems,” we return once again to the notion that “everything is connected to everything” within the context of smart cities and the networked mobility service systems that will serve as their lifeblood. After first providing an encyclopedia of ready-to-go algorithms for addressing classical static network design problems and their

extensions in [Chapter 7](#), Joe circles back to the ties between these methods and the decision-making opportunities proffered by the wealth of real-time data—shared by system operators and users—on the horizon. Through a series of examples and exercises Joe proposes, in [Chapter 8](#), how the network design decisions of the previous chapter can be adapted to a dynamic data-rich, data-sharing environment—albeit one wrought with uncertainties—to achieve operational efficiencies beyond their static, uninformed, and unconnected counterparts.

Smart cities are not here yet, but they will be. The book is not closed on how we can best analyze and design the information-based mobility systems that will be the messy side of the tapestry of urban interactions. Rather, in this treatise, Joe opens the book on this subject with an excellent blueprint for the application of a compendium of modeling techniques and understanding that can serve as a foundation for design of first-generation smart city systems. How exciting is that! Kinda makes me wish that my research career was closer to its beginning stage than to its end.



Will Recker
University of California, Irvine

ACKNOWLEDGMENTS

A work of this magnitude could not have been completed without help and support from many individuals. I am thankful to Elsevier (and Tom Stover who first approached me about this project) for the trust and support in bringing this project to life. As an Assistant Professor at NYU, I had initial doubts about my capacity in successfully producing a monograph that would make a meaningful impact on the field. Most books come from much more established researchers in their prime. Now at its completion, however, I feel much more confident about this and am grateful for the words of encouragement earlier on in this process from my department chair, Magued Iskander, and the director of C2SMART University Transportation Center, Kaan Ozbay.

I initiated the proposal for this book in 2016 and started working on it at the beginning of 2017. Funding support from the National Science Foundation through the CAREER grant (CMMI-1652735) helped support this work, so I am grateful to the institution and to the program managers Yanfeng Ouyang (for his role in awarding me the grant) and Cynthia Chen (for her role in managing the awarded grant).

The material in this book is a combination of lecture notes that I compiled over the years from the literature as well as more recent advances and innovations developed out of my lab. The portions of [Chapter 2](#) dealing with travel momentum fields were largely based on my earlier work with my former postdoc, Xintao Liu, who is now an Assistant Professor at Hong Kong Polytechnic University.

[Chapter 3](#) covers research in two-sided markets and evaluation of flexible transport services that I developed with Shadi Djavadian, now a postdoc at Ryerson University. They were supported by research grants from the Canada Research Chairs program and NSERC. A section in [Chapter 3](#) covers freight transportation assignment and another on on-street parking equilibrium. The former emerged from my role as a postdoc under Stephen Ritchie at UC Irvine for a project developing a statewide freight model for California. The latter was developed with my former MSc student Ahmed Amer. His thesis was supported by Ryerson University's Centre for Urban Research and Land Development.

[Chapter 4](#) as a whole emerged from my experience as a postdoc working with Will Recker at UC Irvine (as I was cosupervised by both him and

Steve Ritchie). This precious time brought me insights to activity scheduling and its intersection with mobility systems and inverse problems. I also had the pleasure of collaborating with Jee Eun Kang, which is evident in the material presented in [Chapter 4](#). My subsequent work in this area with Shadi Djavadian and Adel Nurumbetova led to the mHAPP and activity scheduling market equilibrium extensions. This work was supported by an NSERC Discovery Grant. I also had the pleasure of collaborating with Soyoung You and Steve on the freight analog of this activity scheduling research which I included at the end of [Chapter 4](#). Currently my student Yueshuai He is investigating the use of the MATSim tool, which shares many common fundamentals with the mHAPP framework. I could not have completed the portions in [Chapter 4](#) related to MATSim without Yueshuai's help.

[Chapter 5](#) is the second part of what I developed from my postdoc experience with Will. I spent a few years thinking hard about this subject. After joining NYU in 2015, I finally had the chance to work on it with my student Jia Xu as well as with my long-term collaborator, Mehdi Nourinejad from University of Toronto. I am very grateful to both of them for their contributions, which make up a big portion of this chapter.

[Chapter 6](#) is the shortest chapter in this book because its material is the newest. I am still learning a lot on this topic and I can imagine future versions might see this chapter expanding significantly. The work in this chapter is based on research with Yueshuai, who is supported by the NSF CAREER award.

While [Chapter 7](#) is largely from my Urban Transportation Systems lecture notes, I also have several sections drawn from my research. The server relocation problem with queue delay and the symbiotic network design problems were both developed with my former student Hamid Sayarshad, now a postdoc at Cornell University. Several of the contributions—relocation, continuous network design heuristics—come from my PhD dissertation under the supervision of Amelia Regan at UC Irvine. My dissertation was supported by the US DOT Eisenhower Transportation Graduate Fellowship, which provided me support for three academic years. The activity-based network design problem section was, of course, from my collaboration with Jee Eun and Will. For part of the chapter on line planning problems, I had indispensable help from my current student Gyugeun Yoon in developing the route generation example.

While [Chapter 8](#) is last, it happens to be based on my earliest contributions to this field. Much of it is based on my dissertation on real options applications to network investments and sequential/adaptive network

design and timing under Amelia Regan. I have since conducted research on optimal switching in a collaboration with Qianwen Guo at Sun-Yat Sen University and Paul Schonfeld at University of Maryland. I have also expanded on the “Chow-Regan” (CR) policy with Hamid.

Beyond the research material, I am also very thankful to friends and students who helped me with proofreading or editing various parts of the book: Daniel Rodriguez-Roman (it is in the appendix now!), Gisselle Barrera (less commas!), Weerapan Rujikiatkumjorn for his help in setting up the GitHub sites, and, of course, Amelia for her sage comments.

I am very fortunate to have had the opportunity to work with all these collaborators and students (both current and former). I have learned so much from them. In particular, I want to express my extreme gratitude to Will for agreeing to write a [foreword](#) for my book, one which left me on the verge of tears when I finally had the pleasure to read its touching and flattering appraisal of this book.

Lastly, I want to thank my wife Feng, who, as always, remained patient and supportive during this time. She is certainly my better half and I would not have made it through this process (or much of my earlier research and PhD studies) without her there by my side.

CHAPTER 1

Urban Transport Systems

1.1 INTRODUCTION

Technological innovations are shaping the opportunities and challenges that we face at a far faster pace than in the past. Powerful computing has only been possible for the span of decades; the Internet, half that time; and ubiquitous mobile access to that information has only been around for a few years. The information age of smart cities and Internet of Things offers a stark contrast to what was possible just a lifetime ago. We can visually communicate with others across the globe in real time with apps like Facebook Live and Livestream. Automated vehicles like EasyMile's EZ10 and Google's Waymo provide pilots in multiple cities around the world. Sensors are abundant; for example, NYU CUSP's Urban Observatory ([CUSP, 2017](#)) can monitor infrared and hyperspectral imaging signatures at the city scale using LIDAR and RADAR. These examples do not even include the numerous other innovations we are seeing outside of the information revolution: 3D printing, nanotechnologies, genetics, space exploration, and more. The opportunities brought about by information technologies are clear in the area of mobility: we are seeing the advent of autonomous vehicles, connected vehicles, shared mobility, mobility-on-demand, and mobility-as-a-service, all due to advances in information and communications technologies (ICTs).

Yet, access to these mobility advances is not equally available to all. In communities around the world, advances in mobility and the opportunities derived from them vary widely. Some cities have congestion pricing policies, while others have light rail transit or electric vehicle car-sharing fleets. The nature of transport is that there is no “one-size-fits-all” solution; the fitness of a solution depends on many attributes that include the built environment, the culture of the population, the environmental topography, the attributes of neighboring communities, the population’s history, among others. On top of these differences and the increasing pace of technological advances, the population will continue to grow under an increasingly volatile climate. For example, the current world population of 7.3 billion is expected to reach 9.7 billion by 2050 ([UN, 2015](#)), of which 70% of the population will be urbanized ([WHO, 2010](#)). This is equivalent to taking almost

all our current global population and squeezing us all into cities. In other words, not only are there many different situations to consider for mobility solutions, but these solutions likely need to quickly adapt to new innovations and societal changes.

Because of this heterogeneity in transport circumstances and solutions, there is a pressing need to understand how transport systems perform. Unlike other engineering products, transport systems cannot be simply “manufactured” and sold to consumers to be used in private. Furthermore, inefficient urban mobility is a public problem that requires pooled resources. Highways and transit services cost millions of dollars to build and operate and are typically built by public agencies. Even with the vast resources of some private companies like Google, Tesla, and Amazon to step in to address some of these mobility challenges, any missteps in operation may lead to poor public reception (e.g., Tesla crash: [Vlasic and Boudette, 2016](#)) resulting in loss of trust in the systems or regulatory action. As a result, it is crucial to develop models to analyze and evaluate different system alternatives. This is the function of *urban transport systems engineers*.

The purpose of this book is to compile and organize the science of urban transport systems, place it in the context of the evolving technologies, and to prepare professional engineers and scientists with tools to bring mobility solutions to the public market. It is intended for graduate students with some basic background in systems engineering, economics, mathematical programming, queueing, and random utility models (appendices with some background information are included). A formal definition of this field with examples is provided in the following sections. Before proceeding, some historical references are provided to motivate this work.

The science of urban transport largely stems from the pioneering work of economists and computer scientists in the early to mid-20th century. Ironically, today we see many other disciplines, such as computer science and data science, taking concepts and theories from urban transport as they grapple with applying their technologies to mobility problems. What made transport systems unique from other network science disciplines was the dependence on user behavior, as highlighted by [Wardrop's \(1952\)](#) principles. With the impending massive investments on the US interstate highway system through the Federal Aid Highway Act of 1956, researchers had to quickly come up with modeling tools to evaluate different investment alternatives. Thus was born the RAND report by Beckmann, McGuire, and Winsten on “Studies in the Economics of Transportation” in 1956. A detailed retrospective on this work is covered by [Boyce et al. \(2005\)](#).

This report is significant because in the absence of any prior framework to evaluate urban transport systems at the time, the researchers were able to derive the first model of interaction of a system of roadways with users who behaved selfishly ([Chapter 3](#) will cover more of the network equilibrium consideration).

Several more innovations in urban transport systems came along in the absence of a formal framework to evaluate transport systems. In the 1970s, the politics of public transit subsidies and the long history of lobbyists from the automobile industry culminated in the National Mass Transportation Assistance Act of 1974, which propelled researchers to study justification for subsidizing public transit services. During this time, Mohring published his seminal work on “Optimization and Scale Economies in Urban Bus Transportation” in 1972. The work proved that for public services that consider user service level, there are scale economies in production, which justifies the need for subsidies. Meanwhile, to forecast the demand for transit technologies that were not yet present for the Bay Area Rapid Transit, researchers led by McFadden established the fundamental theories for discrete choice models ([McFadden, 1974](#)).

By the late 1970s, it was clear that a formal framework was needed to embody and describe the urban transport systems science, particularly as researchers looked more toward system integration. [Manheim \(1979\)](#) presented such a framework in his book, which [Florian and Gaudry \(1980\)](#) expanded upon and has persisted since. The Manheim-Florian-Gaudry framework is adopted in this book as well.

The 1980s and 1990s saw the rise of freight transportation research (e.g., [Harker and Friesz, 1986](#)) due to the deregulation of trucking with the Motor Carrier Act and Staggers Rail Act of 1980. On the user behavior side, the need to understand more complex travel behavior led to the development of models to explain activity scheduling behavior ([Recker et al., 1986a](#)). Due to the dependencies between activity scheduling and dynamic transport services, [Chapter 4](#) will be dedicated to this topic. Intelligent transportation systems (ITS) began to flourish in the 1990s with the rise of the Internet. One innovation from this time was the development of national ITS architectures to provide organization and structure for technology adoption and interoperability.

An overview of the Canadian ITS Architecture, which is very similar to the United States one, is presented in [Fig. 1.1](#). Such architectures provide a structure for designing different types of transport systems and are divided into several dimensions. Service bundles or packages categorize the

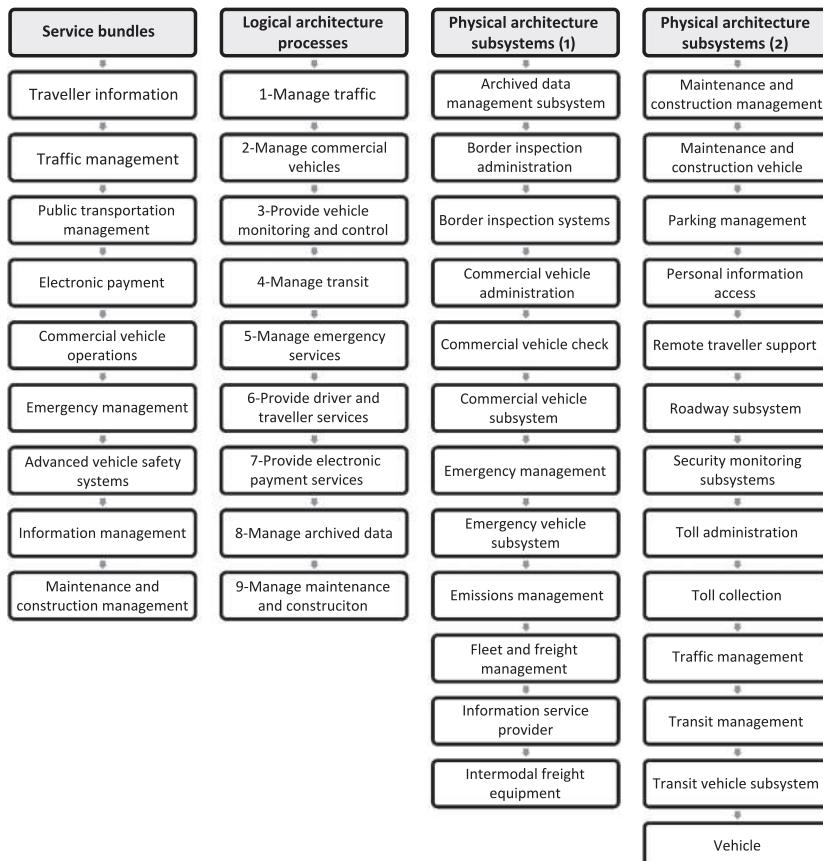


Fig. 1.1 Overview of Canadian ITS Architecture version 2.0.

technologies based on the general area of use. Technologies can also be categorized by logical processes in terms of their functionality. Lastly, technologies can be characterized by their physical architecture. As an example, a dispatch algorithm for on-demand transit vehicles may be used for emergency management, exhibit functions within “4-Manage Transit” and “5-Manage Emergency Services,” and require physical components derived from “Transit Management” and “Emergency Management” physical architectures. This architecture allows systems designers to determine the physical and functional needs of their systems with consistency.

Technologies for evaluating traffic systems matured by the 2000s to 2010s. Representative state-of-the-art system models include Polaris (Auld et al., 2016) and the MATSim multiagent simulation model illustrated in Fig. 1.2.

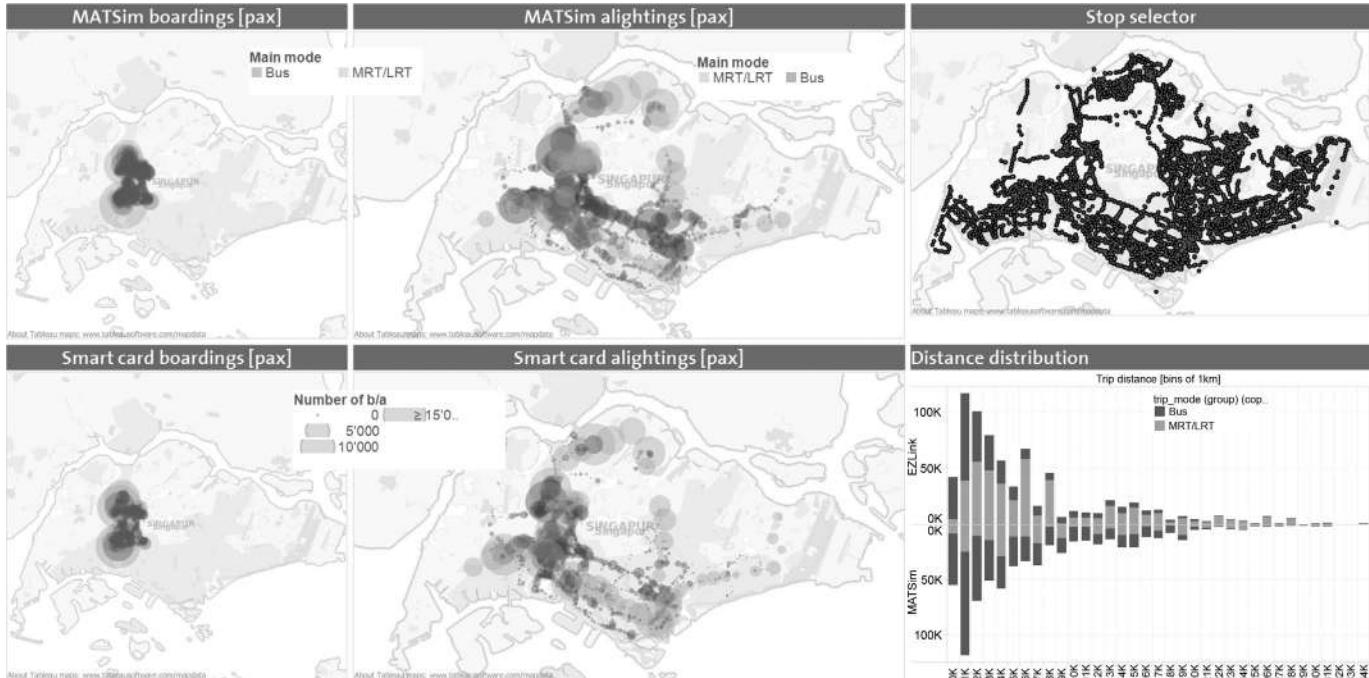


Fig. 1.2 Demonstration of use of MATSim to visualize public transport boardings and alightings in Singapore. (Source: Horni et al., 2016.)

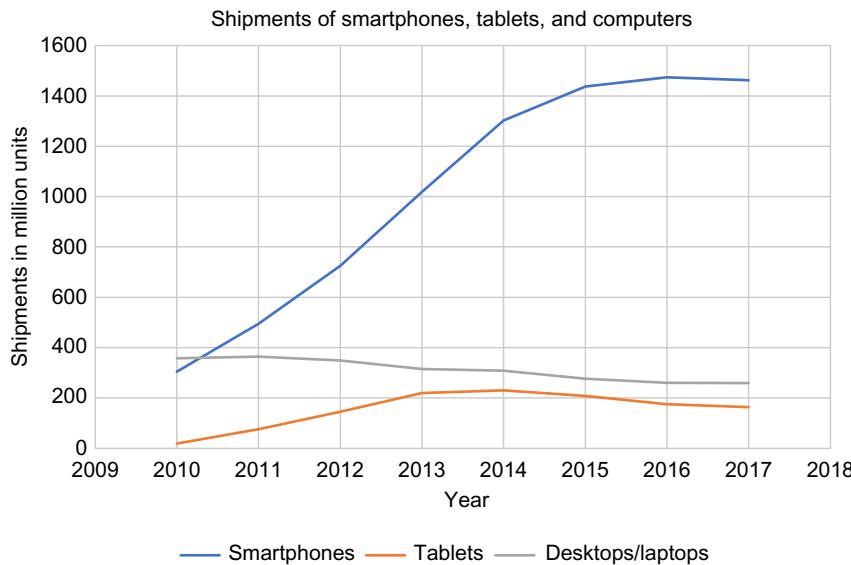


Fig. 1.3 Comparison of personal computer, smartphone, and tablet sales.

However, the late 2000s had two major events that reshaped the evolution of urban transport systems. First was the worldwide economic recession in 2008, which served as a significant external shock to travel behavior, allowing a new paradigm of shared mobility to grow (Miller, 2014). The second was the advance of mobile devices and smartphones, which was needed to operate the shared mobility systems. Fig. 1.3 shows the growth of computers and smartphones over the last 40 years, which shows how much mobile devices have permeated society in the last few years.

Simultaneous to the explosion of smartphone use has been the explosion of new information in urban transport system science, and with it a new paradigm of urban transport, not as an infrastructure component or asset, but as a service. “Mobility-as-a-Service” (MaaS) is a paradigm that focuses on service operation to support mobility of travelers. This new vision is illustrated, for example, in Finland’s MaaS framework shown in Fig. 1.4.

Given this history, is a book about urban transport systems warranted now? There are few other books on this topic. While there are earlier books that covered urban transport networks (e.g., Steenbrink, 1974), it was Stopher and Meyburg (1976) who presented one of the first books on Transportation Systems Evaluation. Manheim’s (1979) book on Fundamentals

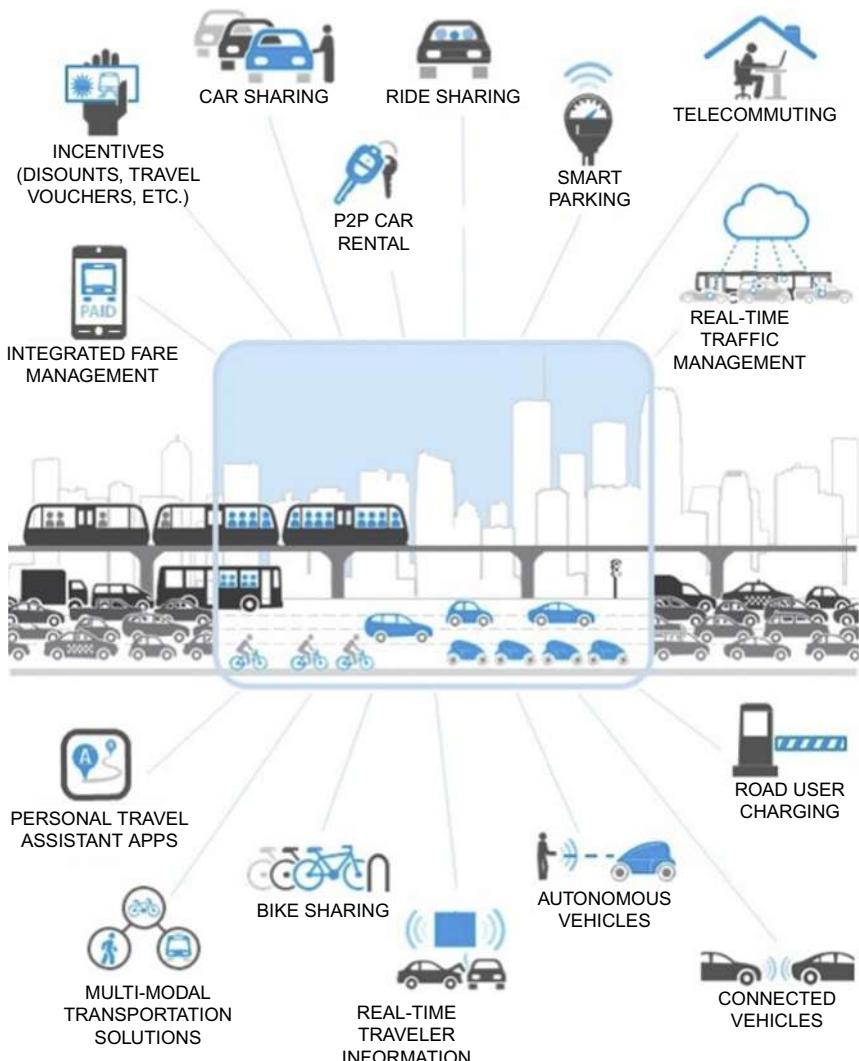


Fig. 1.4 A vision of Mobility-as-a-Service. (Source: Hensher, 2017)

of Transportation Systems Analysis further provided a framework on urban transport systems with user interaction. Sheffi's (1985) book on Urban Transportation Networks is also widely cited, focusing on the evaluation of road traffic networks in a planning context. Sussman (2000) gives an introductory overview of transport systems. The only significant work in recent years is the book by Cascetta (2009) and one by Möller (2014). Of those two, only Cascetta (2009) provides a detailed coverage of both the

supply and demand modeling methods that include within-day dynamics and interactions between supply and demand.

The scope of this book differs somewhat from the recent books. With the new shift in urban transport toward the MaaS paradigm, there is a greater need to compile a knowledge base to address challenges, both old and new, especially as they have evolved over the years. Based on discussions in a workshop on advanced transport modeling curriculum at the 13th International Conference on Travel Behavior Research in 2012, [Chow et al. \(2013\)](#) argued for adaptation to a growing scope of transport science while maintaining core concepts. Evaluation techniques, as discussed in [Cascetta \(2009\)](#), are not just limited to road networks or to fixed route transit services, as illustrated with the advances in MaaS. The presence of Big Data and ICTs means system operations can be much more dynamic and flexible as an informed process. Therefore urban transport systems engineers need to know more than systems evaluation. They need to also know about system optimization and inference considering information availability.

These three aspects—evaluation, inference, and optimization—form the structure of this book. In terms of topics, informed or data-driven systems require knowledge of dynamic optimization, learning, and privacy. These will be covered from the perspective of urban transport systems.

1.2 URBAN TRANSPORT SYSTEMS (UTS): DEFINITIONS

More detailed definitions are given of “urban transport systems” and the components involved in such systems. In this term, the “urban” and the “systems” are descriptors applied to transport (or “transportation” in North America).

Definition 1.1 A *system* is “a construct or collection of different elements that together produce results not obtainable by the elements alone” ([INCOSE, 2017](#)).

Definition 1.2 *Systems engineering* is a “methodological, disciplined approach for the design, realization, technical management, operations, and retirement of a system” ([INCOSE, 2017](#)).

An introduction to systems engineering is provided in [Appendix B](#). A *transport system* is an instance of a system with elements presented in [Table 1.1](#).

Table 1.1 Elements of a transport system

Element	Description
Users	A population of users of the transport system, accompanied by information about <i>preferences</i> , <i>beliefs</i> , and specific <i>space-time-need</i> constraints.
System state	The system state is the set of attributes of the system that may be inferred based on <i>informatics</i> obtained from the <i>environment</i> , which may be revealed stochastically or dynamically. A user's knowledge of the system state may be filtered by their <i>beliefs</i> and the <i>informatics</i> .
Environment	All elements external to the transport system is referred to as the environment. New information can arise either stochastically or in a stochastic dynamic manner where some information is known up to a point in time.
Operating horizon	The operating horizon is the length of time in which the system's operation is under consideration. This is typically divided into three levels: strategic (months or years), tactical (days or weeks), and operational (minutes or hours).
Transport operators	One or more decision-makers that decide the <i>policies</i> used to run the systems to achieve their <i>objectives</i> ; if more than one operator is present, coexisting relations need to be included (Chow and Sayarshad, 2014).
Transport policies	Policies are the rules that determine how the transport system is operated or what <i>decision</i> to make under a given <i>system state</i> over an operating <i>horizon</i> . Policies may be enacted manually, automatically by a computer, in a centralized manner, or in a decentralized manner.
Transport informatics	The informatics is the design of the flow of information between the <i>environment</i> , the <i>operator</i> , and the <i>users</i> of the system.
Transport links	The links are the elements that move users from one <i>node</i> to another. Links may represent abstract connections through space-time, physical roadways, or a specific transit vehicle on a line. Each link has a set of attributes such as <i>costs</i> .
Transport nodes	Nodes are interchanges for links, as well as sources and sinks for <i>users</i> .
Transport vehicles	Some systems deploy vehicles to serve <i>users</i> .
Operator objectives	The objectives of an operator are criteria used to justify decisions made. Objectives include maximizing profit, minimizing operating or user cost, or maximizing social welfare.

Continued

Table 1.1 Elements of a transport system—cont'd

Element	Description
Operator decisions	The decisions capture the wide range of actions that can be taken by a transport system operator: adding new links, setting service frequency, setting a fare or toll, price adjustment for ride sourced drivers, rebalancing empty bikes or vehicles, etc. Decisions are exogenously evaluated, optimized, or endogenously set by a policy.
User policies	User policies determine the rules that determine how they move from <i>node</i> to <i>node</i> . An example is a shortest path problem based on travel time. Another example is an optimal strategy for traversing hyperpaths in transit networks (Spiess and Florian, 1989) or the household activity pattern problem for activity routing policy (Recker, 1995).
User preferences	Each user has a set of preferences such as value of time, desired arrival times and lateness penalties, or parameters for the user's <i>policy</i> .
User beliefs	When information is filtered and incomplete, users have beliefs or perceptions of what the incomplete information is.
User space-time-needs constraints	Space-time-need constraints represent the combination of constraints along the three dimensions of traveler behavior, which can be at the trip level or the activity level (Chow and Nurumbetova, 2015). Examples: when they need to depart, where they live, whether they need to drop off kids at school, whether they have to work 8 h, etc.
User flow	User flow is the movement of users in the system from node to node. This can be individual flow or aggregate flow for a population. The flow can also be probabilistic (e.g., in the case of "hyperpaths"—see Chapter 3).
Vehicular flow	The flow of vehicles may need to be tracked separately from user flows. This is especially the case in shared mobility systems where vehicles serve multiple user trips.
Link costs	The characteristics of a link that are associated with the difficulty of transporting users from node to node. These costs can be generalized to include multiple types of costs: fares, tolls, in-vehicle travel time, wait time, transfers, etc.

Table 1.2 Additional elements for an *urban transport system*

Element	Description
Transport capacities	The capacities in the system, which can be at the link, the node, the path, or the vehicle level.
Transport schedule	A schedule represents the temporal constraints of a system.
Link cost function	Instead of a fixed cost value, the link's cost becomes a function of the flow.
User preference distribution	The user preference is a distribution instead of a single value.
User belief distribution	The user belief is a distribution instead of a single value.

These elements are expanded further in an *urban transport system*. What makes such a system unique is the urban descriptor. It implies that the system operates in a built environment where the system performance is explicitly sensitive to density and diversity of individuals. This can be captured with spatial-temporal capacity effects in the system that requires users to change their travel choices. It can also be captured with congestion effects, where the additional user's presence impacts the performance of the system incrementally. Diversity can be captured with distributions of user preferences or beliefs, such as desired arrival times to a destination. These additional elements are presented in [Table 1.2](#).

Any urban transport system can be described and distinguished from other systems or even the same system operating in a different manner, using the terminology shown here.

1.3 EXAMPLES OF THE NEED FOR UTS ENGINEERS

To illustrate the terminology, consider the following examples of systems that have recently failed in some way. These examples serve to motivate the need for a greater role to be played by UTS engineers ([Exercise 1.1](#)).

Exercise 1.1

Determine the unique UTS components of Toronto bike share.

Bixi was a bike-share program with service in Montreal and Toronto. In Toronto, the service was costing around CAD\$1.5 million to operate in 2013 with 1000 bicycles spread over approximately 80 stations (see [Fig. 1.5](#)). Due to high operating costs and insufficient cost recovery, the service was sold to the Toronto Parking Authority and rebranded as Bike Share

Toronto in 2013 ([Moore, 2013](#)). To better understand the issues with this system, its unique aspects are highlighted next.

Users: The population of users included regular users and tourists. The travel behaviors of these groups were different, resulting in a two-user class population. Regular users may link shared bike trips as last mile portions of multimodal trips. Tourists may have less information about the attributes of the road network. Since reservations were not needed, the total number of users in the system at a given time of day was stochastic. Users typically had real-time information from their mobile devices on the availability of bikes in the system, so their choice of which station to pick up or drop off bikes should take this into account.



Fig. 1.5 Distribution of BIXI bike share in Toronto in 2012. (Source: [Chow and Sayarshad, 2014](#).)

User flow: The user flow involves a user picking up an available bike at a dock and dropping off that bike at an empty dock somewhere else. Access and egress from a station would be considered if station siting was being considered. Flows from station to station must be paired with bike flows.

Vehicle flow: Bikes did not always flow in the same manner as user flows. Idle bikes may be relocated by the operator. Bike flows needed to be

conserved at bike stations. If the focus was on operational strategies, one might also designate a set of flows for the trucks used to pick up and relocate idle bikes.

User space-time-needs constraints: The system used a pricing mechanism (penalties beyond half hour usage) to impose a duration constraint on usage of a bike. Users were also constrained by having to start or end their bike trips at a shared bike station with sufficient capacity.

Transport informatics: The system provided users with near real-time information of bike stations (updated over time intervals) so they would know the availability of bikes and docks if they had access to a mobile device.

Transport capacities: The service included bikes and their docks. Both were limited in availability. If there was not enough supply to meet the demand at a preferred location, users would either have to go to another station, wait at the current station, or choose an alternative transport mode.

Transport operators: Users of the Bixi system may be accessing the shared bikes upon departing from a transit station. In that case, the performance of Bixi and the transit routes were dependent on each other.

Transport policies: Due to the potential for spatially imbalanced supply and demand, Bixi needed to run a bike rebalancing policy. Another example transport policy involved penalizing users for taking out a bike for more than 30 min.

User preferences: There was a distribution of user preferences for using the bike share and these fed into a route choice user policy. Bike preferences had many factors that played prominently in Toronto: weather conditions, availability of bike lanes, and the grade of the roadways, among others.

The cost function for the Bixi system would have been a function of the following elements shown. The term ω refers to a realization of a random element.

Bixi cost = $f[\text{users}(\omega), \text{user preferences}, \text{rebalance policy}, \text{public transit operator}, \text{transport capacities}]$

Like bike sharing, car-sharing services have been growing in cities around the world. There are different types of car-sharing services. For example, Zipcar's service is a round-trip service that requires a car to be returned to the location where it was picked up. Car2Go, on the other hand, has "free floating" systems where users can pick up a vehicle located anywhere within a predefined region and drop them off anywhere else in the same region. While some of the services have been shown to be profitable (Guilford, 2016), there are also examples where they have failed. One such

case is Car2Go's business in San Diego. Originally set as an electric vehicle (EV) fleet, it had to be replaced with a gasoline fleet ([Exercises 1.2 and 1.3](#)). Despite the change, the service was not getting enough demand and announced it was shutting down at the end of 2016 ([Krok, 2016](#)).

Exercise 1.2

Determine the unique UTS components of EV taxis in Hong Kong.

Electric vehicle taxis (e-taxis) have been gaining interest in many cities because they can reduce greenhouse gas emissions and air pollution (although savings are not always the case, as shown by [Huo et al., 2015](#)) without the same range anxiety experienced by owners of personal vehicles.

In Hong Kong, the company BYD tried to launch an e-taxi service for 2 years with great difficulty ([He, 2015](#)). According to the manager, the main components contributing to the high costs were the costs of charging stations and vehicle maintenance. To understand this system better, the system is described as follows.

Users: The population of users included regular users and tourists. The total number of users in the system at a given time of day can be regarded as stochastic.

Transport nodes: Nodes need to be defined for locations of electric charging stations. Their locations and number determine the average time spent by an e-taxi when it needed to recharge. Separate nodes need to be used to represent user demand.

Transport policies: The taxis operated in a decentralized manner, so a separate policy was assumed for each. This policy needed to consider not only picking up and dropping passengers, but also when and where to charge the vehicle when the state of charge was low.

Vehicle flow: The flow of taxis depended on the picking up and dropping off of passengers from taxi stands or from spatial points in a region. EV taxis further needed to consider the flow of taxis to and from charging stations.

System state: The state of the system needed to include the state of charge of each of the vehicles in the fleet.

Transport informatics: The system operated in a way that user demand for a trip was not known until they made a request in real time. Hence, the system operation needed to be conducted in a dynamic manner. Similarly, ride-hailing taxi systems did not convey vehicle location or wait time information to passengers, so they made choices based on beliefs of the vehicle availability and costs.

Transport capacities: The process of going to a charging station to get recharged is a queueing system. Due to long charging times (half hour to one hour for the fast chargers), queue delay was a significant problem.

Environment: The travel times on the roadway depended on the traffic sharing that road. Since the traffic was not making decisions, it can be treated as an external random element from the environment.

The resulting system cost was based on the following factors:

e-taxi cost = f [transport nodes, transport policies [system state], users, transport informatics, transport capacities, environment]

Exercise 1.3

Determine the unique UTS components of Kutsuplus in Helsinki.

Helsinki, Finland, was one of the first cities to try to implement a microtransit service—a shuttle bus system that responded to on-demand requests to pick up and drop off passengers door to door. This system, called Kutsuplus, had its own mobile app for ticketing and informatics. However, after a year and a half, it was shut down at the end of 2015 due to escalating operating costs (Kelly, 2015). Subsequently, some of the earlier routing methodology studied at Aalto University (Hyttiä et al., 2012) was examined by Sayarshad and Chow (2015) and found that it could be improved further with responsive pricing.

As an on-demand “dial-a-ride” service, the system shared similar characteristics to the taxis. However, it also had a centralized dispatch, shared rides resulting in capacities and consequences to routing decisions.

Users: The population of users included regular users and tourists. The total number of users in the system at a given time of day can be regarded as stochastic.

Transport nodes: Nodes need to be used to represent user demand.

Transport policies: The system operated under a centralized dispatch policy where vehicles were assigned to passengers and their routes. The policy needed to be dynamic and dependent on the informatics. In the routing, the system may have user detour as a constraint, requiring passengers detours not to exceed some percent above their door-to-door travel time.

Operator objectives: This system was publicly funded so its objective was oriented to users’ welfare.

User flow: Travelers using the system may not be directly dropped off if another passenger must be picked up or dropped off first. Therefore it was necessary to track user flow at an activity level, in terms of pickup and drop-off locations and the sequence of stops made. Due to vehicle capacities, it was necessary to track user flows in a time-space setting, including their arrival and departure times in the system.

Vehicle flow: The flow of vehicles needs to be tracked individually since user flows were assigned to the vehicles. Vehicle flows depended on the transport policies for routing and dispatch. They are tracked at an activity level, in terms of pickup and drop-off locations and the sequence of stops made.

Transport informatics: The system operated in a way that user demand for a trip was not known until they made a request in real time. Hence, the system operation needed to be conducted in a dynamic manner. Information was conveyed to passengers so they knew their wait times and maximum travel times.

Transport capacities: The vehicles had limited seating so vehicle capacities were explicit.

Transport schedule: Because the system was on-demand with booking and dispatch, a schedule was needed to track the times of arrival of each vehicle at different locations. In the case of operational analysis, both planned and actual schedules were needed.

The resulting system cost was based on the following factors:

Kutsuplus cost = f [transport nodes, transport policies, transport informatics, transport capacities, operator objectives, users]

How would one evaluate the performance of these systems and go about designing an operating policy? While this section provides taxonomy to classify diverse systems, there is no framework for analysis. For example, if fare payments are of most interest to a decision-maker, then the system operator may want to consider the user preferences.

1.4 MANHEIM-FLORIAN-GAUDRY (MFG) FRAMEWORK

How should one go about modeling an UTS and analyzing it? What are the key processes that influence the performance of the system? For that matter, how should one evaluate any generic MaaS system? The framework from [Manheim \(1979, 1980\)](#) is adopted. The primary point of the framework is that a UTS should focus on the market clearing interaction between an activity system **A** and a transportation system **T**. The framework allows a modeler to relate the different state variables in both the long- and short-term planning horizons together. While the framework was derived in a time where mobility was primarily based on personal automobiles and fixed route public transit, it is flexible enough to accommodate the challenges presented in modern data-driven MaaS systems.

Florian and Gaudry (1980) further expanded on this market equilibration framework by explicitly identifying the functions or procedures needed to complete the feedback loop. This loop is shown in Fig. 1.6, redrawn from McNally's (2007) rendition.

In this framework, there is a procedure **D** that takes the activity system **A** and performance procedure **P** to obtain a demand for the transport system **T**. Meanwhile, there is a performance procedure **P** that determines how the system **T** performs, based on the demand procedure **D**. This framework asserts that this is a system of relationships, in which market equilibrium is achievable. The outcome of the equilibration is a set of realized system states **F** that include some of the operator decisions (e.g., vehicle flows, operator's transport schedule), the user flows, and the realized link costs. The equilibration occurs at a within-day, short-term operational decision-making level. The feedback loops consider long-term dynamic system equilibrium, where the location of activities **L** depends on the system costs and include tactical or strategic decision-making variables. For example, improved travel times to the central business district (CBD) may encourage workers to move further away or for more businesses to move closer to the CBD. These decisions impact the activity system **A**. Similarly, the system performance provides feedback to operators and government agencies on

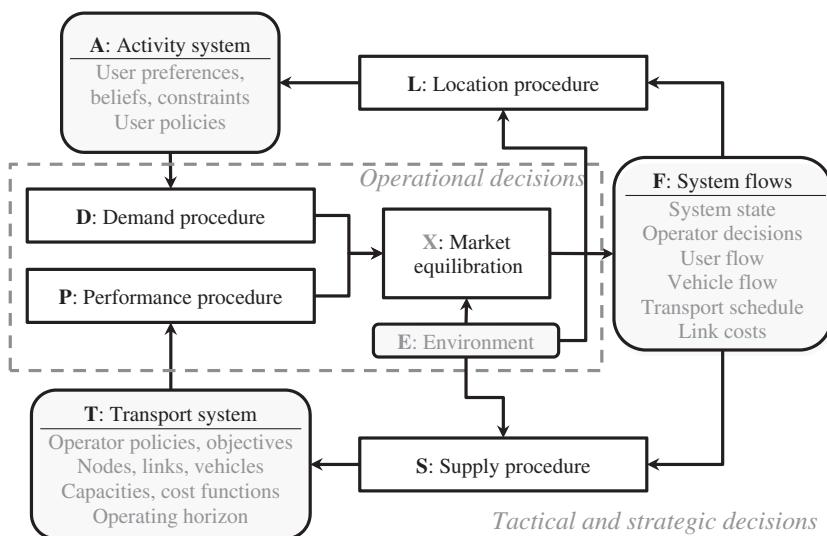


Fig. 1.6 Manheim-Florian-Gaudry framework for transport systems analysis, with new elements in gray.

needs for further resource allocations **S**. This is the point of intervention where system design changes may be incurred on **T**.

A new system, the environment **E**, is added as an element that does not change over multiple iterations but distinguishes UTSs in different cities. For example, the wintry climate and dense built environment in Toronto will tend to affect the system performance and market equilibration of shared bikes differently than the warm climate with sprawled built environment in Los Angeles. The state variables are explicitly listed to point out the variables that need specific data for a particular case. These variables more clearly distinguish the role of operators' operational decisions such as vehicle flows and transport schedule. The symbol **X** is used to describe the equilibration procedure. The two layers of short- and long-term decision-making are also distinguished.

The MFG framework is used to identify elements and functions that need to be considered when trying to address an UTS problem. For example, a highway expansion should not be considered in isolation, as it would lead to more demand shifting to use the road through a route choice demand procedure. This equilibration leads to a realized travel time and congested scenario, causing long-term shifts in economic activities and feedback for the next operator or agency system design intervention. To illustrate the use of this framework for MaaS, let us look once more at the three examples from earlier ([Exercises 1.4–1.6](#)).

Exercise 1.4

Determine the unique aspects of MFG components for Bixi Toronto.

Analysis of Bixi Toronto as a transport system under the MFG framework would involve identifying the components that make up [Fig. 1.6](#). These are divided into the components representing the input and output states (A, T, E, F) and the functions that transform them (D, P, X, L, S). The set of state variables should also be the same for different shared bike systems that were described earlier. The functions should be the same for different shared bike systems in different cities. We describe these here.

D: For a given population of Bixi users that included commuters and tourists with activity origins and destinations, a proper demand procedure **D** should assign the departure time and station pickups and drop-offs. Note that origin/destination is already given because that is assumed to be done at a tactical level and is not directly dependent on the equilibration. However, based on the MFG framework, origin-destination

patterns and activity schedules would be impacted indirectly because of the outer loop over the long term. The function for **D** should be as follows.

[Bixi station desired pickups and drop-offs distributions and parameters by time of day] = **D** [user activity schedules, user mode preferences, spatiotemporal distribution of Bixi travel costs]

P: The performance procedure takes the bike fleet, station locations, operating policies, and demand levels to determine the spatiotemporal distribution of Bixi travel costs. There are three primary operating policies that directly affected this system's operational horizon: a rebalancing policy that assigned idle bikes to stations, an information policy that determined what information travelers had of the bikes, and a pricing policy that determined payments (or incentives) to charge users. Based on these attributes, a performance procedure determines the average travel times, wait times, and costs. The procedure in this case is a spatial queueing system.

[spatiotemporal distribution of Bixi travel costs] = **P**[bike fleet, station locations, rebalancing policy, traveler information policy, pricing policy, spatiotemporal distribution of demand for pickups and drop-offs]

X: A market clearing mechanism is used to assign the user demand to user flows in this system and the bike and station inventories into bike flows. The mechanism for Bixi in Toronto differed from other locations due to the presence of environmental factors specific to Toronto, **E**. Whereas **D** and **P** assume the output of the other procedure is fixed, the equilibration of the two assumes that both are endogenous variables operating under a certain type of market. In the case of traffic without any operational decisions from traffic operators, one may assume the market consists of noncooperative route choices made between travelers resulting in Wardrop's user equilibrium. In the case of Bixi where there is a set of dynamic decisions from both users and operators, a more generalized two-sided stable matching principle is needed. These mechanisms are discussed in more detail in [Chapters 3 through 5](#).

[Bixi bike flows, user flows, equilibrated system costs] = **X**[**P**, **D**, **T**, **A**, **E**]

L: The location procedure represents the long-term decisions made by users in response to the equilibrated system costs of Bixi. The impact may include their choices of activities and their scheduling of them particularly those accessed using transit services that were located near Bixi stations. This affected their preferences for transit and consequently it is also where

external changes may occur. Examples include population growth additional tourism introduction of new activity centers changes to the coexisting public transit service such as new stations or bikes and so on

$[user\ activity\ schedules,\ user\ mode\ preferences] = \mathbf{L}[equilibrated\ Bixi\ costs,\ external\ user\ effects,\ external\ public\ transit\ decisions]$

S: The supply procedure represents the long-term decisions made by Bixi in response to the equilibrated systems costs. The impact may include redistribution of the bike docks and stations as well as changes in the operating policies. As a long-term procedure Bixi may have externally introduced additional budget to expand the system (this is a network design problem) or introduced new policies altogether such as enhanced predictive modeling to modify the rebalancing policy or pricing policy.

$[bike\ fleet,\ station\ locations,\ rebalancing\ policy,\ traveler\ information\ policy,\ pricing\ policy] = \mathbf{S}[equilibrated\ Bixi\ costs,\ external\ Bixi\ decisions\ and\ resources]$

Exercise 1.5

Determine the unique aspects of MFG components for EV taxi system in Hong Kong.

EV taxis in Hong Kong differed from the Bixi system in that the supply side was operated by a decentralized population of operators as opposed to a single centralized authority. In addition, the performance of the system depended on the transport nodes representing charging stations. We describe the interactions of these variables here using the MFG framework.

D: The demand procedure was like that of the Bixi system.

$[EV\ taxi\ desired\ pickups\ and\ drop-offs\ distributions\ and\ parameters\ by\ time\ of\ day] = \mathbf{D}[user\ activity\ schedules,\ user\ mode\ preferences,\ spatiotemporal\ distribution\ of\ EV\ taxi\ travel\ costs]$

P: The performance procedure took the taxi fleet, charging station locations, operating policies, and demand levels to determine the spatiotemporal distribution of taxi travel costs. Unlike the Bixi system, the EV taxi system consisted of a fleet of decentralized decision-makers and the costs of these vehicles depended on the location of charging stations. There were three primary operating policies that directly affected this system's operational

horizon: a cruising or coverage policy for where idle vehicles searched for customers, a customer matching and information technology (e.g., e-hailing or advanced booking options), and a pricing policy that determined payments (or incentives) to charge users. Based on these attributes, a performance procedure determined the average travel times, wait times, and costs. The procedure in this case was a spatial queueing system, although there were two sets of queues happening: the queueing for electric charging and the queueing for customers.

[spatiotemporal distribution of EV taxi travel costs] = **P**[taxi fleet, charging station locations, cruising/coverage policy, traveler matching policy, pricing policy, spatiotemporal distribution of demand for pickups and drop-offs]

X: The market clearing mechanism was similar to the Bixi system. There was a set of dynamic decisions from both users and operators, requiring a more generalized two-sided stable matching principle. What complicated this further was that the equilibrium must account for decentralized decision-making among the taxis as well.

L: The location procedure represented the long-term decisions made by users in response to the equilibrated system costs of the EV taxi and was similar to the Bixi procedure.

[user activity schedules, user mode preferences]
= **L**[equilibrated EV taxi costs, external user effects]

S: The supply procedure was also like that of the Bixi system

[taxi fleet, charging station locations, cruising policy, matching policy, pricing policy]
= **S** [equilibrated EV taxi costs, external EV taxi decisions and resources]

Exercise 1.6

Determine the unique aspects of MFG components for Kutsuplus.

Kutsuplus, as an on-demand shuttle service, acted similarly to the EV taxi service in Hong Kong except that multiple riders could share one vehicle, the operator decision-making was centralized, and there was no EV charging constraint.

D: The demand procedure was similar to the other two systems.

[Kutsuplus desired pickups and drop – offs distributions and parameters by time of day] = **D**[user activity schedules, user mode preferences, spatiotemporal distribution of Kutsuplus travel costs]

P: The performance procedure took the fleet, operating policies, and demand levels to determine the spatiotemporal distribution of taxi travel costs. There were four primary operating policies that directly affected this system's operational horizon: an idle vehicle positioning policy, a customer matching and information technology (e.g., e-hailing or advanced booking options), a passenger detour and routing policy, and a pricing policy that determined payments (or incentives) to charge users. One detour policy might require that passengers not exceed a certain maximum detour from a point-to-point trip. Based on these attributes, a performance procedure determined the average travel times, wait times, and costs. The procedure in this case was a spatial queueing system.

[spatiotemporal distribution of Kutsuplus travel costs] = **P**[vehicle fleet, idle vehicle location policy, matching policy, routing/detour policy, pricing policy, spatiotemporal distribution of demand for pickups and drop-offs]

X: The market clearing mechanism was similar to the Bixi system. There was a set of dynamic decisions from both users and operator, requiring a more generalized two-sided stable matching principle.

L: The location procedure represented the long-term decisions made by users in response to the equilibrated system costs of the Kutsuplus and was similar to the Bixi procedure.

[user activity schedules, user mode preferences]
= **L** [equilibrated Kutsuplus costs, external user effects]

S: The supply procedure was also similar to the Bixi system.

[vehicle fleet, idle vehicle location policy, matching policy, routing policy, pricing policy] = **S**[equilibrated Kutsuplus costs, external Kutsuplus decisions and resources]

1.5 A SIMULATION TOOL FOR EVALUATING UTS: MATSim

The previous examples show that there is a wide range of UTSs. Practical tools for running the market clearing mechanism step often either ignore the activity scheduling behavior of users or only consider traffic and fixed route transit systems. One of the few tools flexible enough to address both points and is also open source is MATSim (<http://matsim.org/>). This is

adopted as the tool of choice for evaluating the transport systems described in this book.

As detailed in [Horni et al. \(2016\)](#), MATSim is a multiagent simulation platform designed to capture users' activity scheduling responses to different transport system designs. On the supply side (the **P**), the simulation handles:

- mesoscopic road traffic flow,
- traffic signal control,
- lane management,
- parking,
- electric vehicles,
- road pricing,
- fixed route public transit,
- on-demand transit,
- multimodal trips,
- car sharing,
- dynamic MaaS, and
- truck deliveries.

Such a tool allows researchers to consistently test and compare new operating policies or algorithms using the same underlying simulation of the equilibration portion of the MFG framework: the **D**, **P**, and **X** procedures. For example, a new rebalancing algorithm for a car-sharing service can be evaluated by implementing it in a study area where real data is available to calibrate from—a simulation test bed—and comparing against a baseline without car sharing or with car sharing operating under a benchmark algorithm. What is more, the algorithm can be further compared to other operational policies, such as road pricing or improvement of transit service coverage, using the same set of performance measures. MATSim operates under a utility-based modeling framework, so measures based on consumer surplus or total travel disutility can be compared.

MATSim has been implemented in a number of cities and regions around the world, including: Gauteng (South Africa), Seoul (South Korea), Munich (Germany), Toronto (Canada), Berlin (Germany), Zurich (Switzerland), Padang (Indonesia), Joinville (Brazil), Caracas (Venezuela), Aliaga (Turkey), Poznan (Poland), Tel Aviv (Israel), Singapore, Vorarlberg (Austria), Quito (Ecuador), Izmir (Turkey), and New York (United States). The functionality of the system is shown in [Fig. 1.7](#). A more detailed coverage of the tool is provided in [Chapter 4](#).

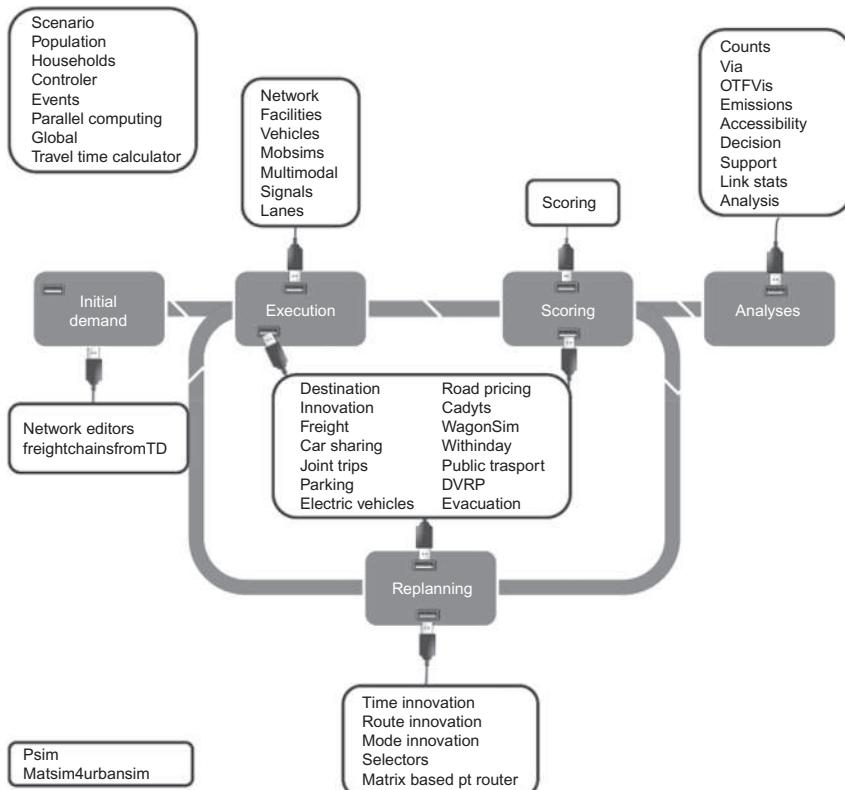


Fig. 1.7 MATSim functionality. (Source: Horni et al., 2016.)

1.6 USE CASE MOTIVATIONS FOR BOOK CHAPTERS

The rest of this book is designed to examine urban transport systems in greater depth. Recent advances in learning, evaluation, and optimization are covered so that this book can be used as an all-in-one guide for UTS engineers to tackle any problem, whether it is designing a highway system, setting a congestion pricing scheme, or operating a microtransit service. The remaining chapters are listed here with example use cases to provide an overview of their purposes.

Chapter 2: Monitoring mobility in smart cities

- To design an informatics system to collect data for an urban transport system;
- To use the data and informatics to describe mobility in cities;

- To explain, measure, or monitor the impact of external events on mobility in cities;

Chapter 3: Network equilibrium under congestion

- To evaluate the effect of a traffic, transit, taxi, or parking system design on users;
- To quantify the value of user information under certain systems;
- To evaluate the role of a public agency as a two-sided platform to engage transport operators and travelers in a smart city context;

Chapter 4: Market equilibrium with activity scheduling

- To evaluate the effect of spatiotemporal changes on activity participation, time of day scheduling, and trip chaining;
- To evaluate users' preferences for different schedule or capacity designs of a multimodal system;
- To evaluate time use substitution effects of other technologies related to mobility, for example, smart grid or e-commerce;

Chapter 5: Inverse transportation problems

- To make short-term predictions of transport system characteristics using real-time data;
- To calibrate latent attributes of a transport system;
- To monitor a system's network attributes using real-time data;

Chapter 6: Privacy in learning

- To provide privacy protection for user or operator data to encourage collaboration;
- To control for privacy trade-offs with performance reliability in system design;
- To design privacy-aware information systems;

Chapter 7: Network design

- To optimize designs or operational strategies in a network setting;
- To evaluate sensitivities of design solutions to different attributes;
- To develop an automated system operator;

Chapter 8: Network portfolio management

- To operate an automated system operator in an online environment;
- To operate an autonomous vehicle fleet;
- To develop a portfolio management system for a transport network;

In addition to the eight chapters, four appendices are provided for supplemental material on university research efforts, systems engineering, network analysis, and discrete choice modeling.

RESEARCH AND DESIGN CHALLENGES

The following are open-ended challenges that sometimes refer to real data sets. Readers are encouraged to try to tackle the problems, and post their designs or solutions on the following GitHub page, <https://github.com/BUILTNYU/Elsevierbook/tree/master/Chapter%201>.

- (1.1) Take a hometown or college campus as a study area and identify a transport system there.
 - a. Draw out the functional flow for the system to include all the elements from [Table 1.1](#) and [Table 1.2](#). Construct a representative toy example.
 - b. Describe the system using use case diagrams, activity diagrams, and deployment diagrams. Define performance measures to use to evaluate the performance of the system with respect to the use cases.
 - c. Modify the informatics of the system such that users send real-time information to the operator (if that is already present, then take it out). Evaluate the impact in terms what new elements need to be considered and how the values may change in the toy example.
 - d. Add a new competing service as a fixed route transit. Evaluate the system changes that need to be made and how they are reflected in the toy example.
 - e. Replace the transport system with an alternative one (e.g., from fixed route transit to car sharing). Use the system elements to define the variables and models needed; use MFG framework to identify the analytical framework to compare this new system against the existing system.
- (1.2) Go to the Citibike map website (<https://member.citibikenyc.com/map/>). Consider a 15-min time interval. Pick out 10 bike stations as your system. Collect data on the state of this system over 30 min (3 time intervals at 0, 15 min, and 30 min marks). For the first two intervals, assume the change in numbers observed at each station represents the demand for bikes and empty space. If that is your arrival rate, and each station is an independent queue, design a bike relocation policy. Test your policy on the remaining time interval. What performance measures are appropriate here—how to determine the quality of the policy?
- (1.3) Visit <http://matsim.org/docs/tutorials>. Run the tutorial on simulation of public transport. Evaluate the performance of the system

under two scenarios: one where service headways are reduced by 20%, and one where the value of time of passengers increases by 20%. Discuss the implications using the MFG framework—which are the key interactions to focus on in this system?

- (1.4) For each of the data elements of the MFG framework (A , T , E , F), survey available public data sources that provide such information.
- (1.5) Design a toy example from scratch and populate it with variables for A and E . Assume procedures for D , P , X , L , S . Propose two alternative transport systems: a carshare (T_1) and a fixed route transit (T_2). Assume an informatics policy for both systems.
 - a. For each alternative system, determine the output performance measures for the equilibrated flows F under two settings: the first one, use the complete MFG framework; for the second one, ignore the outer feedback loop. Compare your findings.
 - b. Design a simulation-based test plan (see [Appendix B](#)) to evaluate each system. Compare the distributions of the performance measures under each system under base condition as well as a condition where the informatics policy is “turned off.”
- (1.6) Read the following news article from Citylab: <https://www.citylab.com/transportation/2017/11/a-bus-shunning-texas-towns-big-leap-to-microtransit/546134/>.
Describe this microtransit system in Arlington using the systems diagrams and the before/after processes of implementing such a system. Focus the description on the key trade-offs of the system.
- (1.7) Review the connected vehicle service packages in the ITS Architecture update: <http://local.iteris.com/arc-it/html/servicepackages/spsheritagecvria.html>.
Pick out two of the service packages and compare them using the MFG framework: identify all the components in the MFG framework that these two service packages, if implemented in a transport system, would impact and how.

CHAPTER 2

Monitoring Mobility in Smart Cities

2.1 INTRODUCTION

Whereas [Chapter 1](#) introduced the classic discipline of transport systems analysis to readers, this chapter presents new challenges and opportunities associated with smart cities. Regarding the “informed” aspect of urban transport systems design and operations, smart cities and Big Data innovations have altered how we obtain information. Through a combination of technological advances, citizen engagement, and policies that require greater visibility, a host of different sources of information have emerged, a sample of which are shown in [Fig. 2.1](#).

The figure shows five different types of data that were not readily available 15–20 years ago. [Fig. 2.1A](#) is an illustration of social media data from Twitter as a source for monitoring the spatial-temporal spikes in activity around major events such as Hurricane Sandy in NYC. [Fig. 2.1B](#) shows real-time monitoring of vehicular traffic made possible with the use of inductive loop detectors combined with a centralized informatics system developed by Caltrans called PeMS. [Fig. 2.1C](#) displays a website that was created to compile real-time and schedule-based public transit feeds made through a General Transit Feed Specification (GTFS) created by Google to standardize transit timetables and route information. This makes it possible to monitor a whole transit network if the real-time GTFS is available. [Fig. 2.1D](#) and E are additional data sources: call detail records from mobile phones and GPS trajectories from taxis.

These data sources were not available even two decades ago, but now they provide a rich source of information on urban transport systems and travel patterns of their users. The underlying technological advances fall into the area of smart cities, so an understanding of smart cities and the role they play in technological innovations is necessary for an understanding of informed urban transport systems design.

This chapter breaks into the evolution of smart cities research and examines specific applications in monitoring mobility in real time made possible

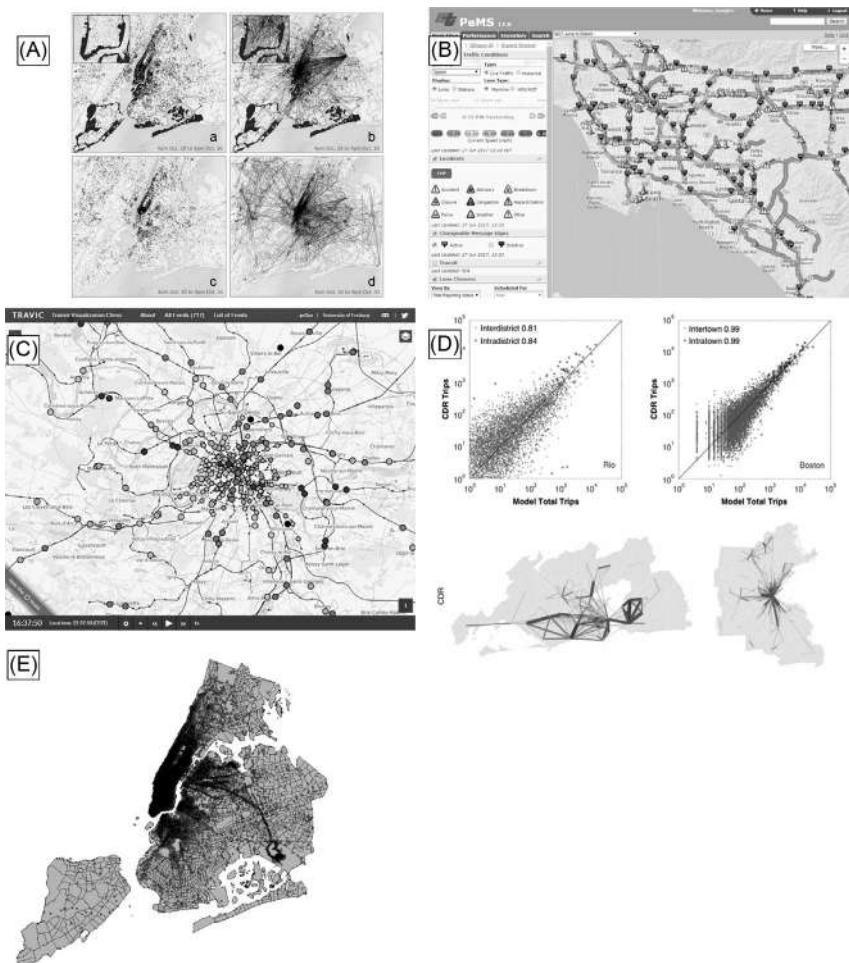


Fig. 2.1 Demonstrations of data opportunities due to advances in Big Data and smart cities: (A) using Twitter data to track urban mobility during Hurricane Sandy; (B) real-time California freeway traffic monitoring from Caltrans PeMS; (C) repository of real-time and scheduled GTFS feeds from around the world including Paris; (D) call detail records used to model travel patterns; (E) Taxi GPS data. ((A) Source: [Wang and Taylor, 2014](#); (B) source: <http://pems.dot.ca.gov/>; (C) source: <http://tracker.geops.de/?z=10&s=1&x=247786.5989&y=6240912.9552&l=transport>; (D) source: [Çolak et al., 2015](#); (E) source: [Sayarshad and Chow, 2016](#).)

with data sources like those in Fig. 2.1. The purpose is to show readers how to readily setup monitoring systems using today's ubiquitous data sources to inform an urban transport system. A new technique is introduced based on time-geographic monitoring of people's daily activities using vector fields of travel momentum.

Readers interested in exploring the classical methods of data collection and (primarily vehicular traffic) are referred to [Stopher and Meyburg \(1976\)](#) and [Washington et al. \(2010\)](#) for data analysis methods, and to [Ortuzar and Willumsen \(2002\)](#) for data collection and survey methods.

2.2 SMART CITIES, BIG DATA, AND THE INTERNET OF THINGS

What constitutes a “smart city”? Emergence of this phenomenon can be attributed to several different trends: the rise of Big Data, the increasing connectivity and digitization of the physical world, the emphasis on smart governance to more efficiently use limited resources, and the rise of artificial intelligence. Each trend has their own definitions of facets of smart cities, leading to a highly fragmented definition. Researchers have sought to converge on a common definition; the following definition from [Caragliu et al. \(2011\)](#) is presented as there is some agreement on its use (e.g., [Batty et al., 2012](#)).

Definition 2.1 ([Caragliu et al., 2011](#)). *Smart cities are cities where investments in human and social capital and traditional (transport) and modern (ICT) communication infrastructure fuel sustainable economic growth and a high quality of life, with a wise management of natural resources, through participatory governance.*

Under this definition, there are six categories of city functions that can be divided out, of which smart mobility is one category, as shown in [Fig. 2.2](#) ([Batty et al., 2012](#)) and similarly defined by [Albino et al. \(2015\)](#).

Smart economy	Smart people	Smart governance
<ul style="list-style-type: none"> Innovative spirit Entrepreneurship Economic image and trademarks Productivity Flexibility of labor market International embeddedness Ability to transform 	<ul style="list-style-type: none"> Level of qualification Affinity to lifelong learning Social and ethnic plurality Flexibility Creativity Cosmopolitanism/open-mindedness Participation in public life 	<ul style="list-style-type: none"> Participation in decision-making Public and social services Transparent governance Political strategies and perspectives
Smart mobility	Smart environment	Smart living
<ul style="list-style-type: none"> Local accessibility (Inter-)national accessibility Availability of ICT-infrastructure Sustainable, innovative and safe transport systems 	<ul style="list-style-type: none"> Attractivity of natural conditions Pollution Environmental protection Sustainable resource management 	<ul style="list-style-type: none"> Cultural facilities Health conditions Individual safety Housing quality Education facilities Touristic attractivity Social cohesion

Fig. 2.2 A typology of smart city functions.

“Smart mobility” deals with transport. However, the other five categories also relate to transport. “Smart economy” deals with the competitiveness of the city with respect to productivity. The economy depends on availability of transport to make activities accessible. “Smart people” deals with social and human capital, in how diverse, creative, and educated they may be. The equity of the transport system impacts the diversity of the population; the system also creates opportunities for people to pursue education and activities to learn. “Smart governance” deals with how engaged the public is with decision-making in the city, as well as the quality and transparency of city services. Some of the services provided by the government are related to transport options. “Smart environment” deals with sustainable resource management to minimize pollution, and transport is one of the major contributors to pollution. “Smart living” deals with the availability of educational and cultural facilities, safe living conditions, and general attractiveness to tourists. Mobility affects many of these aspects: safety, healthy living through active transport options, and accessibility to destinations.

This typology is clearly very complex, so what does a smart city look like if it embraces these functions? In December 2015, the US DOT launched a “Smart City Challenge” asking midsized cities to propose ideas for smart city initiatives and awarding the winning city up to \$40 million to implement those ideas. The winner was Columbus, OH, and the way the city integrated these functions is shown in Fig. 2.3. It features a matrix of different objectives—accessibility, logistics, connected citizens, sustainable transportation—along with the technologies, districts, and functional outcomes (safety, mobility, opportunity, and climate change). This example shows how important it is for a smart city to connect mobility with its other functions.

As shown in Fig. 2.3, one key consideration is the set of enabling technologies. Ultimately, urban transport systems theory is a science of evaluating and designing operational policies for transport technologies, and this task has grown increasingly more complex in a smart cities era. For one thing, transport technologies are not simply oriented around physical infrastructure; they are now cyber-physical in nature. As a result, information about the physical infrastructure can be monitored and communicated in a way that was not possible many years ago, through the concept of Big Data. An aggregation of defining components of Big Data from Gandomi and Haider (2015) is provided as follows.

Definition 2.2 (Gandomi and Haider, 2015). *Big Data is high volume, high velocity, and high variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision-making.... Other*

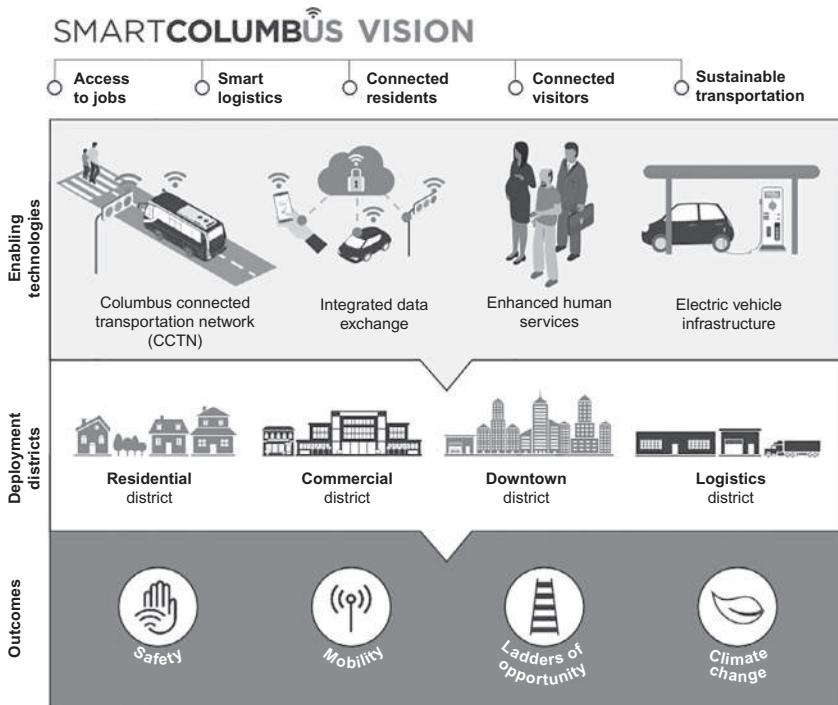


Fig. 2.3 Columbus smart city challenge implementation vision. (Source: [USDOT, 2016](#).)

dimensions of Big Data include... **veracity** (unreliability inherent in some sources of data), **variability** (variation in data flow rates), and **value** (scalability of value as data volume increases).

Bettencourt (2014) examines, at a broader level, whether these recent technological advances and Big Data fundamentally alter what is possible to achieve, or whether they just expand the existing tools further. His argument is that modern ICTs open new windows of opportunity for applying engineering solutions in cities. These solutions are driven by the fact that the overall urban planning problem is a computationally intractable “wicked problem,” but the presence of cyber-physical systems and Big Data make it possible at least to address the “knowledge” facet of the problem. New theories are needed to address the “calculation” facet by reducing the size of the problem to make applications feasible. One example is the research effort from the “statistical physics” researchers who seek common patterns in distribution and scaling of mobility (Gonzalez et al., 2008) and city growth (Bettencourt et al., 2007). These theories can be used to significantly simplify the computational requirements of models of urban mobility.

Therefore the concept of “supply procedure” in the classic Manheim–Florian–Gaudry (MFG) systems analysis framework introduced in Chapter 1 needs to consider a broad definition that encapsulates both the physical and the cyber/digital components. There needs to be a better grasp in defining and classifying the technology, and to envision how it integrates with transport demand in the MFG sense as well as in other smart cities functions.

For a sufficiently broad definition of all transportation and related technologies within a smart cities context, the term “Internet of Things” (IoT) is used. The following definition of IoT is adopted from Zanella et al. (2014).

Definition 2.3 (Zanella et al., 2014). *The Internet of Things is a communication paradigm that envisions a near future in which the objects of everyday life will be equipped with microcontrollers, transceivers for digital communication, and suitable protocol stacks that will make them able to communicate with one another and with the users, becoming an integral part of the Internet.*

While this definition only focuses on communication, the paradigm deals with connected things. Certainly many of the data sources and sensors do not yet fall under such a category, but it represents the idealized scenario in an idealized smart city environment. For the purposes of this book, the technological setting aims toward this IoT paradigm.

The transport system technology therefore cannot simply be represented by infrastructure. An illustration is shown in Fig. 2.4 (Vilajosana et al., 2013)

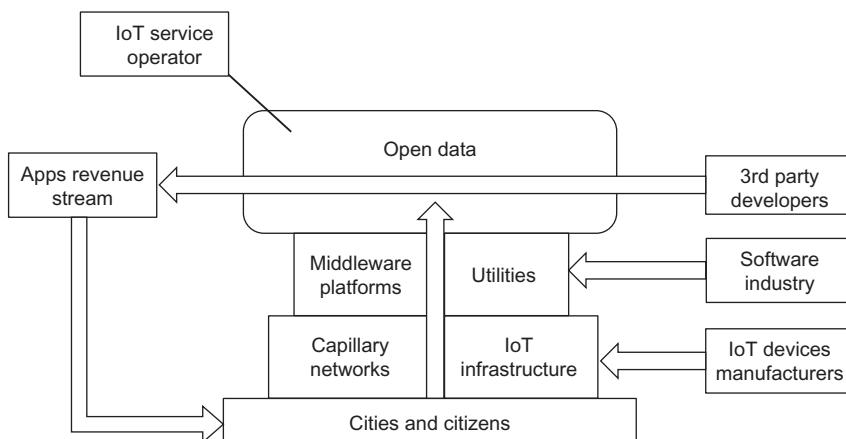


Fig. 2.4 Illustration of connecting IoT infrastructure with cities, users, operators, and software developers.

of how an IoT-based infrastructure technology may integrate with other smart city functions. Not only does analysis of urban transport systems require expanding the view of the systems side to include IoT, Big Data, software, and operators, but the process of operating such systems in a smart city needs to also consider all these elements.

At a more direct level, the information that is now available in cyber-physical transport systems requires new theories and tools for extracting and interpreting it. The following sections discuss the evolution and tools for monitoring mobility in smart cities.

2.3 MONITORING MOBILITY

At the heart of the interface between the rise of smart cities and the Internet of Things is the role of sensing or monitoring (Perera et al., 2014). There are generally two categories of monitoring with respect to mobility: vehicular traffic monitoring and traveler activity monitoring. Vehicular traffic monitoring deals with sensing attributes related to the interface between vehicular flows and the infrastructure: for example, speeds, densities, volumes, and derivative measures like travel time. Such monitoring has been around for many decades because of the observability of such data using sensors like inductive loop detectors, video, radar, and onboard GPS. With more recent technologies, such data is even more observable: LIDAR, drones (Chow, 2016b), Bluetooth, cellphones, satellite, and other mobile devices (Harvey et al., 2016). Many real-time and operational strategies can thus be employed using information monitored in this way: short-term traffic prediction (Vlahogianni et al., 2014), incident management, and infrastructure performance monitoring. One example is how traffic monitoring has changed from “Eulerian” information to “Lagrangian” information (Work and Bayen, 2008) because of the availability of GPS-enabled mobile phones, as experimented in the Mobile Millennium/Century project discussed in Herrera et al. (2010). Eulerian information is control volume-based information, whereas Lagrangian information follows individual probe sensors over a trajectory in time.

The second category of traveler activity monitoring is far more difficult. Until the development of mobile devices, passive observational data of traveler mobility patterns throughout the day did not exist. Even with mobile devices, this observation is limited to geospatial information and lacking in activity/travel purpose information such as whether a trip is made because the person is dropping someone else off or if they are going shopping instead

of getting a haircut. The lack of such data confined the analysis of traveler activity patterns to long-term planning purposes, as household travel surveys and travel diaries are time consuming to conduct, tend to be limited to cross-sectional data, and are limited in sample size. Transportation agencies would conduct such surveys only every several years as a result and insights drawn from the data can only be applied to long-term planning purposes. However, operational analysis is important for understanding the impact of changes to the transportation system on travelers.

Vehicular operational data, while abundant, lacks the direct relationship between transportation system changes and travel demand and activity patterns. Since it is the underlying travel and economic welfare of the public that is primarily of concern, this missing link has made it hard for the transportation profession to monitor impacts on travel. For example, meteorologists have direct sensors on weather patterns and can thus provide evaluations of storms on the public at any given time (e.g., through measures like “inches of rain”) to make 7-day predictions. Transportation, on the other hand, has been limited to monitoring road traffic volumes. For example, they can determine the effect of a baseball game on travel speeds and densities throughout the road network, but they cannot determine how that same game increases individuals’ travel times or reduce participation in other activities.

With the rise of Big Data and IoT, passive traveler activity monitoring is much more feasible. For example, [Çolak et al. \(2015\)](#) illustrate how CDR data can be used to replicate many of the conventional transportation planning models like trip generation or trip distribution. Several uses are now possible for traveler activity monitoring that were only possible with vehicular monitoring in the past.

- Quantify the impact of a change to the transportation system on social welfare of different subpopulations
- Use the data to learn latent system state parameters (see [Chapter 5](#)) in real time
- Estimate longitudinal forecast models of the traveler activity patterns that account for temporal dependencies
- Use as input for dynamic policies or decision processes, for example, vehicle dispatch, headway control, surge pricing

One unique data source that requires special attention is taxi GPS data (see [Yue et al., 2014](#), for a comprehensive survey). For planning purposes, taxi GPS data can be used to mine origin-destination patterns, time of day variations for specific locations, hotspots, trip distances, and so on ([Yue et al., 2009](#);

Zheng et al., 2011; Huang et al., 2015; Tang et al., 2015; Mao et al., 2016; Shen et al., 2017); for clustering land use types (Liu et al., 2015b); and for identifying critical locations or quantifying resilience (Zhou et al., 2015; Zhu et al., 2017). Unlike passenger cars, taxis have a much more direct linkage to travelers' activity patterns (where and when they get picked up and dropped off), and unlike public transit, the patterns are not aggregated over multiple users since it is typically one passenger per taxi trip. This means that tools developed to monitor vehicular trajectories can better infer traveler activity patterns from taxi GPS data than other vehicular data (e.g., from traffic management centers).

A few state-of-the-art methods exist for traveler activity monitoring. One of them is based on the analogy put forth by Park et al. (1984) that a city is similar to a human body. Processes occurring within a city can be measured and observed as "heart beats" forming a "pulse." Miranda et al. (2017) demonstrate the use of this with data from Flickr activity, as illustrated in Fig. 2.5.

In the figure, two distinctly different locations are compared using the same Flickr data source: Rockefeller Center in NYC and Alcatraz Island in San Francisco. The two locations are compared at hourly and monthly resolutions. While they each share similarities within the same resolution, the two locations also have distinctly different sequences of beats leading to distinctive pulses based on Flickr data. An urban pulse is a computationally cheap approach to describe a location based on activity data. For example, taxi pickup data at transit stations in NYC can be used to provide a pulse of

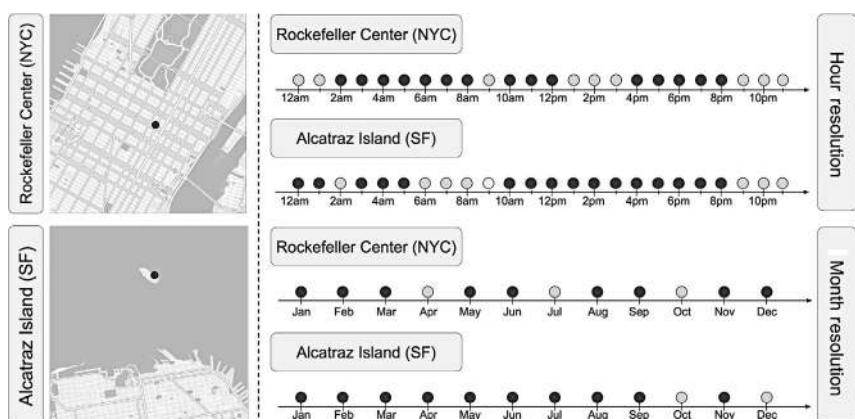


Fig. 2.5 Illustration of beats that form pulses unique to different locations in a city depending on the type of activity data used. (Source: Miranda et al., 2017.)

last mile traveler activity. One can assume that there is a representative weekday pulse for one transit station, which may differ from that of another transit station in the same city. When an extreme event occurs, such as a major storm or a planned event, it is possible to use the disruption in the pulse to quantify the effect of the event. Analogously, a new role for “city doctors” can be defined for monitoring the health of a city by periodically checking the pulse of different locations and activities to detect abnormalities.

Miranda et al. (2017) define a pulse P as follows.

Definition 2.4 (Miranda et al., 2017). A [urban] pulse is formally defined as a pair $P = (L, B)$, where $L = (x, y)$ denotes the location of the activity, and B represents the beats that summarize the variation of that activity over different temporal resolutions at the specified location.

Beats can be monitored as binary signals (“significant or maxima beat”) or as a “function beat” where the value is simply a scalar function at the location. Typical scalar functions may be Gaussian weighted sum density functions. Pulses can be taken at different temporal resolutions, such as hourly or monthly.

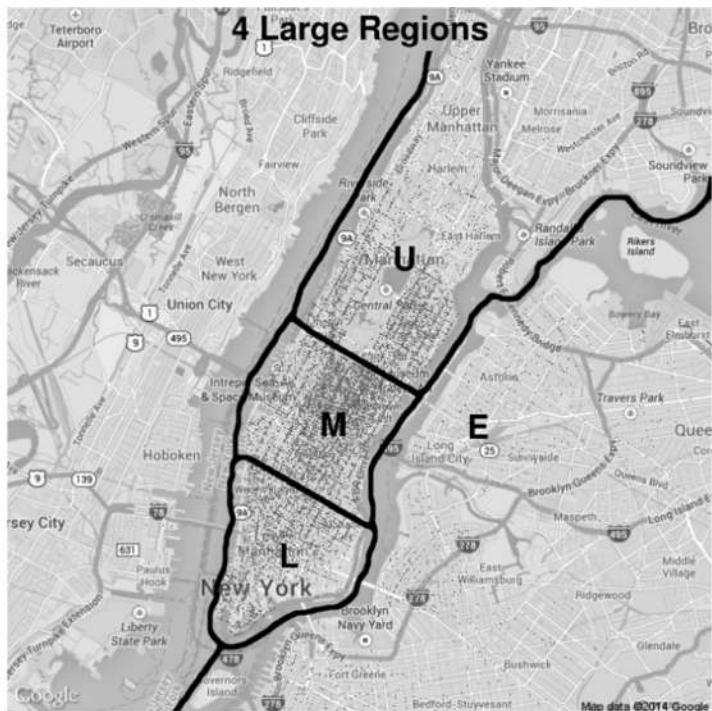
While urban pulses are convenient for capturing activity data at specific locations, for mobility purposes they lack information about change in space over time. As a result, urban pulses may not provide a clear relationship between transport systems and user activity patterns. A second state-of-the-art monitoring methodology is based on Donovan and Work (2017). They propose using taxi trajectory data and measuring a value of “pace” $P_{i,j,t}$ from zone i to zone j at time interval t as the inverse of a generalized average speed on a single road segment, as shown in Eq. (2.1).

$$P_{i,j,t} = \frac{\sum_{r \in T_{i,j,t}} u_r}{\sum_{r \in T_{i,j,t}} l_r} \quad (2.1)$$

where u_r is the travel time of trip $r \in T_{i,j,t}$, l_r is the metered length of trip r , and $T_{i,j,t}$ is the set of all trips from zone i to zone j that started after hour t . For all OD pairs R where $|R| = k^2$, a vector of mean paces can be derived as a_t , where each n th element $a_{t,n}$ is shown in Eq. (2.2).

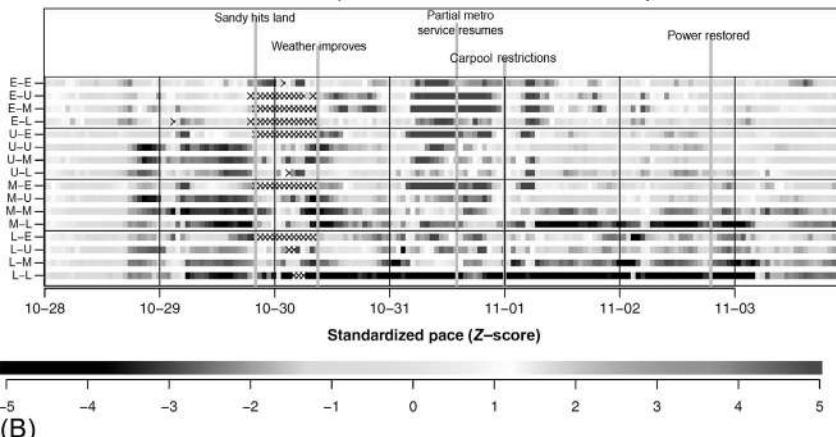
$$a_{t,n} = P \left[\left\lfloor \frac{n}{k} \right\rfloor, n \bmod k, t \right] \quad (2.2)$$

where $n \in \{0, 1, 2, \dots, k^2 - 1\}$. The mean pace vector can be extracted over time to obtain a temporal profile. This is illustrated in Fig. 2.6B extracted



(A)

Standardized pace over time—week of Hurricane Sandy



(B)

Fig. 2.6 Using a (A) 4-region representation of NYC and (B) illustration of a standardized pace vector progressing over time before, during, and after Hurricane Sandy. (Source: *Donovan and Work, 2017*.)

from taxi data during the week of Hurricane Sandy. Several major events are noted on this temporal profile. The date when Sandy makes landfall is noted on October 29th and the weather does not improve until October 30th. Partial metro service from the New York Metropolitan Transportation Authority (MTA) resumes on October 31st. November 1st sees a citywide policy to require carpools entering Manhattan via the bridges to reduce congestion impacts. Power is fully restored by evening of November 2nd. From October 28 to November 3, one can see how the pace changes across the vector of different zones defined from Fig. 2.6A.

The difference between the urban pulse concept and the pace concept is that the former only characterizes a location while the latter characterizes the ability of a transport system to move people through space. However, neither method monitors the impact of changes to a transport system on the full extent of traveler responses. For example, a positive event like a concert held in downtown should be recognized as generating economic activity at the expense of (1) added traffic delays due to the congestion and (2) scheduling impacts of travelers who may have to drop off family, work earlier, give up going shopping that day, and so on. The urban pulse would see a change in the pulse at that location, but would not recognize whether that change represents positive or negative socioeconomic activity. The change in taxi pace would recognize the slowdown in traffic speeds and added trips to and from that location, but would not recognize that people are at that location for 2 or more hours generating socioeconomic activity. Furthermore, the taxi pace method currently requires offline computation because it must run selection queries (the “loop over all trips” in Algorithm 1 in [Donovan and Work, 2017](#)) from the trajectory data that makes it ineffective to operate in an online setting. A third method is thus presented that can capture all these capabilities in an online setting.

2.4 TIME GEOGRAPHY

A third approach keeps track of movements through space and time. Doing so would lead to a higher resolution database that can account for urban pulses, pace of transport systems, and recognize the presence of socioeconomic activities. The analysis of peoples’ movement through space and time of day is known as time geography. A time-geographic space has three dimensions: the x - and y -axes form the physical space and the z -axis forms the time of day. [Hägerstrand \(1970\)](#) first introduced the concept of analyzing people’s travel patterns throughout the day with “activity prisms” defining

the limits of travel in time-space. By examining where people can travel and where they do travel throughout the day, one can derive relationships between the spatial-temporal constraints defined by the built environment and land use with the mobility and accessibility options available to people. An illustration of Hägerstrand's original time-space activity prism is shown in Fig. 2.7A.

For example, the collection of activity prisms of households in Los Angeles will differ from those collected in New York City, and even the accessibility of different sociodemographic groups can be distinguished. Kwan and Lee (2004) demonstrate this by visually distinguishing differences in distributions of space-time paths for African and Asian Americans in Portland as shown in Fig. 2.7B. Time-geographic GIS is in essence a powerful tool for researchers to quantify the socioeconomic effects that the transport and built environment have on different populations. This paved the way for a rich literature in time geography, including many studies that use GIS to visualize time-geographic data: for example, Miller (1991), Golledge et al. (1994), Kwan (2000), Pendyala et al. (2002), Neutens et al. (2008), Chen et al. (2011), and Goodchild (2013).

Due to the increasing availability of Lagrangian information like GPS data sources, in recent years there is increasing recognition that time geography can be used with real-time data to monitor mobility and measure changes in socioeconomic activity patterns. As Donovan and Work (2017) illustrated, the transport system can be susceptible to external shocks, and as a result can impact the underlying socioeconomic activities of the population. Direct measurement of the effects of these shocks was impossible in the past, as there was no direct measurable relationship between, say, a change in a road capacity or a transit service schedule and the resulting activity participation of the population. It is this reason why transport models that deal with travel demand have historically focused on only long-range strategic planning applications. The only available observable data for many years has been traffic data observations of vehicle movements, such as from loop detectors. However, vehicle data is only a surrogate of the underlying human activity data. This is ultimately the significance of all the data sources shown in Fig. 2.1. *In today's smart cities, we have the technology available to directly measure and monitor economic activities in real time.*

To accomplish this, a time-geographic tool is needed to store and visualize all the information. Earlier forms of time-geographic GIS and activity prisms have certain limitations. Although densities or prisms based off GPS trajectories can be generated, the queries needed to obtain the information

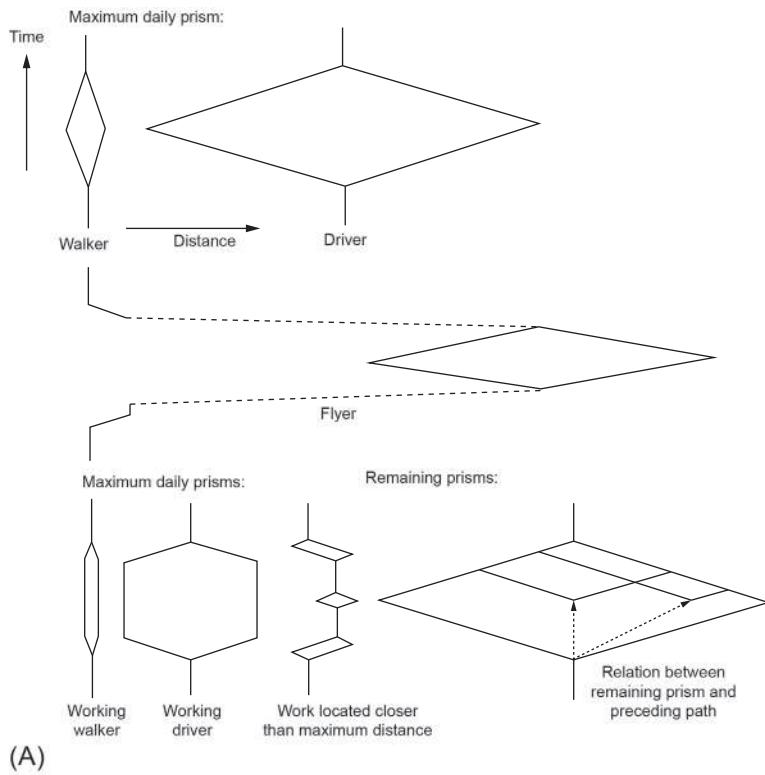


Fig. 2.7 (A) Original illustration of activity prisms and (B) demonstration of space-time aquarium of space-time paths of African and Asian Americans in Portland. ((A) Source: Hägerstrand, 1970; (B) source: Kwan and Lee, 2004.)

for an online setting can be computationally prohibitive. As a result, we have not seen much online monitoring using time–geographic tools yet. Activity prisms on their own are poor fits for population space-time constraints (Pendyala et al., 2002). This is because the constraints are highly location specific and dependent on individual scheduling preferences. A person living in one neighborhood who leaves early for work would have a very different prism than another living in another neighborhood who goes to school. Even at the individual level, however, the prisms have historically been constructed using static and constant information such as free flow travel times that do not change over space–time. So as more trajectory data became available, researchers did not initially have tools to customize prisms to visualize these patterns.

Researchers have sought to address this latter point using nonlinear vector fields. A d -dimensional vector field is a continuum in a \mathbb{R}^d space defined by a vector with both magnitude and direction. One example is a velocity field, where the speed and direction can change continuously from one point in space to another. This can represent, for example, population travel from the edge of a city to the center, where increasing congestion may lead to reduction in average speed and even a change in direction as travelers spread out or converge at different locations. The study of velocity fields in an urban setting began in the early 1970s. One such study is that of Angel and Hyman (1970), who derived 2D velocity fields for different cities. An example is shown in Fig. 2.8 for the city of Manchester in 1965, based on a point located 2.5 miles east of the city center.

A vector field drawn directly from a data source is a very powerful tool. One can simply apply vector calculus to, for example, aggregate effects of

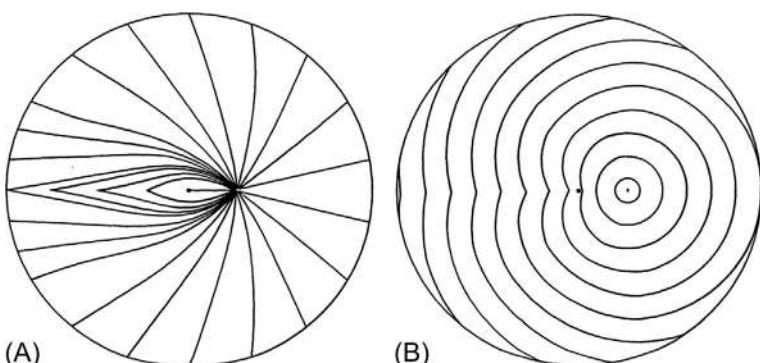


Fig. 2.8 (A) Minimum paths and (B) velocity field isochrones for a point 2.5 miles east of Manchester city center in 1965. (Source: Angel and Hyman, 1970.)

state changes, compare two different states, or determine relationships between objects in the space with the field, *all without resorting to travel forecast models*. For example, [Angel and Hyman \(1970\)](#) suggested that if further road investments were made in Manchester, the vector field would then be augmented. If the data was provided to measure the before and after vector fields, then a vector difference can capture the state change, and an integral over a region can aggregate the effect there.

Although vector fields were investigated decades ago, the technology that would most benefit from it was not yet mature enough to realize its potential. As a result, the theory remained in 2D space for planning purposes only. Two recent breakthroughs have made it possible to apply vector fields to monitor travel patterns in an online, real-time setting.

The first is the nonlinear space-time prisms from [Miller and Bridwell \(2009\)](#). They acknowledged the anisotropic characteristic of travel cost fields in operational setting. Using [Puu and Beckmann's \(1999\)](#) continuous space concept, one can obtain the cost C_p of a path p from point t_i to point t_j as a path integral as shown in Eq. (2.3).

$$C_p[t_i, t_j] = \int_{t_i}^{t_j} k[x(t), x'(t)] \|x'(t)\| dt \quad (2.3)$$

where x is the location vector represented by two dimensions (x, y), x' is the velocity vector, and $k[x(t), x'(t)]$ is a direction-specific cost function at time t . An illustration of this path construction is shown in [Fig. 2.9](#) from [Miller and Bridwell \(2009\)](#).

An anisotropic space-time prism relative to a given point k can therefore be constructed by integrating an inverse velocity field ν^{-1} along the shortest cost path P_{kl}^* from k to l . Note that inverse velocity is also equivalent to a vector form of the pace quantity defined in [Section 2.4](#), so this representation is a continuous, vector generalization of the measure from [Donovan and Work \(2017\)](#). The minimum travel time t_{kl}^* is determined from Eq. (2.4).

$$t_{kl}^* = \int_{P_{kl}^*} \nu^{-1}[x, x'] dx \quad (2.4)$$

An illustration of an anisotropic space-time prism drawn from pace vector fields is shown in [Fig. 2.10](#) from [Miller and Bridwell \(2009\)](#), which is constructed for a lunch time activity in downtown Salt Lake City.

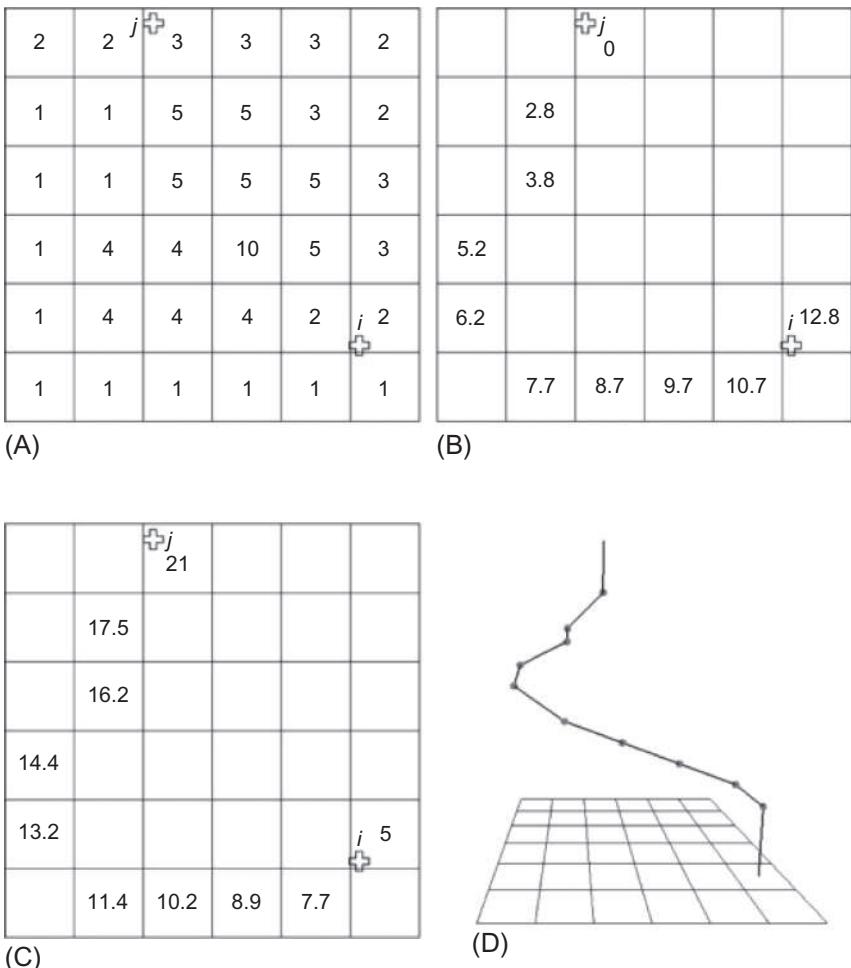


Fig. 2.9 Constructing a least-cost path from a cost vector field. (A) Initialization, (B) least cost path with accumulated cost to j , (C) temporally referenced least cost path, and (D) extracted path as a space-time polyline. (Source: [Miller and Bridwell, 2009](#).)

Since the cost and pace vector fields are stored a priori, an online query is possible using this method. In addition, the method can handle dynamic travel costs over space and time. The only drawback is that the method is applied only to an individual level. The second recent breakthrough by [Liu et al. \(2015c\)](#) addresses that issue by extending the methodology to population level. The remaining sections in this chapter discuss that method.

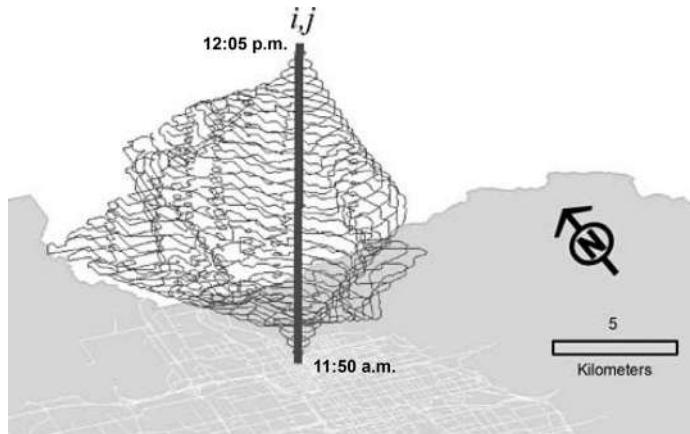


Fig. 2.10 Illustration of a field-based space-time prism from [Miller and Bridwell \(2009\)](#).

2.5 TRAVEL MOMENTUM FIELDS (TMFs)

In this section we introduce the concept of travel momentum fields (TMFs), which has all the advantages of the method from [Miller and Bridwell \(2009\)](#), but is also applicable to monitoring population data.

[Liu et al. \(2015c\)](#) present a methodology to capture, at the population level, a vector field of the “travel momentum.” First, consider a different kind of “cost” field for an individual n defined by the velocity of that individual at that location $k[x_n(t), x'_n(t)] \equiv (x'_{nx}(t), x'_{ny}(t))$. It is not a real travel cost, but simply a measure of an individual’s ability to move through the time-geographic space. This vector should always be positive with respect to time, and a higher speed would be reflected by a larger magnitude vector given the same interval in time, as shown in [Fig. 2.11](#) where the $x = (x, y)$

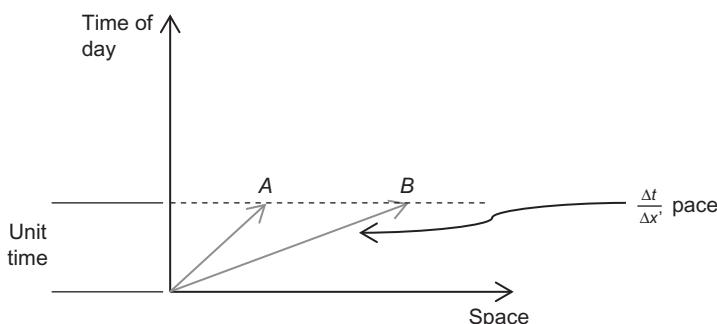


Fig. 2.11 Illustration of two different individual velocity vectors.

coordinates are collapsed into a single dimension for simplicity. In the figure, the length of vector \mathbf{A} is shorter than that of vector \mathbf{B} because $x'_B > x'_A$, leading to more distance covered in the same unit time. The slope of the vector (change in time Δt over change in space Δx) along the time dimension is the pace.

To obtain a vector field $K[x(t), x'(t)]$ corresponding to the population N of activity at any point x in time t , we can simply do a vector sum as shown in Eq. (2.5). A vector sum is illustrated in [Exercise 2.1](#).

$$K[x(t), x'(t)] = \sum_{n \in N} k[x_n(t), x'_n(t)] \quad (2.5)$$

Exercise 2.1

Given three individuals at the same location in time-geographic space (x, y, t) with velocity vectors $(0, 1)$, $(1, 1)$, $(1, -2)$ in unit time, determine the population vector at that location.

This is simply a vector summation: along the x -axis, $0 + 1 + 1 = 2$; along the y -axis, $1 + 1 - 2 = 0$. This leads to a population vector of $(2, 0)$. Note that the time dimension is not added.

Recall from physics the term momentum refers to the product of mass and velocity. In this urban mobility context, population is the mass, and therefore $K[x, x']$ is a momentum vector field. We call this field a “travel momentum field” as it reflects the momentum of the population to move in a certain direction at any point in the time-geographic space.

Definition 2.5 $K[x, x']$ is a **travel momentum field** that defines a vector at any location x and time t in the time-geographic space representing the sum of every individual's velocity vectors x'_n at that same location.

There are several characteristics of this vector field that makes it a valuable methodology for monitoring urban mobility. First, this is simply a data representation, not a forecast model in the traditional transportation planning sense. While the field is an aggregation of speed and volume (one drawback), it is a straightforward expression of the underlying travel patterns that is not clouded by structures assumed by models.

Second, the field can be updated incrementally for each unit time in an online setting. Once updated, the field is prestored data that can be easily

queried in real time using simple vector operations. On the contrary, a spatial query using a large set of GPS trajectories would require searching through each trajectory one by one. Empirical tests by Liu et al. (2018) using taxi GPS data in Beijing suggest the vector field method can reduce computational time by 20 times.

Third, transport systems can also be represented as vector fields, and therefore we can draw direct relationships between the transport system and the population TMF as they change over time. This is very powerful for monitoring purposes as it can perform before-after analysis without resorting to travel behavior or network flow models of any kind.

Fourth, the momentum concept is highly analogous to the concept of flow. Whereas flow is the scalar product of speed and density, the momentum field is a vector formed by the product of velocities and the volume over a point in space. In other words, travel momentum is equivalent to a flow vector.

In practice we will unlikely ever possess all of a population's travel trajectory data. However, there are well-established techniques to extrapolate from a spatial sample to obtain a population-level aggregation using kernel density functions. Liu has developed an open source code for estimating kernel densities from sample trajectory data and made it available at this link: <https://github.com/xiantao/3DKernel>. Lastly, note that in Liu et al. (2015c) the terminology refers to the TMF as a demand, but simply calling it travel momentum is more appropriate as flow is only realized demand.

Before demonstrating the different uses of TMFs, we illustrate what this method can do. We obtain OD patterns from 2011 household travel survey data from the Greater Toronto Area as depicted in Fig. 2.12A. Since actual trajectory data are not available in this data set, they are synthesized with simple straight paths connecting origin to destination to illustrate the methodology. Based on this "trajectory" data, a traditional visualization as per Kwan and Lee (2004), for example, would produce a density over space for a specific time interval as shown in Fig. 2.12B. This is a map of the kernel density of the trajectories at the 8 a.m. time interval. What our TMF can visualize is shown in Fig. 2.12C and D. Fig. 2.12C illustrates the TMF at the same 8 a.m. time interval shown in Fig. 2.12B. There is direction information stored in this visualization, indicating a momentum toward downtown Toronto at that time in the day. In Fig. 2.12D, an isometric view of the visualization in Fig. 2.12C shows the magnitude along with the directionality. While most of the activity occurs in the downtown Toronto area, peaks representing Hamilton and Oshawa can also be identified.

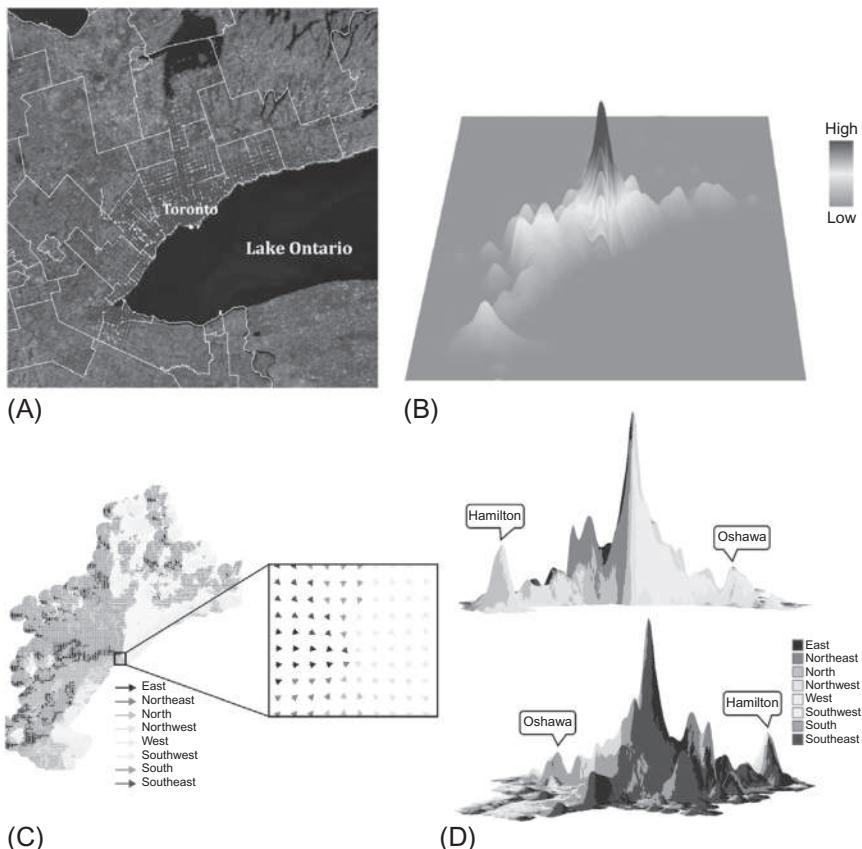


Fig. 2.12 (A) Household travel survey data from Greater Toronto Area; (B) kernel density map of the trajectories at 8 a.m.; (C) travel momentum field at 8 a.m.; (D) isometric views of travel momentum field in (C). (*Source: Liu et al., 2015c.*)

Let us demonstrate the method using a simple example in [Exercise 2.2](#).

Exercise 2.2

Construct a TMF from the trajectories shown in Fig. 2.13.

For simplicity we represent the TMF through the same four time slices. First, we approximate the velocity vectors of each individual in each time slice by taking the difference between (x_n, y_n, t) and $(x_n, y_n, t+1)$ (since change in time is always 1, we leave that out). Time 4 is left out since we do not have the location information at time 5.

Trajectory 1:

Time 1		
	(2,1)	

Time 2		
		(0,0)

Time 3		
		(-2,-1)

For the other three trajectories, we have the following.

Trajectory 2:

Time 1		
(-1,1)		

Time 2		
(1,1)		

Time 3		
		(0,-1)

Trajectory 3:

Time 1		
		(1,0)

Time 2		
(0.5,-1)		

Time 3		
		(0.5,-1)

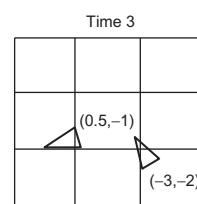
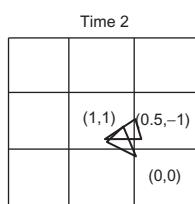
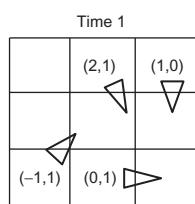
Trajectory 4:

Time 1		
(0,1)		

Time 2		
(0,0)		

Time 3		
		(-1,0)

The TMF for the three time slices are shown as follows.



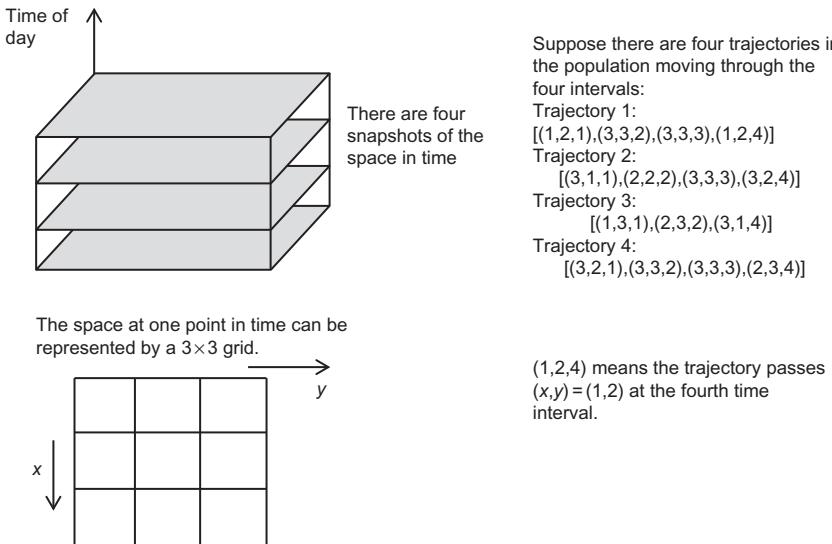


Fig. 2.13 Example for Exercise 2.2.

The data is now stored in a vector field structure. This field can be visualized as shown in Fig. 2.12 with color coding, for example. In the exercise here, we see momentum of the population to converge toward $(3,3)$, but by time 3 there is a momentum to move out of that spot. Also, this exercise does not include population aggregation from sample trajectories. Readers can refer to Liu et al. (2015c) for more details on that.

The methodology has been applied to monitoring taxi GPS data in Beijing (Liu et al., 2018). Since taxis are single-ride trips from origin to destination, they are a good surrogate of taxi user population travel momentum. Tracking the movement of public transport vehicles would lose the last mile portions of user mobility while tracking private passenger vehicles misses out on the drivers' movements after parking their vehicle.

GPS data from 12,000 taxis operating in Beijing from November 2nd to 5th in 2012 was collected for this demonstration. In each minute, the GPS-enabled device installed on each taxi automatically sends its current location (i.e., latitude, longitude) and other attributes (e.g., taxi ID, timestamp, operation status, speed, direction and GPS status, etc.) to a centralized server. There are approximately $12,000 \times 24 \times 60 = 17,280,000$ GPS points per day. The trajectory of a taxi is composed of a sequence of such time-stamped GPS points in chronological order. Sample data from Friday, November 2nd, 2012, is shown in Fig. 2.14.

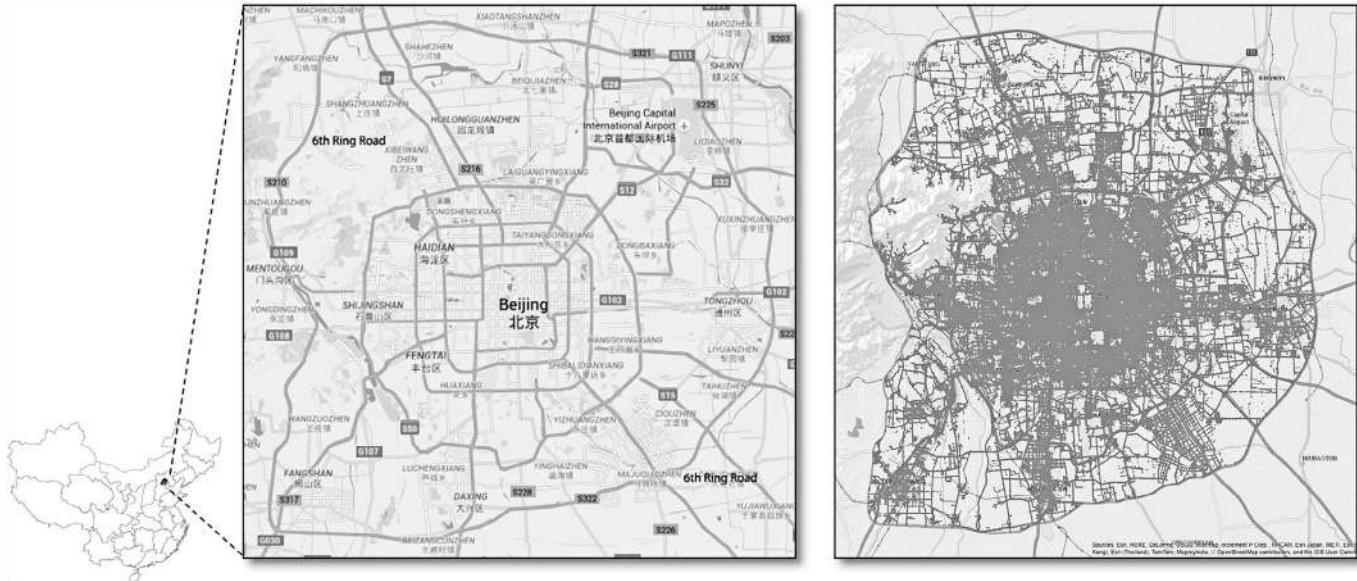


Fig. 2.14 (left) Beijing, China, and (right) total daily real-time GPS locations from 12,000 taxis on November 2, 2012. (Source: Liu et al., 2018.)

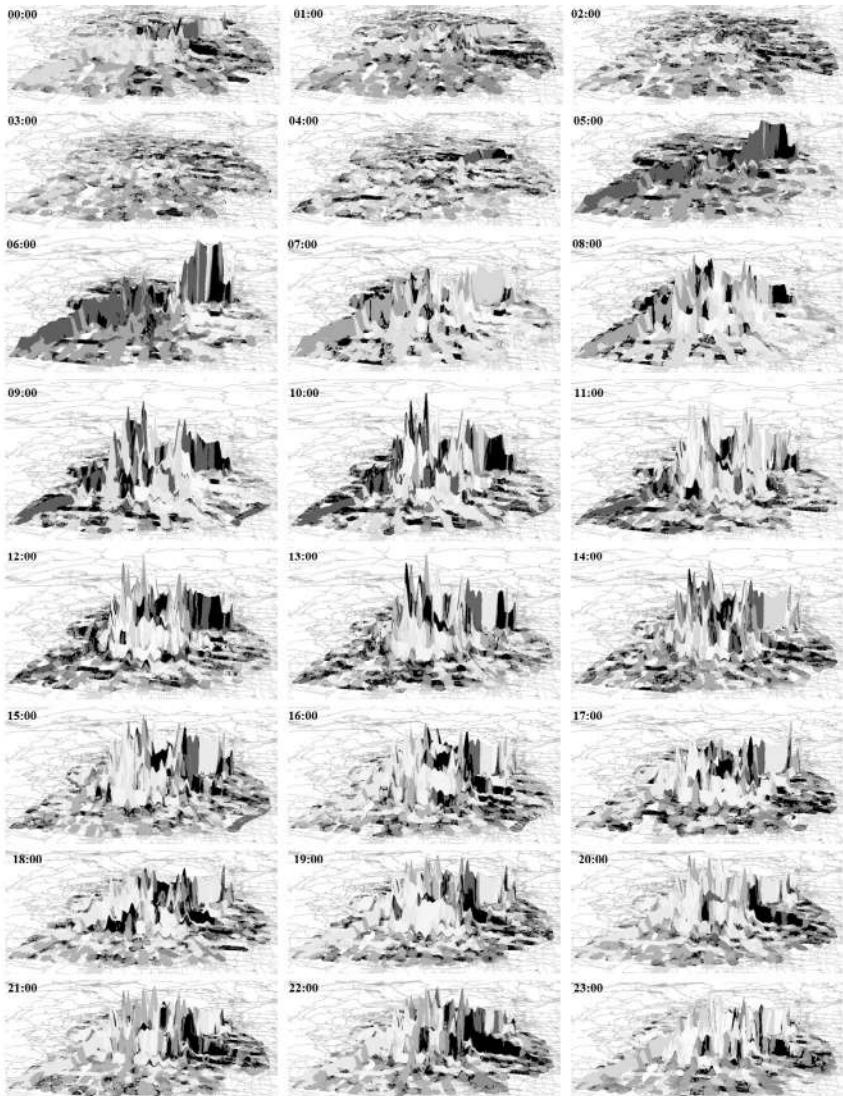


Fig. 2.15 Snapshots of kernel density estimated taxi population TMF in 24 h. (Source: [Liu et al., 2018](#).)

The TMF over hourly snapshots in a 24-h period is shown in Fig. 2.15.

The height measures the magnitude of taxi density while the color refers to different directions of travel momentum. One can evaluate how overall taxi travel momentum evolves in different hours. For example, at 5:00 and 6:00 in the morning, eastbound momentum is dramatically increasing along a particular corridor, whereas in the evening it is more dispersed along all directions. In fact, the peak vector kernel density area at the upper-right part of the 6:00 subgraph is the international airport. Therefore the physical

meaning behind this anomaly is how travel momentum at 5:00 and 6:00 a.m. is distributed and related to the airport. As time goes to 7:00 a.m., travel momentum starts to shift to central areas in Beijing.

2.6 TMF: TRANSPORT ROUTE PROJECTIONS

While the TMF data structure produces a nice visualization, the main value of the methodology comes from the online applications. Spatial queries performed with this preconstructed data structure can be much faster and can be used for real-time operations. Prior to the development of this data structure, it would be too cumbersome to query large data sets in real time. That is why many of the GIS studies related to GPS data relate to offline analysis. With this capability, one can truly monitor minute by minute how the momentum of the population coincides with existing infrastructure or subareas, or how to quantify changes in state using these fields.

The first application to explore is the concept of projection. [Miller \(1991\)](#) discussed how space-time prisms can be projected onto the 2D GIS map to identify areas of accessibility. Projection in a vector field sense is similar. Recall how a projection works: the projection v_{ab} of a vector a onto another vector b is the dot product between the first vector with the unit vector of the second, as shown in Eq. (2.6) and Fig. 2.16.

$$v_{ab} = \left(a \cdot \frac{b}{|b|} \right) \frac{b}{|b|} \quad (2.6)$$

Projection essentially relates one vector to another by quantifying the component of one that is aligned with the other. There are different ways to use this concept. The first involves projecting travel momentum onto the transport infrastructure. The TMF that we have been discussing thus far relate to the observed mobility of the population over time. A transport route can also be represented by a vector field. A route is defined broadly here; it can refer to a transit fixed route, or a mobility-on-demand route that is observed to move without a predefined route, or a roadway that facilitates

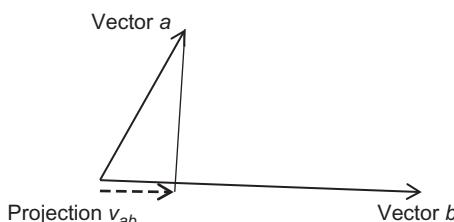


Fig. 2.16 Illustration of vector projection.

travel in a direction. For example, a Manhattan street grid may be composed of thousands of transport routes.

Definition 2.6 A transport route $r \in R$ in a time-geographic space (x, y, t) is a vector field $k[x_r(t), x'_r(t)]$ defined by the velocities x'_r corresponding to the locations x_r of that route at time t .

Then the transport route projection is defined as follows.

Definition 2.7 A transport route projection D_r is a nonnegative projection of the travel momentum field onto the transport route $r \in R$, $D_r = \langle K[x(t), x'(t)], k[x_r(t), x'_r(t)] \rangle_+$.

The nonnegative component is included because projections that lead to momentum in the opposite direction of the route would not be associated with that route.

A projection is not a prediction of ridership or assignment of passengers to a route. It is simply a data association; given that there is a route that shares the space and time of a population of users, how much of the momentum of those users align with the travel enabled by the route? The results give policymakers a performance measure that they can readily access in real time. Consider the following example in [Exercise 2.3](#).

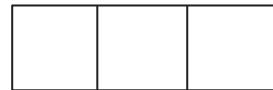
Exercise 2.3

For the TMFs shown in [Fig. 2.17](#), determine the projection onto the transport route and the aggregate route-level temporal profile over the five time steps.

There is a transport route moving from west to east with $x' = (0, 10)$ for all time t .



The TMF for this space is shown for three time intervals.



$$t=1 \quad (-9, 4) \quad (6, 12) \quad (0, 8)$$

$$t=2 \quad (-3, 7) \quad (1, -3) \quad (0, -2)$$

$$t=3 \quad (0, 3) \quad (4, 2) \quad (3, 0)$$

Fig. 2.17 TMFs for [Exercise 2.3](#).

We first determine the projected values. For each cell, we obtain the projected value. For example, $(-9, 4)$ projected onto $(0, 10)$ is $(0, 4)$. The results are shown in Fig. 2.18.

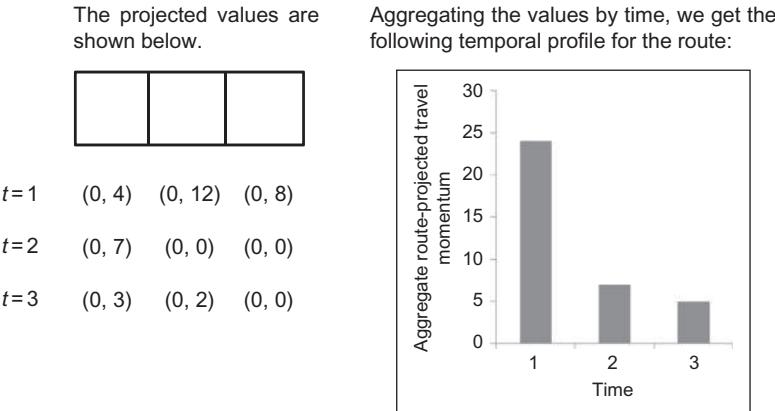


Fig. 2.18 Projected values and temporal profile.

Note that the projected TMF has opposing vectors zeroed out. It shows that the most momentum in the direction of the route during $t=1$ occurs in the middle segment. The aggregated temporal profile shows how projections on the route overall change over time. Based on the profile, we can see the momentum drastically drop from the first time step to the latter two time steps. If this is recurrent from a day-to-day basis, it may be because there is a drop in peak demand leading to this shift. If it is nonrecurrent, it may be the result of an incident that led to lower momentum (lower speeds and/or lower volumes). We can also compute the relative change from $t=1$ to $t=2$: there is a 71% drop in projected momentum, followed by another 29% drop from $t=2$ to $t=3$.

Although the projection measures are not equivalent to ridership, they form a direct data-driven relationship between observed population mobility and observed route mobility over time-geographic space. From a monitoring perspective, policymakers now have a real-time tool to assess the effects of changes.

The route projection is further illustrated using the Toronto travel survey from Fig. 2.12. A bus route (TTC bus no. 506) with scheduled operating speed is obtained and shown in Fig. 2.19.

Using the projections, we demonstrate how bus stop spacing design can be conducted in a data-driven manner. In the service of bus stop planning, the TMF route projections can be computed for the existing condition at the level of each bus stop and aggregated to get a temporal profile to show the



Fig. 2.19 Bus route no. 506 from Toronto Transit Commission. (Source: [Liu et al., 2015c](#).)

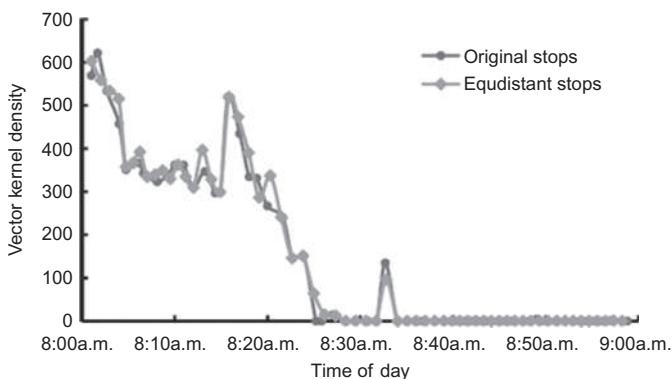


Fig. 2.20 Temporal profile of route projections of TMF on TTC bus no. 506. (Source: [Liu et al., 2015c](#).)

degree of travel momentum alignment. The result is shown in Fig. 2.20. Note that the y -axis refers to the vector kernel density since we are estimating population-level momentum based on sampled constructed trajectories. The projected momentum nearly dies down after 8:40 a.m. through 9 a.m. relative to the earlier times.

We end this section with a continuation of Exercise 2.2 in Exercise 2.4.

Exercise 2.4

For the TMF obtained in Exercise 2.2, suppose there is a transport route that moves from $(1, 2)$ down to $(3, 2)$ for all times t at a velocity of $(10, 0)$. Determine the projection on this route.

First we reiterate the TMF as shown in Fig. 2.21A. The projections that we get are shown in Fig. 2.21B.

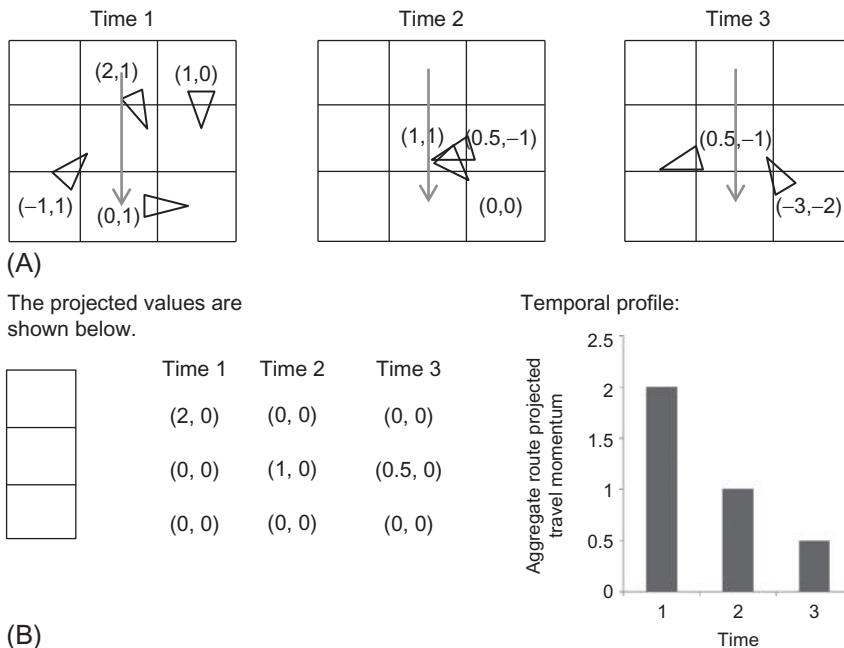


Fig. 2.21 (A) TMFs for Exercise 2.4 and (B) project values along with temporal profile.

2.7 TMF: BEFORE-AFTER ANALYSIS

The second major application is the use of data to quantify changes in the TMF as a part of a before-after analysis. This can apply to measuring the effects of external shocks such as adverse weather or large-scale disasters, or it can also be used for evaluating the change in the transport system due to a new policy or technology. Current methods to perform such analysis may be based on aggregate system measures like total system travel time, vehicle miles traveled, or total ridership. However, if the time-geographic data is made available for the population, we can identify spatial-temporal effects and isolate them for different subpopulations. This would allow policymakers to, for example, see if a new bike-share program leads to more travel momentum among lower income residents or whether longer transit service hours extends the momentum into more parts of the day.

First we define the concept of travel impulse field (TIF).

Definition 2.8 *The travel impulse field $\Delta[K_1, K_2]$ is the change in momentum from one TMF, K_1 , to another, K_2 . The impulse field is a vector field as shown*

in Eq. (2.7). The magnitude of the impulse is $\pm|\Delta|$. The change is positive if $|K_2| > |K_1|$ and negative if $|K_2| < |K_1|$.

$$\Delta[K_1, K_2] = K_2 - K_1 \quad (2.7)$$

In other words, it is as simple as taking a vector subtraction to measure the difference between two states. The impulse can be illustrated for the Toronto example. The data used in Fig. 2.12 is obtained from 2011 survey data. Since 2006 survey data is also available, it is possible to compute the TIF. An isometric view of the magnitude of the impulse for the same 8 a.m. time interval is shown in Fig. 2.22.

Fig. 2.22 suggests that in the 5-year period since 2006, morning travel momentum has shifted significantly out of the downtown Toronto area (where there is the major negative trough). Instead, growth in momentum has taken place in other areas nearby. The two biggest peaks represent the Toronto Pearson International Airport region and the Markham region. The airport is within the Region of Peel, which has significant freight and industry activity. Markham is a hub for technology companies. These changes in momentum make sense. In particular, downtown Toronto is likely experiencing a drop in momentum because of increasing congestion leading to lower travel speeds. Meanwhile the outer regions are experiencing growth in volume.

Let us now try to illustrate this before-after analysis in Exercise 2.5 for a single cell to illustrate the analytic opportunities.

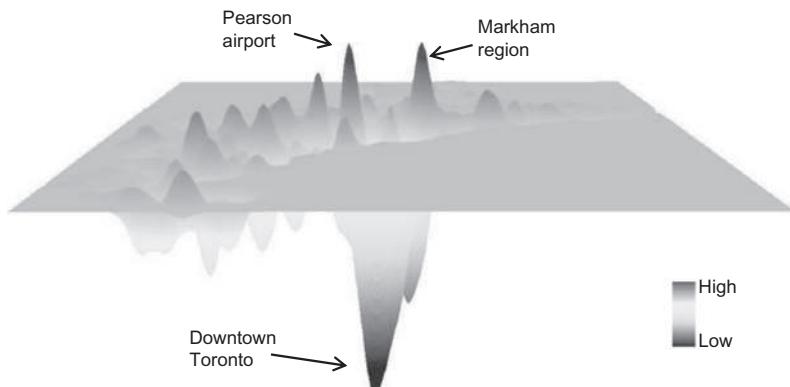


Fig. 2.22 Magnitude of travel impulse field between 2011 and 2006 TMFs at 8 a.m. (Source: Liu et al., 2015c.)

Exercise 2.5

Compute and analyze the travel impulse field $\Delta[K_1, K_2]$ where $K_1=(2, 5)$ and $K_2=(5, -10)$ for the same point in space and empty everywhere else, and for the reverse change.

In this example, the change from K_1 to K_2 indicates an increase in momentum (the size of the vector has increased from 5.385 to 11.180). This can be due to increase in velocity of travel at this location or increase in travel volume. A naive assumption might be that the magnitude of the impulse is the difference of these two scalars, 5.795. However, that is not necessarily true. While it is an increasing change in momentum, there is also a significant change in direction of the momentum. The resulting vector difference is $\Delta=(3, -15)$. In other words, while the momentum grew there was also a significant change in the momentum with a magnitude of +15.297.

If the state change was reversed, the momentum change would be negative, with a vector difference of $\Delta=(-3, 15)$, and a magnitude of -15.297.

TIFs can change the way we communicate transportation system effects to the public and to policymakers. Currently, daily traffic reports talk about which highway segments might have slower traffic or bottlenecks due to congestion. When there are major storms or when a bridge collapses, there is no direct way to quantify the change on population travel patterns due to the change in the transport system. Existing methods are limited to vehicular spot speeds, which do not provide a comprehensive picture of the effect. With this state-of-the-art methodology, it is possible to have daily traffic reports queried by the public in real time. Policymakers can see that a technology, policy, or environmental change can impact travel momentum positively or negatively, by how much, and for whom.

RESEARCH AND DESIGN CHALLENGES

The following are open-ended challenges that sometimes refer to real data sets. Readers are encouraged to try to tackle the problems, and post their designs or solutions on the following GitHub page, <https://github.com/BUILTNYU/Elsevierbook/tree/master/Chapter%202>.

- (2.1)** Pick your hometown and conduct a survey of components of smart cities that it exhibits and those that it lacks. Sketch out a conceptual

architecture of a smart cities process (Fig. 2.4) and propose two new apps that would improve mobility to address the gaps.

- (2.2) Take the NYC taxi pickup data (http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml) for one arbitrary day and query the pickups over time for two locations: Penn Station and JFK Airport. Using the Gaussian density function discussed in [Miranda et al. \(2017\)](#), construct a plot of function beats and significant beats. Compare the results. Make an online app that can be fed taxi data to generate the pulse dynamically.
- (2.3) Go to the source code for Donovan and Work's study: <https://github.com/Lab-Work/gpsresilience>. Replicate their results using the same taxi data (or use a new taxi data source) and discuss.
- (2.4) Consider a household located at zip code 10012 in NYC. If this household has to arrive at work at zip code 11201 by 9 a.m. and depart from home no earlier than 6 a.m., use a Google Maps API (e.g., <https://github.com/BUILTNYU/Google-Map-API-Query-Program-and-Documentation>) to determine nearby travel times for zip codes and create a morning commute time-space prism for driving, walking, and public transit.
- (2.5) Pick a traffic analysis zone in NYC (e.g., <https://catalog.data.gov/dataset/tiger-line-shapefile-2011-2010-state-new-york-2010-census-traffic-analysis-zone-taz-state-based29d50>) and query all trips belonging to households residing in that zone from the 2010/2011 NYMTC Household Travel Survey (<https://www.nymtc.org/DATA-AND-MODELING/Travel-Surveys/2010-11-Travel-Survey>).
- Visualize these in GIS.
 - Deconstruct them and estimate the travel momentum vector field for this population using <https://github.com/xintao/3DKernel>.
 - Obtain transit trips for one line near the TAZ operating between 8 and 10 a.m. Project the vector fields onto these trips (one trip is a vehicle run from start to end of route). Create a temporal profile for each trip.
- (2.6) Take a look at the T-Drive taxi trajectory data: <https://www.microsoft.com/en-us/research/publication/t-drive-trajectory-data-sample/>.

For this data, use <https://github.com/xintao/3DKernel> to estimate a population travel momentum field for 1 day and then simulate online monitoring by updating each hour for the next day based on newly received trajectory data.

- (2.7) Create synthetic region with a population of 12 individuals. Fix the home locations of these 12 and simulate trajectories in this region for each one. Create the travel momentum field for this population. Now assume travel times take longer because of 20% slower travel speeds. Recreate new trajectories with same destinations and sequences for the 12 individuals. Construct a before-after impulse field to compare the result.
- (2.8) If traveler trajectory data is available to the reader, consider waiting for the next time there is a major adverse weather event. Track the trajectory data before, during, and after the event for a sample of trajectories and use estimate the population impulse vector fields.
- (2.9) In the T-Drive data, identify a major tourist landmark. Project the taxi trips to a single point of interest (see [Liu et al., 2018](#)). What use can this projection serve?
- (2.10) Find the NYC taxi data before, during, and after Hurricane Sandy. Since only pickup and drop-off information may be available, assume direct vectors from pickup to drop-off. Determine the impulse field for each of the day to day to study how the method could be applied to extreme event monitoring.

CHAPTER 3

Network Equilibrium Under Congestion

3.1 THE NEED TO EVALUATE CONGESTION EFFECTS

In [Chapter 1](#), a classic framework for analyzing urban transport systems is introduced. [Chapter 2](#) covers emerging technologies that significantly impact what we can do in transport analysis. Those first two chapters only provide tools to identify and measure important relationships between users and the systems. However, unlike some other disciplines, transport systems also rely significantly on models to evaluate their performance. There are three major reasons for this.

- (1) Urban transport systems, which can be highly costly to deploy and operate, require public investment and involvement. Making an investment prior to understanding its impact may lead to costly failures. Furthermore, high profile failures (e.g., Car2Go in San Diego ([Garrick, 2016](#)) or Kutsuplus in Helsinki ([Sulopuisto, 2016](#))) can make it harder for policymakers to support them in the future.
- (2) Transport technologies and policies can have very diverse effects on their users because the users can differ so significantly from one city to another or even from one neighborhood to another in the same city. This population heterogeneity makes it impossible to develop one-size-fits-all solutions. So unlike other engineered products like Apple iPhones or even buildings, there is no single building standard to follow. Therefore a service like Car2Go can succeed in a city like Seattle and still end up failing in San Diego.
- (3) Even if the ideal policy or technology is deployed to fit the make of a population, there are two properties of interaction between users and the system that will make it difficult to evaluate the system's performance.

The first is the presence of system capacities and congestion effects such that the more users sharing the system, the worse the system's performance degrades.

The second is in the complex, high-dimensional, responses of users to changes in system performances. Users can change route, departure time, mode or series of modes, destination or series of destinations, schedule of destinations, or activity agenda and duration. That does not even include interpersonal choices such as carpooling or picking up kids at school, or long-term choices such as home location, employment, and marriage.

This chapter and [Chapter 4](#) deal with the last of these reasons, as part of the market equilibration portion of the MFG framework. This chapter specifically addresses modeling methods to properly evaluate the effects that users' preferences can have on each other in a transport system represented as a network. Due to the origins of urban transport systems field from the highway expansion programs of the 1950s in the United States, much of the classic urban transport systems science deals with this first effect because policymakers need to predict flows on links in the networks that they invest in. As we see more multimodal systems, shared mobility, and Mobility-as-a-Service (MaaS) mature in cities, it is possible that the science for the activity scheduling responses of travelers will gain further in importance.

This chapter deals with material that assumes some basic knowledge of network analysis and mathematical programming. [Ahuja et al. \(1993\)](#) is a good reference for network analysis while [Bradley et al. \(1977\)](#) is an excellent source for mathematical programming. An understanding of queueing analysis is also assumed; [Appendix C](#) is provided to review this subject.

The chapter is divided into the following sections. [Section 3.2](#) deals with congestion effect in road networks that considers only travelers' decisions with complete information. [Section 3.3](#) considers congested networks where travelers' incomplete information of the system is explicitly considered through the construction of "hyperpaths." The primary application is in fixed route transit assignment. [Section 3.4](#) covers other types of congestion effects, whether it is in departure time, in cruising for parking space, or when considering the matching friction between users and taxis operating in a decentralized environment. [Section 3.5](#) examines explicit consideration of system operator decisions in a two-sided market framework.

Since only one chapter is devoted to the topic of transport network analysis under congestion, interested readers are referred to other sources for more detailed coverage: [Sheffi \(1985\)](#) for traffic assignment models, and [Cascetta \(2009\)](#) for another perspective with an introduction to day-to-day models.

3.2 USER EQUILIBRIUM IN ROAD NETWORKS

In this section we seek a model that can explain how flows of people, goods, and other commodities may distribute themselves onto a transport system. The primary concern is that users seek to travel from one location to another, but the travel time or generalized cost of travel of a component (e.g., highway segment) is dependent on the number of users assigned to that component and the resulting system performance. Users are initially assumed to have full knowledge of the collective choices made by other users (i.e., steady state) and the resulting system state, although we also consider the relaxation of that latter assumption.

3.2.1 Congestion Effect

Economists have long understood travel to be a derived demand, where mobility to a destination is primarily driven by the original demand to conduct an activity at that destination. Transport systems, as with all economic goods provided to serve a demand, are finite in quantity. The presence of flow capacities leads to an interpretation of transport systems as queues. The performance of a queue is well known: generally, the more customers arriving at a queue per unit time, the greater the delay experienced by everyone.

To understand the effect of the quantity, we consider an aggregate performance measure: total delay C . Total delay is simply the sum of the delay experienced by every user. For example, if there is a queue with one server who takes t minutes to serve a user, one queue in which the first user in is the first user out, and n users present, then the delays up to the n th passenger are $t, 2t, 3t, \dots, nt$. The total delay by observation is equal to $C = \frac{1}{2}t(n+1)n$. The marginal delay applied to the n th user is obtained from the derivative $\frac{dC}{dn}$ to obtain $\frac{dC}{dn} = \frac{1}{2}t(2n+1)$. A plot of the number of users against the marginal cost shows that it increases with a higher numbers of users. When $\frac{dC}{dn} > 0$, the system is “congested” because the presence of congestion effect exists. When $\frac{dC}{dn} = 0$, the system is uncongested.

As economists have shown, transport and other public queueing systems can be analyzed as markets in which the price of the good is the generalized travel cost and the quantity is the number of users served by the system. In this case, we can represent the market with a classic supply-demand curve as shown in Fig. 3.1. The demand function is a measure of the number of users willing to pay to use the system for a given travel cost. For a more detailed synthesis of demand functions from consumer theory and utility maximization, see Kanafani (1983). Appendix D also provides a review of demand analysis and consumer theory. In a classic centralized system, point A in

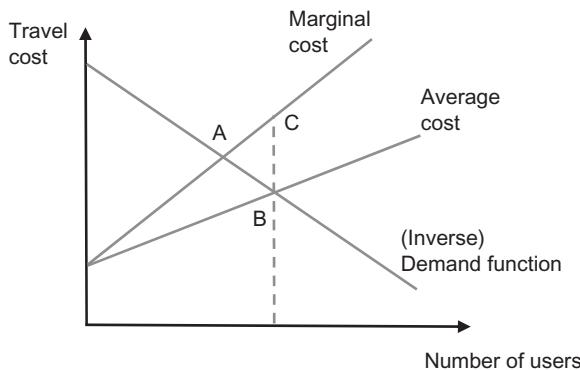


Fig. 3.1 Supply-demand curves for a transport system.

Fig. 3.1 would maximize social surplus because that is where the total benefit (area under inverse demand function) minus total cost (area under marginal cost function) is maximum.

However, transport is a *public good*. Because transport is a public good, the assumption is that there is no coordination between the users. As a result, the perceived cost for an individual user prior to entering the market, knowing only that there would be n users in the system on average (and not the order that they would enter the queue), is based on the average cost instead of the marginal cost. To better understand why this is, consider [Exercise 3.1](#) of a single server queue at a coffee shop.

Exercise 3.1

A server at a coffee shop always takes 2 min to serve a customer, and during one period there happens to be five customers who all have to wait in the queue simultaneously. What is the perceived cost of this exchange for one of those customers that period?

The cost depends on whether the person is first, second, third, and so on, upon arriving on site. The first customer would have a realized cost of 2 min. Second would be 4 min. These are each marginal costs, so accumulating them one would get the total cost. For example, for two people the total cost would be 6 min and the average over the two would be 3 min each. We plot this in [Table 3.1](#).

Because the customers are not coordinating who arrives first, second, and so forth, then the decision to enter the coffee shop would be based not on the marginal costs but on the average cost. For the five customers, each one's perceived cost for the decision is 6 min, even though the fifth person entering the queue would add 10 min to the total delay.

Table 3.1 Costs of coffee shop queue

Number of customers	Marginal cost (min)	Total cost (min)	Average cost (min)
1	2	2	2
2	4	6	3
3	6	12	4
4	8	20	5
5	10	30	6

As a result, for a public system, the equilibrium quantity in Fig. 3.1 does not correspond to point A, but rather to point B. For the n_B users who enter the system, the “last one in the queue” will incur a marginal cost of t_C . In fact, it has been shown (see Small and Verhoef, 2007) that when the cost curves are monotonically increasing with quantity, the marginal cost is always equal to or higher than the average cost. This difference is called the marginal external cost of congestion (MECC). One can derive this value for monotonically increasing cost functions. In a congested setting, the average cost for n users is defined as $t[n]$. The total cost is therefore $C = t[n]n$. The marginal cost is the derivative, which leads to $\frac{dC}{dn} = t[n] + \frac{dt}{dn}n$. The marginal cost is therefore composed of two parts: the average cost $t[n]$ and the MECC: $\frac{dt}{dn}n \geq 0$.

The MECC is a negative externality that users impose on others when they use the transport system. This means that when they make their decision to enter the market based on average cost, the remaining portion in the MECC is not internalized when compared against willingness to pay despite being borne by everyone. Average cost is a user-perceived cost for decision-making whereas marginal cost is the actual cost borne by the system.

Since point A (n_A, t_A) in Fig. 3.1 results in the optimal social surplus, then by operating at point B (n_B, t_C) the system is operating suboptimally. The loss in social surplus is the area defined by ABC. This loss is also called the Price of Anarchy, referring to the cost of decentralization.

The average cost function can be modified. If the original was some monotonically increasing function $t[n; K]$, where K is a parameter that determines the rate of increase of the function, the system operator may choose to increase $K \rightarrow K'$. This effectively pushes the equilibrium quantity higher and lowers the average travel cost. Alternatively, the operator may choose to charge all users a toll $\tau = \beta \frac{dt}{dn}n$, where β (\$/h) is a value of travel time of the users. This effectively increases the average cost function to

$t' = t[n; K] + \frac{dt}{dn}n$. If $\tau = \beta \frac{dt}{dn} \Big|_{n_a}$, the increase will set the equilibrium point at A. This is how marginal cost pricing works. A more detailed explanation of this marginal cost pricing can also be found in [Yang and Huang \(1998\)](#).

[Pigou \(1920\)](#) first examined the distribution of users over two different public goods. This is graphically illustrated in [Fig. 3.2](#). The left shows a population n of travelers from origin O to destination D given a choice between two parallel paths whose average cost functions $\{t_1, t_2\}$ and corresponding marginal cost functions $\{m_1, m_2\}$ are shown on the right. The number of users choosing path i is denoted n_i , where $n_1 + n_2 = n$. In this case, demand is fixed but the allocation on each path is not. The optimal point is where the total cost borne by all users is minimized. This is at the point A. However, as a public good, users will converge toward an equilibrium at point B ($n_{1B}, n_{2B}, t_{1C}, t_{2D}$) because the average costs are minimized there (areas under t_1 and t_2 are minimal at point B). As a result, there is a loss of social surplus of the area of ACD. If the two paths are priced according to the MECC values at point A, that would achieve an optimal path allocation. This is illustrated in [Exercise 3.2](#).

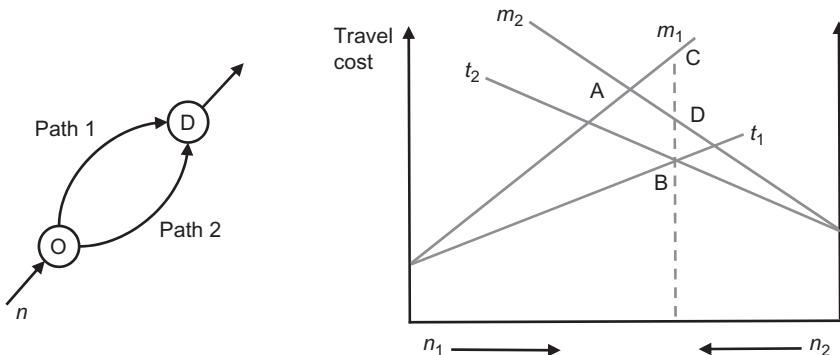


Fig. 3.2 Equilibrium allocation of travelers on two parallel routes.

Exercise 3.2

For the example in [Fig. 3.2](#), consider $n = 100$, $t_1 = 1 + 2n_1$, and $t_2 = 3 + n_2$, where travel times are in minutes. Determine the optimal tolls on each path if value of time is $\beta = \$0.33/\text{min}$.

The marginal cost functions are:

$$m_1 = \frac{d(t_1 n_1)}{n_1} = 1 + 4n_1$$

$$m_2 = \frac{d(t_2 n_2)}{n_2} = 3 + 2n_2$$

The optimum point A is where $m_1 = m_2$. Including the demand constraint, we have a system of two equations.

$$1 + 4n_1 = 3 + 2n_2$$

$$n_1 + n_2 = 100$$

This leads to:

$$\begin{aligned} 1 + 4n_1 &= 3 + 2(100 - n_1) \\ n_1 &= \frac{101}{3} = 33.67, \quad n_2 = \frac{199}{3} = 66.33 \end{aligned}$$

The average travel times are:

$$t_1 = 1 + 2(33.67) = 68.33 \text{ min}$$

$$t_2 = 3 + 66.33 = 69.33 \text{ min}$$

Tolls are:

$$\tau_1 = \beta \frac{dt_1}{dn_1} n_1 = \left(\frac{\$0.33}{\text{min}} \right) (2)(33.67) = \$22.22$$

$$\tau_2 = \$21.89$$

The tolls ensure that the congestion externalities are internalized. In addition, the people who choose path 2 end up delayed by one more minute, but they are tolled less by \$0.33 than those taking path 1 to ensure no traveler can unilaterally improve travel times by switching path.

[Wardrop \(1952\)](#) formalized this traveler behavioral assumption made by [Pigou \(1920\)](#) as a set of principles shown in [Principle 3.1](#) of users' behavior in a network of multiple congested routes. [Principle 3.1](#) assumes travelers are rational decisionmakers with full information on the network travel time functions and are making route choices simultaneously with all other travelers. The criteria refer to travelers going from the same origin to the same destination. Note that a network may have multiple OD demand, where the principles would apply for each.

Principle 3.1

[\(Wardrop, 1952\)](#). Two criteria can be used to determine the distribution of flows on routes, as follows:

- (1) The journey times on all the routes actually used are equal and less than those which would be experienced by a single vehicle on any unused route.
- (2) The average journey time is a minimum.

The first principle refers to what is called User Equilibrium (UE) behavior. It reflects selfish user decision-making such that multiple routes used for the same OD pair should have equal average travel times. On the contrary, the second principle refers to System Optimum (SO) behavior. This principle applies to a centralized decision-maker dictating which agent goes to which route.

Equilibration of users on a road transport network per the MFG framework is based on finding a set of flows that satisfies the following conditions:

- (1) One of the criteria in Principle 3.1 or similar variant (to be shown later);
- (2) Demand conservation: flow emerging from origin nodes has to equal the OD demand and the flow entering the corresponding destination nodes;
- (3) Flow conservation: all flows entering every node have to balance all the flows exiting the same nodes.

For simple networks, these criteria can be determined by establishing a system of equations to be solved. This is illustrated in [Exercise 3.3](#).

Exercise 3.3

For the network in [Fig. 3.3](#), there are two sets of OD demand, one from node 1 to node 2, and one from node 1 to node 3. The average travel times of each link are shown next to the corresponding link. Determine the UE and SO link flows.

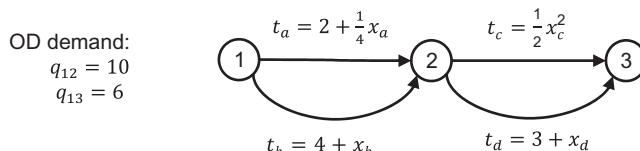


Fig. 3.3 Network for [Exercise 3.3](#).

There are eight unknowns (four link flow variables and four link travel time variables). The system of equations is as follows. For flow and demand conservation:

$$\begin{aligned} x_a + x_b &= 10 + 6 \\ x_c + x_d &= 6 \end{aligned}$$

Including the four travel time functions, that is six equations. Two more equations are needed. For UE criteria:

$$t_a = t_b, \text{ if } x_a > 0, x_b > 0$$

$$t_c = t_d, \text{ if } x_c > 0, x_d > 0$$

If $x_a = 0$, then based on Wardrop's principle the average travel times form an inequality instead: $t_a \geq t_b$. In addition, Wardrop's UE principle refers to path flows, not link flows. If we explicitly consider path flows, then we should add in four new path flow variables $\{(a, c), (a, d), (b, c), (b, d)\}$ and require them all to be equal to each other if they are traversed. Since we require $t_a = t_b$, then the constraints simplify to the link-level criteria previously. The relationship between path flows and link flows is discussed in more detail in the next section.

Assuming initially that all links are traversed, we solve the system of eight equations and obtain the following UE flows:

$$\text{UE : } x_a^* = 14.4, x_b^* = 1.6, x_c^* = 3.36, x_d^* = 2.64, t_a^* = t_b^* = 5.6, t_c^* = t_d^* = 5.64$$

The performance of the network under this behavioral assumption can be measured based on total system travel time, $TSTT = 14.4(5.6) + 1.6(5.6) + 3.36(5.64) + 2.64(5.64) = 123.45$.

How does this compare to the SO assumption? In SO behavior, total travel times are minimized. This means the marginal cost functions have to be equal. First, the marginal travel time functions are derived:

$$m_a = 2 + \frac{1}{2}x_a$$

$$m_b = 4 + 2x_b$$

$$m_c = \frac{3}{2}x_c^2$$

$$m_d = 3 + 2x_d$$

Then we replace the UE criteria with these link-level simplified SO criteria:

$$m_a = m_b, \text{ if } x_a > 0, x_b > 0$$

$$m_c = m_d, \text{ if } x_c > 0, x_d > 0$$

Solving this system of equations obtains the following SO flows:

$$\text{SO : } x_a^* = 13.6, x_b^* = 2.4, x_c^* = 2.57, x_d^* = 3.43, t_a^* = 5.4, t_b^* = 6.4, t_c^* = 3.29,$$

$$t_d^* = 6.43$$

The average travel times are no longer equal, and selfish travelers would have an incentive to change path. Under this flow assignment, we get the following TSTT: 119.34, which is lower cost than for the UE condition.

For larger networks with hundreds or thousands of links and nodes, it is not feasible to try to set up a system of equations to solve. A more systematic approach is needed.

3.2.2 Road Network Assignment

The history of traffic network assignment and influence of key players such as Beckmann et al. (1956) is noted in Boyce et al. (2005). At the time when Wardrop laid down the principles for network flow distribution, the United States was in a time of significant roadway expansion heading into President Eisenhower's era. As president, he championed the interstate highway system to improve intercity mobility. There was a great need for methods to efficiently predict the equilibration of flows on different links of a network.

One applicable science that emerged from that time was the theory of mathematical programming pioneered by George Dantzig, John von Neumann, and others. For example, it appeared that uncongested (fixed travel costs), uncapacitated network flow assignment can be expressed as a linear program (LP). This means one can represent each OD pair as a “commodity” and formulate a constrained optimization problem to assign flows to minimize total system costs.

A mathematical programming model describes a system in which different states are defined by different decision variables. The desired state is captured by the optimal solution of the model. Such models may be used to guide algorithms to obtain the optimal solution. More importantly, they can be used to analyze the effects of perturbations for any solution—optimal or feasible—as implications to the more complex, real-life system they are modeling.

For network flow, the system is divided into a set of components: links, nodes, and commodities. These can all be represented by sets: A for a set of directed links, N for a set of nodes, W for a set of commodities that need transport across the network. The whole system is defined as a graph $G[N, A]$ with commodities in W . A basic model of traveler assignment on an urban road network based on constant, uncongested travel costs is defined from the perspective of a system seeking to minimize total system travel times as a commodity-based link flow formulation in Eq. (3.1).

$$\min Z = \sum_{(i, j) \in A} \sum_{(r, s) \in W} t_{ij} x_{ij, rs} \quad (3.1a)$$

Subject to

$$\sum_{j \in N} x_{ij,rs} - \sum_{j \in N} x_{ji,rs} = q_{i,rs}, \quad \forall i \in N, \forall (r, s) \in W \quad (3.1b)$$

$$x_{ij,rs} \in \mathbb{Z}_+, \quad \forall (i, j) \in A, (r, s) \in W \quad (3.1c)$$

where $q_{i,rs} > 0$ if there is demand of $q_{i,rs}$ and $i=r$, $q_{i,rs} < 0$ if there is demand of $q_{i,rs}$ and $i=s$, and $q_{i,rs}=0$ otherwise.

Despite having integer constraints, the uncapacitated model satisfies the Unimodularity Theorem and Eq. (3.1c) can therefore be relaxed into $x_{ij,rs} \geq 0$ without worrying about getting a suboptimal solution. Note that the capacitated version does not satisfy the theorem and is why it is NP complete in complexity (Even et al., 1975). As an LP for the uncapacitated version, algorithms like the simplex method (see Dantzig, 1951) can be used to systematically reach the optimal solution for large instances of these problems. Furthermore, the unconstrained model can be trivially unbundled into individual shortest path problems for each OD pair and assigning all the flow to the shortest path. Shortest path problems have algorithms (e.g., Dijkstra, 1959) which are more efficient than LP methods for networks because their constraints tend to be sparse matrices. The model in Eq. (3.1) is called an “All-or-Nothing” assignment because all travelers are assigned to a shortest path for each OD commodity (Algorithm 3.1).

Algorithm 3.1 All-or-Nothing Assignment via Dijkstra's Algorithm

Inputs: link travel times t_{ij} , adjacency matrix δ , OD demand q_{rs} , unloaded link flows $x_{ij}=0$

1. For each OD (r, s)
 - a. Find shortest path P
 - i. Initiate: $d_r := 0$, $d_{i \neq r} := \infty$, $p_{i \neq r} := \emptyset$, $i := r$, $S := N \setminus r$
 - ii. While $s \in S$
 1. For all j where $\delta_{ij} = 1$, if $d_i + t_{ij} < d_j$, set $d_j := d_i + t_{ij}$ and $p_j := i$
 2. Set $S \setminus \arg\min_j \{d_{j \in S}\}$ and $i := j$
 - iii. Set $P := \{r, \dots, p_p, p_s, s\}$
 - b. For $i = 1 : |P| - 1$, set $x_{P_i, P_{i+1}} = q_{rs}$

Outputs: loaded link flows x_{ij}

How does one deal with congestion effect? Using this modeling framework, the original link cost parameter is now a variable that depends on total

link flow, $t_{ij} = \int [x_{ij}]$, where $x_{ij} = \sum_{(r,s) \in W} x_{ij,rs}$ is the bundle of all the different commodities that flow through link (i,j) . Due to the congestion effect, the objective is now nonlinear. Furthermore, solving the model with objective (3.1a) does not achieve Wardrop's UE condition.

In a seminal report for RAND Corporation, Beckmann et al. (1956) addressed this problem by proposing a nonlinear path flow formulation and showing that its optimal solution is equivalent to Wardrop's UE principle. First, a path flow formulation is a model in which the underlying decision variables are the path flow variables instead of the link flow variables. The path flow formulation of the uncongested, uncapacitated multicommodity flow problem is shown in Eq. (3.2).

$$\min Z = \sum_{k \in K} \sum_{(r,s) \in W} c_k f_{rs,k} \quad (3.2a)$$

Subject to

$$\sum_{k \in K} f_{rs,k} = q_{rs}, \quad \forall (r,s) \in W \quad (3.2b)$$

$$f_{rs,k} \geq 0, \quad \forall k \in K, (r,s) \in W \quad (3.2c)$$

where K is the set of paths, $f_{rs,k}$ is the path flow, q_{rs} is the demand for commodity OD pair (r,s) , and c_k is the path cost. The advantages of a path flow formulation are discussed by Tomlin (1971) and others cited within: the number of decision variables and constraints tend to be smaller for the path flow model, it works well with solution methods that implicitly enumerate paths, and the model can exhibit optimality conditions in terms of path flows and costs.

Beckmann et al. (1956) formulated the assignment model with two considerations. The first was the use of the path-based formulation. The second was the use of an objective function that achieved the criteria put forth by Pigou (1920). This UE formulation is shown in Eq. (3.3).

$$\min Z = \sum_{(i,j) \in A} \int_0^{x_{ij}} t_{ij}[w] dw \quad (3.3a)$$

Subject to

$$\sum_{k \in K} f_{rs,k} = q_{rs}, \quad \forall (r,s) \in W \quad (3.3b)$$

$$x_{ij} = \sum_{k \in K} \sum_{(r, s) \in W} \delta_{rs, ijk} f_{rs, k}, \quad \forall (i, j) \in A \quad (3.3c)$$

$$f_{rs, k} \geq 0, \quad \forall k \in K, (r, s) \in W \quad (3.3d)$$

where $\delta_{rs, ijk}$ is an indicator parameter set to 1 if path k for commodity (r, s) traverses link (i, j) . The objective can be interpreted as minimizing the area under the average cost function consistent with the criterion used in [Exercise 3.2](#). It does not have any economic interpretation, however, and should not be used to represent social welfare.

The SO formulation is simply replacing Eq. (3.3a) with Eq. (3.1a), keeping in mind that t_{ij} is a function of link flow x_{ij} . An illustration of the UE formulation applied to a small instance is provided in [Exercise 3.4](#).

Exercise 3.4

Formulate the UE model for the example in [Exercise 3.3](#).

We define the path-link incidence matrix δ as presented in [Table 3.2](#).

Table 3.2 Path-link incidence matrix for [Exercise 3.4](#)

Link\OD-path	12, 1	12, 2	13, 3	13, 4	13, 5	13, 6
a	1	0	1	1	0	0
b	0	1	0	0	1	1
c	0	0	1	0	1	0
d	0	0	0	1	0	1

Then the model is as follows.

$$\min Z = \left(2x_a + \frac{1}{8}x_a^2 \right) + \left(4x_b + \frac{1}{2}x_b^2 \right) + \left(\frac{1}{6}x_c^3 \right) + \left(3x_d + \frac{1}{2}x_d^2 \right)$$

Subject to

$$\begin{aligned} f_{12,1} + f_{12,2} &= 10 \\ f_{13,3} + f_{13,4} + f_{13,5} + f_{13,6} &= 6 \\ x_a &= f_{12,1} + f_{13,3} + f_{13,4} \\ x_b &= f_{12,2} + f_{13,5} + f_{13,6} \\ x_c &= f_{13,3} + f_{13,5} \\ x_d &= f_{13,4} + f_{13,6} \\ f_{rs, k} &\geq 0, \quad \forall (r, s) \in W, k \in K \end{aligned}$$

The equality constraints can also be represented in matrix form.

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} f_{12,1} \\ f_{12,2} \\ f_{13,3} \\ f_{13,4} \\ f_{13,5} \\ f_{13,6} \\ x_a \\ x_b \\ x_c \\ x_d \end{bmatrix} = \begin{bmatrix} 10 \\ 6 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

As this example shows, the constraint matrix is very sparse, even for such a compact network. The link flow variables can be removed entirely by replacing them all with the appropriate sums of path flows and taking those constraints out. This formulation is composed of only equality constraints or nonnegativity inequality constraints.

[Beckmann et al. \(1956\)](#) further proved existence and uniqueness conditions and that the optimum is indeed equivalent to Wardrop's UE principle. Later, [Dafermos and Sparrow \(1969\)](#) noted that the solution to the UE problem is also a Nash equilibrium.

Theorem 3.1 ([Beckmann et al., 1956](#)). *The optimum of the nonlinear program in Eq. (3.3), when the link cost functions are monotonically increasing with respect to the link flows, is unique with respect to the link flows.*

Proof Since the link cost functions are monotonically increasing with respect to the link flows, it can be shown that the Hessian of the objective function is positive semidefinite. Therefore the set of link flows corresponding to the optimum is a global optimum and unique. ▀

Theorem 3.2 ([Beckmann et al., 1956](#)). *The optimum of the nonlinear program in Eq. (3.3) satisfies Wardrop's UE condition ([Principle 3.1](#)).*

Proof The proof consists of finding the optimality conditions and showing that they are equivalent to Wardrop's UE principle. For nonlinear constrained optimization, optimality conditions can be obtained using Karush–Kuhn–Tucker (KKT) conditions. These are found by first formulating the Lagrangian of Eq. (3.3), as shown in Eq. (3.4), by relaxing the demand flow conservation constraints in Eq. (3.3b). Since the link flow conservation constraints in Eq. (3.3c) are only definitional, relaxing Eq. (3.3b) makes the Lagrangian a nonlinear optimization model with only nonnegativity constraints.

$$L[f, u] = Z[f] + \sum_{(r, s) \in W} u_{rs} \left(q_{rs} - \sum_{k \in K} f_{rs,k} \right) \quad (3.4)$$

where u_{rs} is a dual variable pertaining to flow conservation constraint. KKT conditions of the Lagrangian require that the following conditions in Eq. (3.5) are satisfied.

$$f_{rs,k} \frac{\partial L}{\partial f_{rs,k}} = 0, \quad \forall k \in K, (r, s) \in W \quad (3.5a)$$

$$\frac{\partial L}{\partial f_{rs,k}} \geq 0, \quad \forall k \in K, (r, s) \in W \quad (3.5b)$$

$$\frac{\partial L}{\partial u_{rs}} = 0, \quad \forall (r, s) \in W \quad (3.5c)$$

$$f_{rs,k} \geq 0, \quad \forall k \in K, (r, s) \in W \quad (3.5d)$$

$\frac{\partial L}{\partial f_{rs,k}}$ has two terms. The first term, $\frac{\partial Z}{\partial f_{rs,k}}$, essentially sums up all the link travel times that are on the path, that is, $\frac{\partial Z}{\partial f_{rs,k}} = \sum_{(i, j)} t_{ij} \delta_{rs, ijk}$. Since this is simply the path cost shown in Eq. (3.2a), we get $\frac{\partial Z}{\partial f_{rs,k}} = c_{rs,k}$. The second term is the Lagrangian term: $\frac{\partial}{\partial f_{rs,k}} \sum_{(r, s) \in W} u_{rs} (q_{rs} - \sum_{k \in K} f_{rs,k}) = -u_{rs}$. Substituting in the terms to Eq. (3.5), we get the following in Eq. (3.6).

$$f_{rs,k} (c_{rs,k} - u_{rs}) = 0, \quad \forall k \in K, (r, s) \in W \quad (3.6a)$$

$$c_{rs,k} - u_{rs} \geq 0, \quad \forall k \in K, (r, s) \in W \quad (3.6b)$$

$$\sum_{k \in K} f_{rs,k} = q_{rs}, \quad \forall (r, s) \in W \quad (3.6c)$$

$$f_{rs,k} \geq 0, \quad \forall k \in K, (r, s) \in W \quad (3.6d)$$

Eq. (3.6a) is a complementary slackness condition. When $f_{rs,k}=0$, then $c_{rs,k} \geq u_{rs}$ based on Eq. (3.6b). When $f_{rs,k}>0$, then $c_{rs,k}=u_{rs}$. As the dual price, u_{rs} is equivalent to the minimum path cost among all paths for OD (r, s) . The complementary slackness conditions spell out Wardrop's UE condition: when there is path flow ($f_{rs,k}>0$), the path costs are equal (to u_{rs}); when there is no path flow ($f_{rs,k}=0$), the cost of that path k is greater than or equal to the minimum path cost ($c_{rs,k} \geq u_{rs}$). ■

Of great importance is the fact that uniqueness of path flows is not guaranteed. This is because the Hessian of the objective function with respect to

the path flow variables is not guaranteed to be positive definite unless the network is of a simplified structure where path flows have a one-to-one relationship with link flows. The implication is that the solution of this model does not actually output path flows and choices; it only outputs the more aggregate link flow solution. For transport modelers looking to determine the volumes on different road segments, this model is sufficient.

The model can be solved by any solution algorithms for constrained nonlinear programming problems. However, such algorithms were not known to transport modelers for some time. Heuristics were developed to obtain solutions to the UE model. For example, “iterative assignment” updates the travel times based on the assigned flows, and then use the new travel times to determine flows as decomposed shortest path problems. Convergence was not guaranteed (and usually not possible if the solution was an interior point in the constraint space). In “incremental assignment,” portions of the flow are loaded onto the network incrementally (e.g., 10%, then 20%, then 30%, then 40%), where the updated travel times from the cumulatively updated flows are used to assign flows in the next increment.

[LeBlanc et al. \(1975\)](#) first proposed using a nonlinear programming algorithm developed by [Frank and Wolfe \(1956\)](#), ironically published in the same year as [Beckmann et al. \(1956\)](#). The Frank-Wolfe (F-W) algorithm is designed for finding a local optimum to a nonlinear constrained optimization model with linear constraints. The methodology iteratively takes the gradient at its current solution as coefficients of a linear objective, resulting in a solvable linear program. The vertex solution of the linear program is then used as the direction relative to the current location, at which point it is possible to collapse the multidimensional optimization problem into a scalar line search problem to find the next point for the next iteration.

The algorithm from [LeBlanc et al. \(1975\)](#), as applied to transport network assignment, uses the same methodology as F-W algorithm. The gradient at a current solution is simply the all-or-nothing assignment because the gradient of the UE assignment objective is just the fixed travel times at the current value of link flows. The algorithm is then to iteratively assume updated travel times are fixed, solve the new all-or-nothing problem based on [Algorithm 3.1](#), use that solution as a “vertex” to direct a scalar line search from the current point. There is a further interpretation. Each time the algorithm finds a new corner point, it is proposing to add a new path to the set of path flows. The line search is finding an optimal weighting between all previously found paths and the new paths. In that sense, it is similar to the column generation techniques to implicitly enumerate path flows as discussed

by Tomlin (1971) for capacitated multicommodity flow problems. LeBlanc et al.'s F-W algorithm is shown in [Algorithm 3.2](#) and demonstrated in [Exercise 3.5](#).

Algorithm 3.2 Frank-Wolfe Algorithm for Transport Network Assignment (LeBlanc et al., 1975)

Inputs: network and demand attributes stored in $G[N, A]$ and W , link cost functions t_{ij} , marginal cost functions m_{ij} if solving SO, unloaded link flows $x_{ij}^0 = 0$, desired stopping threshold ϵ .

1. Initiate $n=0$, $t_{ij}^0[x_{ij}^0]$ and then finding x_{ij}^1 using [Algorithm 3.1](#) with t_{ij}^0
2. While stopping threshold ϵ is not met
 - a. $n=n+1$, update $t_{ij}^n[x_{ij}^n]$
 - b. Based on t_{ij}^n , run [Algorithm 3.1](#) to obtain auxiliary link flows y_{ij}^n
 - c. Solve the following scalar optimization problem with respect to decision variable α :

$$\min Z[x^n + \alpha(y^n - x^n)] \quad (3.7)$$

where $Z_{UE}[x_{ij}] = \sum_{x_{ij}} \int_0^{x_{ij}} t_{ij}[w] dw$ if under UE principle, and

$Z_{SO}[x_{ij}] = \sum_{x_{ij}} \int_0^{x_{ij}} m_{ij}(w) dw$ if under SO principle.

- d. Update: $x^{n+1} = x^n + \alpha^n(y^n - x^n)$ based on optimal α^n
3. Set $x_{ij}^* = x_{ij}^n$

Outputs: loaded link flows x_{ij}^*

Exercise 3.5

Show one iteration of [Algorithm 3.2](#) to solve for the SO for the example in [Exercise 3.3](#).

1. First, since this is SO, we find the marginal cost functions:

$$m_a = 2 + \frac{1}{2}x_a, \quad m_b = 4 + 2x_b, \quad m_c = \frac{3}{2}x_c^2, \quad m_d = 3 + 2x_d$$

2. We set $n=0$, $x_a^0 = x_b^0 = x_c^0 = x_d^0 = 0$, and initial values for travel times: $t_a^0 = 2$, $t_b^0 = 4$, $t_c^0 = 0$, $t_d^0 = 3$ based on the initial flows.
3. Set $n=1$. [Algorithm 3.1](#) using t^0 : $x_a^1 = 10 + 6 = 16$, $x_b^1 = 0$, $x_c^1 = 6$, $x_d^1 = 0$. This is our initial feasible flow.
4. Update $t_a^1[16] = 6$, $t_b^1[0] = 4$, $t_c^1[6] = 18$, $t_d^1[0] = 3$.
5. Find auxiliary flows. For OD (1, 2), the shortest path is $\{b\}$, so we assign all the auxiliary flow on there: $y_b^1 = 10$. For OD (1, 3), the shortest path is $\{b, d\}$. We update: $y_a^1 = 0$, $y_b^1 = 10 + 6 = 16$, $y_c^1 = 0$, $y_d^1 = 6$.

6. We set up the line search based on setting the derivative of Eq. (3.7) under SO principle:

$$\frac{\partial}{\partial \alpha} Z_{SO}[x^n + \alpha^n(y^n - x^n)] = \sum_{(i,j)} \left(y_{ij}^n - x_{ij}^n \right) m_{ij} \left[x_{ij}^n + \alpha^n \left(y_{ij}^n - x_{ij}^n \right) \right] = 0$$

The terms are:

$$\begin{aligned} & (0-16) \left(2 + \frac{1}{2} (16 + \alpha^1(0-16)) \right) + (16-0) (4 + 2(0 + \alpha^1(16-0))) \\ & + (0-6) \left(\frac{3}{2} (6 + \alpha^1(0-6))^2 \right) + (6-0) (3 + 2(0 + \alpha^1(6-0))) \\ & = -160 + 128\alpha^1 + 64 + 512\alpha^1 - 324 + 648\alpha^1 - 324 [\alpha^1]^2 \\ & + 18 + 72\alpha^1 = 0 \end{aligned}$$

This can be solved using any numerical solution method, such as bisection method or “Goal Seek” in Excel, to find $0 \leq \alpha^n \leq 1$. A value of $\alpha^n = 0$ implies that the optimal solution has been reached. We get $\alpha^1 = 0.320$.

7. Update flows: $x_a^2 = 16 + (0.320)(0-16) = 10.880$, $x_b^2 = 0 + 0.320(16-0) = 5.120$, $x_c^2 = 6 + 0.320(0-6) = 4.080$, and $x_d^2 = 0 + 0.320(6-0) = 1.920$. These values are now closer to the true solution of $x_a^* = 13.6$, $x_b^* = 2.4$, $x_c^* = 2.57$, $x_d^* = 3.43$ (from [Exercise 3.3](#)). Note that the updated travel times would be $t_a^2 = 4.720$, $t_b^2 = 9.120$, $t_c^2 = 8.324$, $t_d^2 = 4.920$, which would be used for [Algorithm 3.1](#) for the next iteration.
-

Other algorithms have also been introduced to solve the model (e.g., [Jayakrishnan et al., 1994](#); [Bar-Gera, 2002](#)), but many of the current transport modeling software still make use of the original F-W algorithm as a default option for obtaining the link flows.

While there are other methods to forecast flows at the facility level, the network approach presented here ensures that equilibration of OD demand distributes flows onto the network according to economic principles consistent with division of public goods known since [Pigou \(1920\)](#). It is important to predict how travelers would adjust flows on a network based on changes to the network because not all investments on a link in a network will improve the system performance. This is famously illustrated by Braess’ Paradox (see [Murchland, 1970](#)) shown in the following example.

Consider a 4-node network with a OD demand of 6 units for (o, r) , as shown in [Fig. 3.4](#). There are two scenarios, a “before” scenario where there are only four links, and an “after” scenario where a fifth link (p, q) has been

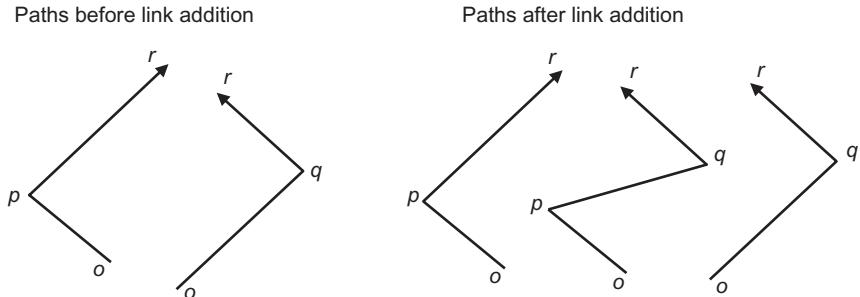


Fig. 3.4 Before and after scenarios of network for Braess' Paradox example.

added. Fig. 3.4 shows the possible paths in both scenarios. The following average link cost functions (in minutes) are used:

$$\begin{aligned} t_{op} &= 10x_{op}, \quad t_{qr} = 10x_{qr} \\ t_{pr} &= 50 + x_{pr}, \quad t_{oq} = 50 + x_{oq} \\ t_{pq} &= 10 + x_{pq} \end{aligned}$$

In the “before” scenario, under Wardrop’s UE principle, the 6 units of flow are distributed equally along the two routes (o, p, r) and (o, q, r) , because that leads to the same average travel times of 83 min. No traveler would unilaterally change route.

In the “after” scenario, however, the additional path created by adding (p, q) results in a new UE assignment where each route receives 2 units of flow. The average travel time of each route becomes 92 min, which is 9 min higher than before adding the link. The assumption that simply adding new capacity to a network would improve system performance is proven to be false. This counterexample explains why it is important to evaluate the performance of *the network as a whole* instead of in parts and to consider *the behavior of travelers* when doing so.

3.2.3 Traffic Assignment Variants

There have been many variants of road network equilibrium models over the years to account for different types of policy concerns and questions. Several key developments are reviewed here.

One of the major changes is to the set of behavioral principles proposed by Wardrop. Daganzo and Sheffi (1977) proposed a third principle, **Principle 3.2**, for the case of travelers behaving under information with perception error.

Principle 3.2

([Daganzo and Sheffi, 1977](#)). In a stochastic user equilibrium network no user believes he can improve his travel time by unilaterally changing routes.

By adding beliefs under incomplete information, it is possible to tune parameters in network assignment models to account for the availability of information to travelers. This works by first modeling the route assignment in a choice-based framework instead of a network flow framework. Discrete choice models ([Luce, 1959](#); [McFadden, 1974](#)) provide a theoretical framework that aligns with psychological behavior under a random utility model ([Manski, 1977](#)). [Appendix D](#) provides a review of discrete choice models.

Individuals' probabilistic preferences of alternatives among a choice set can be predicted based on a utility function that relates preferences to different attributes pertaining to the individual and the alternative, as shown in Eq. (3.8a). The variable U_{in} is the utility gained by individual n when selecting alternative i among a set of alternatives available to them, C_n . This utility is divided between the set of k observable attributes and the sum of all unobservable factors represented by a random term ε_{in} . When the term is fitted with an extreme value Gumbel distribution, then it is possible to obtain the distribution of the maximum among the choice set (e.g., $\max(U_{1n}, \dots, U_{|C_n|,n})$) as a logistic function shown in Eq. (3.8b). Other distributions for ε_{in} are also possible.

$$U_{in} = \beta_{n1}x_{in1} + \dots + \beta_{nk}x_{ink} + \varepsilon_{in} = V_{in} + \varepsilon_{in} \quad (3.8a)$$

$$P_n[i | C_n] = P_n[U_{in} \geq U_{jn}, j \neq i] = \frac{e^{V_{in}}}{\sum_{j \in C_n} e^{V_{jn}}} \quad (3.8b)$$

This model can be used as an aggregate route assignment model where the representative utility V_{in} is a function of route characteristics such as travel time. [Dial \(1971\)](#) proposed an algorithm to efficiently assign flows onto an uncongested network via a two-phase algorithm by noting that the route-level preferences can be broken down into link-level attributes and probabilities ([Algorithm 3.3](#)).

The parameter θ , where $\theta \geq 0$, represents the degree of certainty that travelers have of the shortest path. As θ increases, travelers become more certain of which path is shortest.

The algorithm is illustrated in [Exercise 3.6](#).

Algorithm 3.3 Dial's (1971) Algorithm

Inputs: Network $G[N, A]$ with link travel times t_{ij} , and OD demands q_{rs} with $(r, s) \in W$

0. Initiate for each node i the shortest path from r to i as p_i and the shortest path from i to s as q_i , the set of downstream links I_i and upstream links F_i of a node i , and defining a likelihood function for flow in Eq. (3.9)

$$l_{ij} = \begin{cases} e^{\theta(p_j - p_i - t_{ij})} & \text{if } p_i < p_j, q_j < q_i \\ 0 & \text{otherwise} \end{cases} \quad (3.9)$$

1. Forward pass. In ascending order by p_i from r to s , calculate the “link weight” as shown in Eq. (3.10)

$$w_{ij} = \begin{cases} l_{ij} & \text{if } i = r \\ l_{ij} \sum_{(k, i) \in F_i} w_{ki} & \text{otherwise} \end{cases} \quad (3.10)$$

2. Backward pass. In descending order by p_j from s to r , assign link flows x_{ij} as shown in Eq. (3.11)

$$x_{ij} = \begin{cases} \frac{q_{rs} w_{ij}}{\sum_{(k, j) \in F_j} w_{kj}} & \text{if } j = s \\ w_{ij} \sum_{(j, k) \in I_i} x_{jk} & \text{otherwise} \\ \frac{q_{rs} w_{ij}}{\sum_{(k, j) \in F_j} w_{kj}} & \text{otherwise} \end{cases} \quad (3.11)$$

Outputs: Link flows x_{ij}

The assignments made in [Exercise 3.6](#) do not consider congestion effects. To account for congestion, the deterministic UE objective can be modified to incorporate the satisfaction of utility from a choice set, as shown by [Fisk \(1980\)](#) in Eq. (3.12).

$$\min Z_{SUE} = \frac{1}{\theta} \sum_{(r, s) \in W} \sum_{k \in K} f_{rsk}(\ln f_{rsk}) + \sum_{(i, j) \in A} \int_0^{x_{ij}} t_{ij}[w] dw \quad (3.12)$$

Note that $\lim_{f_{rsk} \rightarrow 0} f_{rsk}(\ln f_{rsk}) = 0$. To solve this model, there needs to be some type of iterative approach to converge toward a fixed point solution. However, because cost functions do not monotonically improve anymore, the F-W algorithm is not applicable.

Exercise 3.6 Assuming the network is uncongested, apply Algorithm 3.3 to obtain link flows for the example from Exercise 3.3 under $\theta=0.1$ and $\theta=0.5$.

1. For $\theta=0.1$ (more uncertainty):

- a. For OD (1, 2), the flow assignment is straightforward:

$$x_{12,a} = \frac{e^{-0.1(2)} 10}{e^{-0.1(2)} + e^{-0.1(4)}} = 5.498, x_{12,b} = 10 - 5.498 = 4.502$$

- b. For OD (1, 3), apply Algorithm 3.3:

- i. Set $p_1=0, p_2=2, p_3=2, q_1=2, q_2=0, q_3=0$

- ii. Set $l_a=e^{0.1(2-0-2)}=1, l_b=e^{0.1(2-0-4)}=0.819,$

$l_c=e^{0.1(2-2-0)}=1, l_d=e^{0.1(2-2-3)}=0.741$ (although $p_2=p_3$, we can assume logically that movement from node 2 to node 3 is reasonable)

- iii. Forward pass: $w_a=1, w_b=0.819, w_c=1(1+0.819)=1.819,$

$$w_d=0.741(1+0.819)=1.347$$

- iv. Backward pass: $x_{13,d}=\frac{6(1.347)}{1.347+1.819}=2.553,$

$$x_{13,c}=\frac{6(1.819)}{1.347+1.819}=3.447, x_{13,b}=\frac{0.819(2.553+3.447)}{0.819+1}$$

$$=2.701, x_{13,a}=\frac{1(2.553+3.447)}{0.819+1}=3.299$$

- c. Summing the OD-specific flows up, we get: $x_a=5.498+3.299=8.797, x_b=4.502+2.701=7.203, x_c=3.447, x_d=2.553$

2. For $\theta=0.5$ (less uncertainty):

- a. OD (1, 2): $x_{12,a}=\frac{e^{-0.5(2)} 10}{e^{-0.5(2)} + e^{-0.5(4)}}=7.311, x_{12,b}=2.689$

- b. OD (1, 3):

- i. Set $p_1=0, p_2=2, p_3=2, q_1=2, q_2=0, q_3=0$

- ii. Set $l_a=1, l_b=e^{0.5(2-0-4)}=0.368, l_c=1, l_d=e^{0.5(2-2-3)}=0.223$

- iii. $w_a=1, w_b=0.368, w_c=1(1+0.368)=1.368,$

$$w_d=0.223(1+0.368)=0.305$$

- iv. $x_{13,d}=\frac{6(0.305)}{0.305+1.368}=1.095,$

$$x_{13,c}=\frac{6(1.368)}{0.305+1.368}=4.905, x_{13,b}=\frac{0.368(1.095+4.905)}{0.368+1}$$

$$=1.614, x_{13,a}=\frac{1(1.095+4.905)}{0.368+1}=4.386$$

- v. Summing up: $x_a=7.311+4.386=11.697, x_b=4.303, x_c=4.905, x_d=1.095$

As this example illustrates, the network assignment can be calibrated to fit the degree of uncertainty present in the travelers' shortest path choice decision-making.

Fisk (1980) proposed a Method of Successive Averages (MSA), for which Powell and Sheffi (1982) provided a proof of asymptotic convergence, to solve the SUE. The MSA method shown in [Algorithm 3.4](#) is a very flexible successive average approximation approach that is applicable to many fixed point problems. The only major setback to the approach is that it is very slow to converge (although recent studies like Liu et al. (2009) have sought to remedy that).

Algorithm 3.4 (as Described in Sheffi, 1985): Method of Successive Averages for SUE

Inputs: Network $G[N, A]$ with link travel time functions $t_{ij}[x_{ij}]$ and OD demands q_{rs} , $(r, s) \in W$, convergence threshold ϵ .

1. Initiate with $t_{ij}[0]$ to obtain x^1 using [Algorithm 3.3](#). Set $n := 0$
2. While convergence threshold ϵ not met
 - a. $n := n + 1$
 - b. Update t^n based on x^n .
 - c. Run [Algorithm 3.3](#) using t^n to obtain auxiliary link flows y^n .
 - d. Update $x^{n+1} = x^n + \frac{1}{n}(y^n - x^n)$
3. Set $x^* = x^{n+1}$

Outputs: congested link flows x_{ij}^* under SUE

The MSA algorithm is illustrated in [Exercise 3.7](#).

Exercise 3.7

Using the results from [Exercise 3.6](#) as the initiated x^1 , perform one iteration of [Algorithm 3.4](#) for the case with $\theta = 0.5$.

1. The initial flows are $x_a^1 = 11.697$, $x_b^1 = 4.303$, $x_c^1 = 4.905$, $x_d^1 = 1.095$.
2. Update travel times: $t_a^1 = 4.924$, $t_b^1 = 8.303$, $t_c^1 = 12.030$, $t_d^1 = 4.095$.
3. Run [Algorithm 3.3](#):
 - a. OD (1, 2): $y_{12,a}^1 = \frac{e^{-0.5(4.924)}10}{e^{-0.1(4.924)} + e^{-0.1(8.303)}} = 8.441$, $y_{12,b}^1 = 1.559$
 - b. OD (1, 3):
 - i. Set $p_1 = 0$, $p_2 = 4.924$, $p_3 = 9.019$, $q_1 = 9.019$, $q_2 = 4.924$, $q_3 = 0$
 - ii. Set $l_a = 1$, $l_b = e^{0.5(4.924-0-8.303)} = 0.185$,
 $l_c = e^{0.5(9.019-4.924-12.030)} = 0.019$, $l_d = e^{0.5(9.019-4.924-4.095)} = 1$
 - iii. $w_a = 1$, $w_b = 0.185$, $w_c = 0.019(1 + 0.185) = 0.022$,
 $w_d = 1(1 + 0.185) = 1.185$
 - iv. $y_{13,d}^1 = \frac{6(1.185)}{1.185 + 0.022} = 5.889$,
 - $y_{13,c}^1 = 0.111$, $y_{13,b}^1 = \frac{0.185(6)}{1 + 0.185} = 0.935$, $y_{13,a}^1 = 5.065$

c. Summing up: $y_a^1 = 8.441 + 5.065 = 13.506$, $y_b^1 = 2.494$, $y_c^1 = 0.111$, $y_d^1 = 5.889$

4. Update:

$$x^2 = \begin{bmatrix} 11.697 \\ 4.303 \\ 4.905 \\ 1.095 \end{bmatrix} + \frac{1}{1} \left(\begin{bmatrix} 13.506 \\ 2.494 \\ 0.111 \\ 5.889 \end{bmatrix} - \begin{bmatrix} 11.697 \\ 4.303 \\ 4.905 \\ 1.095 \end{bmatrix} \right) = \begin{bmatrix} 13.506 \\ 2.494 \\ 0.111 \\ 5.889 \end{bmatrix}$$

The following iterations would take $\frac{1}{2}$ the difference, then $\frac{1}{3}, \frac{1}{4}, \dots$, until the n^{th} iteration would only contribute $\frac{1}{n}$ to the weight.

There are many other variations of the UE network assignment model in the literature. Several of them are reviewed in detail in Sheffi (1985): endogenous demand, joint distribution and assignment (Florian et al., 1975), mode choice and assignment on a supernetwork (Florian and Nguyen, 1978), and asymmetric cost functions. While Beckmann et al. (1956) introduced a nonlinear programming model for the UE assignment, it is also possible to reformulate this as a “variational inequality” to address asymmetric cost functions. Smith (1979) proposed an equivalent formulation which Dafermos (1980) showed was equivalent to variational inequalities (VI). This equivalent VI formulation, shown in Eq. (3.13), is satisfied if a link flow vector \bar{f} is user optimized.

$$c[\bar{f}] \cdot (f - \bar{f}) \geq 0 \quad \forall f \in \kappa \quad (3.13)$$

where c is a link cost vector, $f \in \kappa$ is a vector of link flows. This is equivalent to the KKT conditions in Eq. (3.6), and furthermore, have more flexible properties such as being able to handle scenarios where travel costs on one link may depend on flows on another link (i.e., nonseparability), or where there are interactions between two different modes of transportation on the same link. Examples of nonseparable problems can be found in Watling and Hazelton (2003). The difference between a separable problem and nonseparable problem discussed in that study is shown here. Consider a single OD pair with two parallel routes serving 2 units of demand.

Separable problem: $c_1[f] = 3f_1 + 1$, $c_2[f] = 2f_2 + 2$, $n > 0$

In this example, regardless of n , the separable problem has a UE solution at $(f_1, f_2) = (1, 1)$.

Nonseparable problem: $c_1[f] = 3f_1 + f_2 + 1$, $c_2[f] = 2f_1 + f_2 + 2$

The UE solution is still at $(f_1, f_2) = (1, 1)$, although as pointed out in Smith (1984a) this solution is not necessarily stable because perturbed

travelers (say, at $(f_1, f_2) = (\frac{3}{4}, \frac{5}{4})$) would not have incentive to switch back to $(f_1, f_2) = (1, 1)$ because the cost of route 2 under the perturbed scenario is lower than in the UE state. Based on such differences, there is a stricter definition of “user optimality” in which Wardrop’s UE principle is satisfied and the solution is locally stable. These situations can occur in such asymmetric traffic assignment problems.

The VI is therefore a more generalized form of the UE assignment model than the mathematical programming formulation from [Beckmann et al. \(1956\)](#). [Dafermos \(1982\)](#) further proved convergence for a solution algorithm for multimodal network assignment with mode choice. Alternatively, one approach is based on the use of a gap function ([Hearn, 1982](#)) as proposed by [Marcotte \(1985\)](#). This function has been found to be useful for network design problems as well and is revisited in [Chapter 7](#).

One of the major advantages of VIs is the possibility of systematically conducting sensitivity analysis on the network’s performance with respect to perturbations in path flows. Some of the early work in this area include [Dafermos and Nagurney \(1984\)](#), who proposed a methodology under certain assumptions for the number of paths in the network, and a more generalized approach by [Tobin and Friesz \(1988\)](#) for fixed demand equilibrium problems ([Yang and Bell, 2007](#), published a correction to this method). [Yang \(1997\)](#) proposed a sensitivity analysis method for elastic demand user equilibrium problems. [Clark and Watling \(2000\)](#) proposed a sensitivity analysis method for probit-based SUE problems. [Patriksson \(2004\)](#) proposed an alternative to [Tobin and Friesz’s \(1988\)](#) method that was meant to be more aligned with the original traffic equilibrium model.

[Beckmann et al. \(1956\)](#) also studied the endogenous demand case in which the OD demand q_{rs} is a decision variable governed by a demand function, $q_{rs} = D_{rs}[u_{rs}]$, where u_{rs} is the minimum travel time between (r, s) . Eq. (3.3a) can be modified to Eq. (3.14) to include demand consideration. In addition, a new set of nonnegativity constraints are added for the demand variables, $q_{rs} \geq 0 \forall (r, s) \in W$.

$$\min Z_D = \sum_{(i, j) \in A} \int_0^{x_{ij}} t_{ij}[w] dw - \sum_{(r, s) \in W} \int_0^{q_{rs}} D_{rs}^{-1}[w] dw \quad (3.14)$$

Three other variants are worth covering because of their relevance to capacitated multimodal networks in a smart cities setting. The first is the consideration of steady-state queue delays with link flow capacities. One example of such is [Bell’s \(1995\)](#) formulation of an SUE model with queue

delay. This has applications in transport service networks that feature vehicle or station capacities such as fixed route public transit systems. Although the model does not directly handle on-demand systems, it might also be relevant to those systems as well.

A second consideration is for freight systems. Freight demand is high dimensional because it is not just multiple OD pairs, but also multiple commodity types. In addition to link facilities, freight systems differ from urban passenger transport networks in the emphasis on nodal facilities: transshipment facilities, ports, rail terminals, distribution centers, and urban consolidation centers. Guelat et al. (1990) studied this problem by considering freight nodal facilities as hubs of virtual links. Chow et al. (2014) further extended the model to distinguish service vehicular flows that do not enter or exit the system from commodity flows that do. This separation of vehicular flows and demand flows is also important for MaaS systems in smart cities. As such, the formulation is shown here in Eq. (3.15), where the network $G[N, A]$ consists of links that are divided into multiple modes $m \in M$ where $A_m \subseteq A$ and commodities W that are divided into multiple products $p \in P$ where $W_p \subseteq W$. This model combines link-based commodity flow variables (vehicular flows) with path-based variables (commodity flows).

$$\begin{aligned} \min Z_{CV}[f, x, \gamma, e, g] = & \sum_{m \in M} \sum_{(i, j) \in A_m} \int_0^{\gamma_{mij}} c_{mij}[w] dw \\ & + \sum_{m \in M} \sum_{n \in M} \sum_{\substack{(i, j) \in A_m \\ n \neq m}} \sum_{(j, k) \in A_n} \int_0^{x_{mnijk}} t_{mnijk}[w] dw \end{aligned} \quad (3.15a)$$

Subject to

$$\sum_{p \in P} r_{pmj} f_{pmij} + e_{mij} = \gamma_{mij}, \quad \forall m \in M, (i, j) \in A_m \quad (3.15b)$$

$$\sum_{\substack{i \in N \\ (i, j) \in A_m}} \gamma_{mij} - \sum_{\substack{i \in N \\ (j, i) \in A_m}} \gamma_{mji} = 0, \quad \forall m \in M, j \in N \quad (3.15c)$$

$$\sum_{\pi} g_{p\pi rs} = q_{prs}, \quad \forall p \in P, (r, s) \in W_p \quad (3.15d)$$

$$\sum_{(r, s) \in W_p} \sum_{\pi \in K} g_{p\pi rs} \delta_{rs\pi mij} = f_{pmij}, \quad \forall p \in P, (i, j) \in N \quad (3.15e)$$

$$\sum_{(r,s) \in W_p} \sum_{\pi \in K} g_{p\pi rs} \delta_{rs\pi mnijk} = x_{pmnijk}, \forall p \in P, j \in N \quad (3.15f)$$

$$\sum_{n \in M} \sum_{k \in N} x_{pmnijk} = f_{pmij}, \forall p \in P, m \in M, (i,j) \in A_m \quad (3.15g)$$

$$\sum_{m \in M} \sum_{i \in N} x_{pmnijk} = f_{pnjk}, \forall p \in P, n \in M, (j,k) \in A_n \quad (3.15h)$$

$$e_{mij} \geq 0, g_{p\pi rs} \geq 0 \quad (3.15i)$$

where f is the commodity link flow, x is the total commodity link transfers, from mode m on link (i,j) to mode n on link (j,k) , γ is the total vehicular link flow, e is the empty vehicular link flow, and g is the commodity path flow on path $\pi \in K$. In addition, c_{mij} is a link cost function, t_{mnijk} is a transfer cost function, r_{pm} is a commodity-to-vehicle conversion rate (payload factor), $\delta_{rs\pi mnijk}$ is an indicator that path π for OD (r,s) lies on link (i,j) in mode m .

The objective (3.15a) ensures that a UE solution is reached where the first term is for vehicular delay on links and the second term is for vehicular delay on transfers. Note that vehicular flows are equal to commodity flows plus empty flows, as ensured by Eq. (3.15b). Eq. (3.15c) ensures that all vehicular flow into a node is equal to the same flow leaving the node, for all nodes (including commodity origins and destinations). In this way, vehicular flows are preserved in the network to serve commodities. Eqs. (3.15d)–(3.15f) ensure that commodity path flows satisfy the demand and sum up to the link flows. Eqs. (3.15g)–(3.15h) ensure that transfer flows add back up to commodity flows. The model is a nonlinear programming problem with linear equality constraints (same as Eq. 3.3). The variables in the objective are vehicular link and transfer flows; it is shown in Chow et al. (2014) that the objective is convex with respect to these link flows and thus any nonlinear programming algorithm like F-W algorithm would converge to a unique solution with respect to those flows.

An example of the model is shown in Fig. 3.5, where a simple bimodal network of rail links (dashed lines connecting nodes 4, 5, and 6) and highway links (solid lines) can have an assignment of commodities and vehicles by rail and truck. Obviously this model can also be applied in the multimodal public transit context with some minor modifications to account for service headways and line frequency (e.g., Guan et al., 2006).

One variant that is highly relevant to the sharing economy is Xu et al.'s (2015) work on a user equilibrium model with capacitated ridesharing. In this variant, agents choose to be solo drivers, ridesharing drivers, or passengers, in getting from one zone to another. The decision for travelers is not

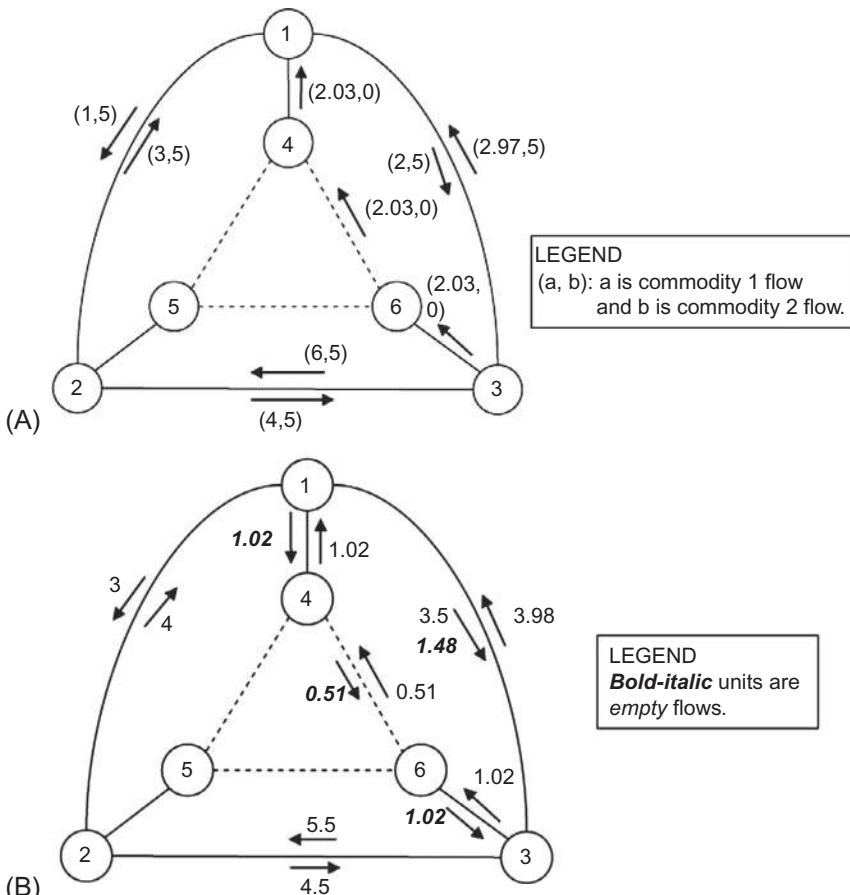


Fig. 3.5 Illustration of freight assignment of (A) path-based commodity flows, and simultaneously, (B) link-based cyclic truck and rail vehicular flows. (Source: Chow et al. 2014)

just which path to take, but also which of the three roles to take on. Due to the complexity of the model, the authors resort to using a VI formulation because the problem cannot be formulated with constrained optimization.

3.2.4 Data and Software

Over the years, researchers have proposed several different algorithms to solve the traffic assignment problem and its variants. Computational efficiency is important because transportation planners may have to run numerous scenarios for large-scale networks that encompass thousands of nodes and links.

For these research efforts, it is important to have consistent benchmarks for proper comparisons to be made between algorithms or to evaluate the scalability of an algorithm over a range of instances. Benchmark networks exist for this purpose. One of the classic benchmark networks for evaluating traffic assignment algorithms (and network design algorithms, as we see in [Chapter 7](#)) is the Sioux Falls, SD, network shown in [Fig. 3.6](#). The network was first published in [LeBlanc et al.'s \(1975\)](#) work on applying the F-W algorithm to traffic assignment. A number of other test networks for evaluating traffic assignment and network design algorithms have since been used. A list of these test networks available for download from the website <https://github.com/bstabler/TransportationNetworks> is presented in [Table 3.3](#).

In addition to the test data, there are several existing tools developed over the years to run traffic assignment algorithms. Several of them are listed in [Table 3.4](#). Some of these tools, like EMME, were developed by researchers who contributed significantly to the literature in traffic assignment in the last 40 years (e.g., Mike Florian). One example of a network assignment using one of these tools is shown in [Fig. 3.7](#) of a transit network assignment in TransCAD for the Greater Toronto Area. MATSim is not listed among these tools since its convergence toward a stable network flow solution is based on day-to-day adjustments, which is discussed in [Section 3.5](#).

3.3 FIXED ROUTE TRANSIT ASSIGNMENT

The second major category of network assignment models is exemplified by fixed route transit assignment. What differentiates these types of systems is that a user's performance in a network is not dictated only by the user's choice and the choices of other users (as in the assignment models in [Section 3.2](#)) but also on the operating and information sharing policies of a service operator. For example, a user may choose to travel to a metro station, but which line they end up taking may depend on the availability and frequencies of the lines that pass through and the type of traveler information system used by the operator. A user on a bus may experience different travel times if the bus operator employs stop skipping or bus headway control strategies.

Dependence on operator policies requires two modifications to the conventional assignment modeling framework:

- (1) The need to model performance measures outside of the operator's routes: wait time, access/egress time, transfers, and so on (since

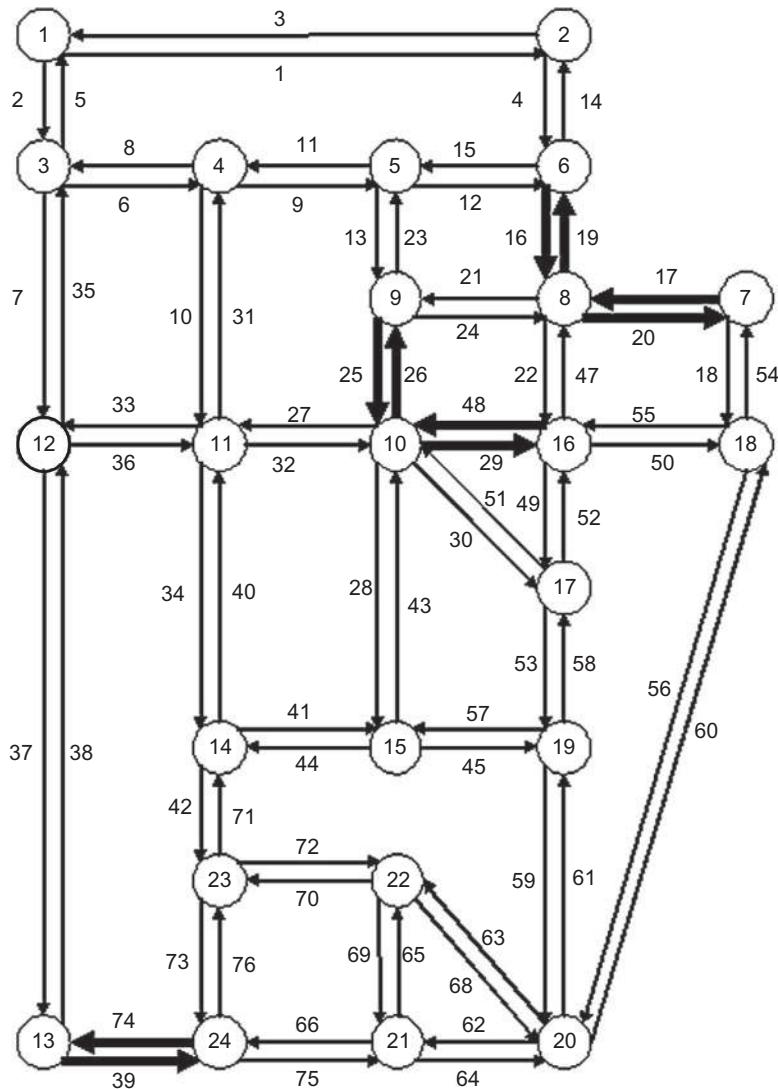


Fig. 3.6 Sioux Falls network with link IDs and link flows (ID/flow) under UE. (Source: Chow and Regan, 2011c.)

performance depends on how the operator's system is overlaid on top of an existing activity network of OD patterns)

- (2) The need to consider hyperpaths, which are user policies for state-dependent shortest path decisions (since performance depends on operator's control of the system state), instead of following an a priori shortest path decision

Table 3.3 Test networks

Network	Zones	Links	Nodes
Anaheim	38	914	416
Austin	7388	18,961	7388
Barcelona	110	2522	1020
Berlin-Center	865	28,376	12,981
Berlin-Friedrichshain	23	523	224
Berlin-Mitte-Center	36	871	398
Berlin-Mitte-Prenzlauerberg-Friedrichshain-Center	98	2184	975
Berlin-Prenzlauerberg-Center	38	749	352
Berlin-Tiergarten	26	766	361
Birmingham-England	898	33,937	14,639
Braess-Example	2	5	4
Chicago-Sketch	387	2950	933
Eastern-Massachusetts	74	258	74
Gold Coast	1068	11,140	4807
Hessen-Asymmetric	245	6674	4660
Philadelphia	1525	40,003	13,389
Sioux Falls	24	76	24
Sydney	3264	75,379	33,113
Terrassa-Asymmetric	55	3264	1609
Winnipeg	147	2836	1052
Winnipeg-Asymmetric	154	2535	1057
Chicago-regional	1790	39,018	12,982

(Source: <https://github.com/bstabler/TransportationNetworks.>)**Table 3.4** Commercial software and open source tools for traffic assignment

Type	Software	Provider	Link
Commercial	EMME	INRO	https://www.inrosoftware.com/en/products/emme/
	TransCAD	Caliper	http://www.caliper.com/
	Cube	Citilabs	http://www.citilabs.com/software/cube/
Open source	AequilibraE	P. Camargo	http://aequilibrae.com/
	Origin-Based Assignment	H. Bar-Gera	http://www.openchannelsoftware.org/projects/Origin-Based_Assignment/
	TrafficAssignment.jl	C. Kwon	https://github.com/chkwon/TrafficAssignment.jl
	NeXTA	X. Zhou	https://code.google.com/archive/p/nexta/

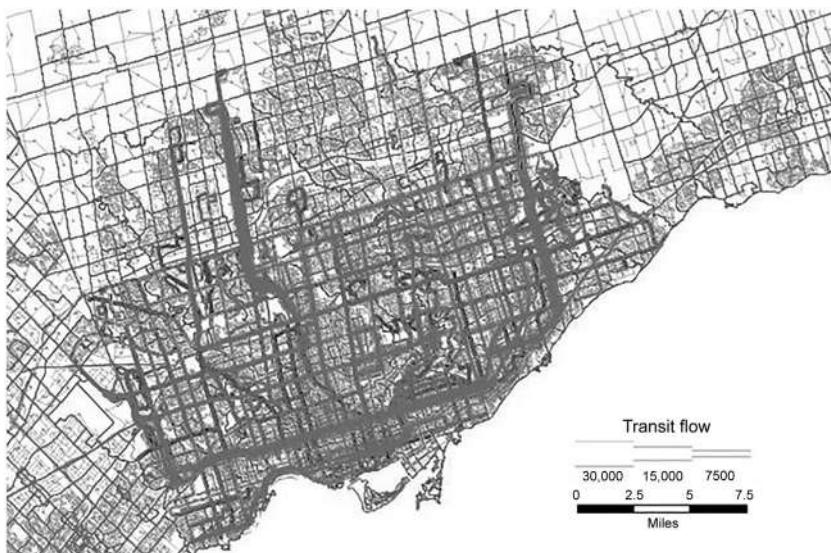


Fig. 3.7 Illustration of network assignment in Greater Toronto Area using TransCAD.
(Source: [Lorion et al., 2014.](#))

Transit assignment models are used by transit operators to evaluate and compare the ridership and system performance for different kinds of transit alternatives. For example, they can be used to compare social impacts of different route designs, different stop/station locations, different headways and timetables, or different fare payment schemes. There are two general approaches to model transit assignment: “frequency-based” assignment and “schedule-based” assignment. Each is discussed in a separate subsection, followed by a discussion of recent variants.

3.3.1 Frequency-Based Assignment

Unlike road systems, fixed route transit systems move passengers from one point to another through lines. Each line is served by a series of vehicles that are separated by headways. The difference in network representation is illustrated by [De Cea and Fernández \(1993\)](#) and shown in Fig. 3.8. Whereas a 4-node road network in Section 3.2 would simply have four links connecting them, the transit network here may have six different lines. A person boarding line L1 at node N1 may alight at node N3, whereas another person boarding L1 at node N2 would have to wait at the station until a vehicle arrives. The person may also choose between L1, L5, and L6 to get to N3. If a vehicle on L1 is at capacity when arriving at N2, the person waiting

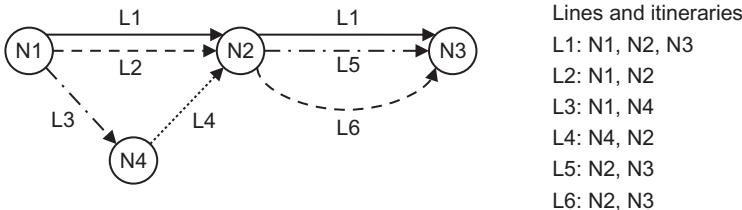


Fig. 3.8 Illustration of transit lines operating over a 4-node network.

there would not be able to enter at that time and will have to wait for the next vehicle.

The two types of transit assignment deal with how time is considered. In frequency-based assignment, time is ignored for simplicity. Timetables are therefore not kept in the model. Only service frequency and headway are incorporated through probabilistic distributions of a wait cost imposed on passengers arriving at a node.

Even with only wait cost distribution without time, a transit assignment model can consider the problem of information availability. In practice, when a transit service does not provide real time vehicle arrival information at a stop (under Logical Architecture Process 4-Manage Transit” in Fig. 1.1) or when there is a high degree of uncertainty in actual vehicle arrival time, travelers do not have complete information about travel times on each route. In this incomplete information setting, travelers have a perception of arrival time and of the subsequent travel time on a specific transit line.

[Chriqui and Robillard \(1975\)](#) called this the “common lines problem” to reflect a behavioral principle that travelers use when traveling in transit networks with incomplete information about the scheduled arrival times ([Principle 3.3](#)).

Consider an example shown for three parallel lines in Fig. 3.9, where each line l has an in-vehicle travel time t_l (minutes) and a service frequency f_l (per minute).

Principle 3.3

([Chriqui and Robillard, 1975](#)). *Travelers at a station in a public transit network with multiple lines and incomplete information about arrival times would choose a subset of lines from which to board the first arriving vehicle, such that undesirable lines are excluded and expected travel time (waiting time plus in-vehicle travel time) is minimized. This behavioral assumption is called the “common lines problem.”*

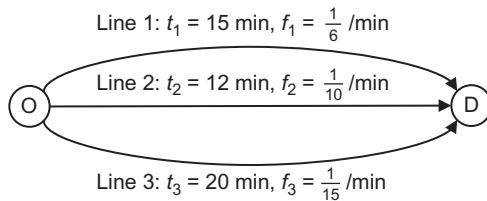


Fig. 3.9 Sample network to illustrate common lines problem.

Let us assume a uniformly distributed random passenger arrival at node O and independently distributed headways for vehicles along the three lines. The average wait time for such a passenger considering only a single line is equal to $W[l] = \frac{\alpha}{f_l}$, where $\alpha = 1$ if the vehicle interarrival times are exponential distributed and $\alpha \approx 0.5$ if the interarrival times are constant at $\frac{1}{f_l}$ (see Spiess and Florian, 1989). Assuming $\alpha = 1$, an a priori shortest path considering average travel time and average wait time would be to take line 1 for average travel time of 21 min (6 min wait, 15 min in-vehicle).

For the case of taking the first vehicle that arrives among a number of lines with different frequencies that operate independently of one another, let us define the set of lines at node i as \bar{A}_i^+ s. The wait time $W[\bar{A}_i^+]$ and probability of taking line $P_l[\bar{A}_i^+]$ under this adaptive strategy are shown in Eq. (3.16). The common lines problem for this example is shown in Exercise 3.8.

$$W[\bar{A}_i^+] = \frac{\alpha}{\sum_{l \in \bar{A}_i^+} f_l} \quad (3.16a)$$

$$P_l[\bar{A}_i^+] = \frac{f_l}{\sum_{l' \in \bar{A}_i^+} f_{l'}} \quad (3.16b)$$

Exercise 3.8

Solve the common lines problem for the example in Fig. 3.9, assuming $\alpha = 1$.

Now we can see how the common lines problem works. If a traveler considers all three lines in Fig. 3.9 as possible lines to take upon being the first to arrive at node O, then the average wait time is

$$W[\bar{A}_O^+ = \{1, 2, 3\}] = \left(\frac{\frac{1}{6}}{\frac{1}{6} + \frac{1}{10} + \frac{1}{15}} \right) = 3.000 \text{ min until any one line}$$

arrives. The probability of that line being line 1, 2, or 3 is equal to $P_1[\bar{A}_O^+ = \{1, 2, 3\}] = 0.500$, $P_2[\bar{A}_O^+ = \{1, 2, 3\}] = 0.300$, and $P_3[\bar{A}_O^+ = \{1, 2, 3\}] = 0.200$. The expected total travel time $E[u_O[\bar{A}_O^+ = \{1, 2, 3\}]]$ under this strategy of taking any of the three lines is therefore equal to: $E[u_O[\bar{A}_O^+ = \{1, 2, 3\}]] = 3 + 0.5(15) + 0.3(12) + 0.2(20) = 18.100$ min. By allowing adaptation to information (seeing when a vehicle arrives first), the passenger can improve the expected travel time from the a priori strategy.

Furthermore, if the passenger removes line 3 from consideration at all to obtain the set of common lines $\bar{A}_O^+ = \{1, 2\}$, the new wait time is $W[\bar{A}_O^+ = \{1, 2\}] = 3.750$ min, probabilities of taking lines are $P_1[\bar{A}_O^+ = \{1, 2\}] = 0.625$ and $P_2[\bar{A}_O^+ = \{1, 2\}] = 0.375$, and the expected travel time by considering only lines 1 and 2 is $E[T[\bar{A}_O^+ = \{1, 2\}]] = 3.750 + 0.625(15) + 0.375(12) = 17.625$ min. Clearly removing line 3 leads to a better result. The common lines problem is to find the optimal set \bar{A}^* such that the minimum expected travel time can be achieved. By adopting a common lines problem principle, we assume that travelers in public transit systems with incomplete information behave in this manner.

In a network setting, solving the common lines problem is nontrivial. One way is to systematically break out the shortest path into all the possible paths depending on the realized arrival times through a network. This stochastic path based on incomplete arrival time information is called a hyperpath (Nguyen and Pallotino, 1988). While the hyperpath shares similar qualities to the stochastic network loading shown in Section 3.2.3, the latter comes from heterogeneity in travelers' perceived travel times whereas the former derives the probabilistic path flows from incomplete information about the system.

Spiess and Florian (1989) proposed a way to find a hyperpath (which they called an “optimal strategy”) by using an algorithm similar to Dial’s algorithm.

**Algorithm 3.5 Spiess and Florian's (1989) Optimal Strategy
Algorithm to Find the Hyperpaths and Optimal Common Lines
for an Uncongested Transit Network**

Inputs: Transit network $G[N, A]$ where A is a link-based representation of the network that includes link frequencies f_a , link costs c_a , and demand of g_i for all OD pair (i, s) , $i \neq s$.

1. Initialize expected travel times $u_i := \infty$, $i \in N \setminus \{s\}$, $u_s := 0$, frequencies $f_i := 0 \forall i$, set of links that have not yet been examined $S := A$, set of links in optimal strategy $\bar{A} := \emptyset$
2. While $S \neq \emptyset$,
 - a. Find $a = (i, j) \in S$ which satisfies $u_j + c_a \leq u_i + c_a$, $a' = (i', j') \in S$
 - b. $S := S \setminus \{a\}$
 - c. If $u_i \geq u_j + c_a$ then
 - i. $u_i := \frac{f_i u_i + f_a (u_j + c_a)}{f_i + f_a}$
 - ii. $f_i := f_i + f_a$
 - iii. $\bar{A} := \bar{A} \cup \{a\}$
3. $V_i := g_i \forall i$
4. For $\forall a \in A$ in decreasing order of $(u_j + c_a)$
 - a. If $a \in \bar{A}$ then
 - i. $v_a := \frac{f_a}{f_i} V_i$
 - ii. $V_j := V_j + v_a$
 - b. Else $v_a := 0$

Outputs: Link flows v_a

Algorithm 3.5 is applied to the earlier sample network modified to have frequencies. The algorithm is illustrated with Exercise 3.9.

Exercise 3.9 Apply Algorithm 3.5 to assign the demand onto the sample network in Fig. 3.10. How does this compare against the a priori shortest path assignment?

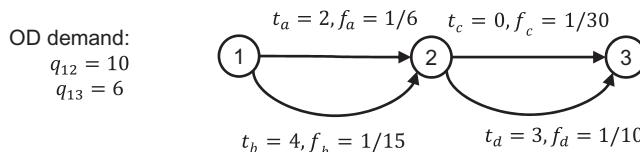


Fig. 3.10 Sample network for Exercise 3.9.

1. For (1,2):

- a. Initiate: $u_1 = \infty$, $u_2 = 0$, $f_1 = f_2 = 0$, $S = \{a, b\}$
- b. Determine $u_j + c_a$: $\{a\} : 0 + 2 = 2$, $\{b\} : 0 + 4 = 4$. We start with node a : $S = \{b\}$.

c. Since $u_1 > 0 + 2$, update: $u_1 := \frac{0(\infty) + \frac{1}{6}(0+2)}{1} = \frac{1}{1} = 8$,
 $f_1 = 0 + \frac{1}{6} = \frac{1}{6}$, $\bar{A} = \{a\}$

- d. For node b : $S = \emptyset$. since $u_1 = 8 > 0 + 4$, we update:

$$u_1 := \frac{\frac{1}{6}(8) + \frac{1}{15}(0+4)}{\frac{1}{6} + \frac{1}{15}} = \frac{48}{7} = 6.857, f_1 = \frac{1}{6} + \frac{1}{15} = \frac{7}{30}, \bar{A} = \{a, b\}$$

- c. $V_1 = 10$, $V_2 = -10$. We start with $\{b\}$:

$$\nu_{12,b} = \frac{\frac{1}{15}}{\left(\frac{7}{30}\right)} 10 = \frac{20}{7} = 2.857, V_2 = -10 + \frac{20}{7} = -\frac{50}{7}$$

d. Now for $\{a\}$: $\nu_{12,a} = \frac{\left(\frac{1}{6}\right)}{\left(\frac{7}{30}\right)} 10 = \frac{50}{7} = 7.143, V_2 = -\frac{50}{7} + \frac{50}{7} = 0$.

2. For (1,3):

- a. Initiate: $u_1 = u_2 = \infty$, $u_3 = 0$, $f_1 = f_2 = f_3 = 0$, $S = \{a, b, c, d\}$
- b. Determine $u_j + c_a$: $\{a\} : \infty + 2 = \infty$, $\{b\} : \infty + 4 = \infty$, $\{c\} : 0 + 0 = 0$, $\{d\} : 0 + 3 = 3$. We start with node c : $S = \{a, b, d\}$.

c. Since $u_2 > 0 + 0$, update: $u_2 := \frac{0(\infty) + \frac{1}{30}(0+0)}{1} = \frac{1}{1} = 30$,

$$f_2 = 0 + \frac{1}{30} = \frac{1}{30}, \bar{A} = \{c\}.$$

- d. Since $u_2 = 30$, we update $u_j + c_a$ for those impacted: $\{a\} : 30 + 2 = 32$, $\{b\} : 30 + 4 = 34$. The next smallest is $\{d\}$. $S = \{a, b\}$

e. Since $u_2 = 30 > 0 + 3$, update: $u_2 := \frac{30\left(\frac{1}{30}\right) + \frac{1}{10}(0+3)}{\frac{1}{30} + \frac{1}{10}} = 9.750$,

$$f_2 = \frac{1}{30} + \frac{1}{10} = \frac{2}{15}, \bar{A} = \{c, d\}.$$

- f. Since $u_2 = 9.750$, we update $u_j + c_a$ for those impacted: $\{a\} : 9.750 + 2 = 11.750$, $\{b\} : 9.750 + 4 = 13.750$. The next smallest is $\{a\}$. $S = \{b\}$

g. Since $u_1 > 9.750 + 2$, we update: $u_1 := \frac{0(\infty) + \frac{1}{6}(9.750+2)}{1} = 17.750$,

$$f_1 = 0 + \frac{1}{6} = 1/6, \bar{A} = \{a, c, d\}$$

h. Now node b : since $u_1 = 17.750 > 9.750 + 4$, we update:

$$u_1 := \frac{\frac{17.750}{6} + \frac{1}{15}(9.750 + 4)}{\frac{1}{6} + \frac{1}{15}} = 16.607, \quad f_1 = \frac{1}{6} + \frac{1}{15} = \frac{7}{30},$$

$$\overline{A} = \{a, b, c, d\}$$

i. $V_1 = 6$, $V_2 = 0$, $V_3 = -6$. We start with $\{b\}$: $v_{13,b} = \frac{1}{\left(\frac{7}{30}\right)} 6 = \frac{12}{7} = 1.714$, $V_2 = 0 + \frac{12}{7} = \frac{12}{7}$

j. Now for $\{a\}$: $v_{13,a} = \frac{1}{\left(\frac{6}{30}\right)} 6 = \frac{30}{7} = 4.286$, $V_2 = \frac{12}{7} + \frac{30}{7} = 6$

k. Then for $\{d\}$: $v_{13,d} = \frac{1}{\left(\frac{10}{15}\right)} 6 = 4.5$, $V_2 = -6 + 4.5 = -1.5$

l. Then for $\{c\}$: $v_{13,c} = \frac{1}{\left(\frac{30}{15}\right)} 6 = 1.5$, $V_2 = -1.5 + 1.5 = 0$.

3. Sum up the flows: $v_a = \frac{50}{7} + \frac{30}{7} = 11.429$, $v_b = \frac{20}{7} + \frac{12}{7} = 4.571$, $v_c = 1.5$, $v_d = 4.5$.

4. The total system travel time is: $TSTT = 10(4.286 + 1.429 + 1.143) + 6(4.286 + 7.5 + 0.357 + 2.679 + 0.286 + 1.5) = 168.228$ passenger min

5. By comparison, the a priori shortest path assignment (along links a and d) would have the following TSTT: $10(8) + 6(8 + 13) = 206$ passenger min

Algorithm 3.5 ignores congestion effects. In crowded transit networks, there is a crowding effect that should not be ignored. [De Cea and Fernández \(1993\)](#) proposed a UE model that accounted for the common lines problem but were not able to guarantee a unique solution. [Wu et al. \(1994\)](#) formulated a VI problem for a congested transit network equilibrium with the underlying hyperpath-based network loading from [Spiess and Florian \(1989\)](#). They proved conditions for convergence of a gap function-based algorithm to the UE. [Lam et al. \(1999\)](#) proposed an SUE model for a transit network, although their mathematical programming approach does not assume travelers behave under a hyperpath

approach that accounts for common lines problem behavior. Chin et al. (2016) applied the method, modified with station-based fare pricing, to evaluate distance-based fare schemes. Cominetti and Correa (2001) studied UE conditions where common lines behavior that also considered frequencies that depend on volume, $f_i[v]$, based on queue delay. Kurauchi et al. (2003) proposed a hyperpath-based assignment model with capacity constraints (no congestion effects). A heuristic was proposed to solve the UE model with common lines behavior and capacity constraints (Cepeda et al., 2006).

The VI solution algorithm from Wu et al. (1994) is presented here in **Algorithm 3.6**.

Algorithm 3.6 Symmetric Linearization Algorithm From Wu et al. (1994) for Transit Equilibrium Assignment Problem

Inputs: Transit network $G[N, A]$ with demand W , where link cost $s[v]$ is a function of link flow v , δ is a link-hyperpath incidence matrix, and ϵ is a tolerance.

1. Initialize a feasible hyperpath flow h^0 and link flow v^0 , and initial hyperpath set $K_{rs}^0 \forall (r, s) \in W$. Set $l = 1$.
2. For each $(r, s) \in W$, update link costs $s[v^{l-1}]$, find shortest hyperpath k_l using [Algorithm 3.5](#) and compute $GAP[h^{l-1}]$, where

$$GAP[h] = \min_{x \in \Omega} S[h]^T (h - x) \quad (3.17)$$

where the k th row in $S[h]$ is determined by Eq. (3.18).

$$S_k[h] = (\delta_k)^T (s[v] + w_k) \quad (3.18)$$

w_k is obtained from Eq. (3.16a) for each link a in hyperpath k ,

Ω is the set of feasible hyperpath flows,

$S[h]$ is the hyperpath cost with hyperpath flows h .

Let $K_{rs}^l = K_{rs}^{l-1} \cup k_l$.

3. If $GAP[h^{l-1}] \leq \epsilon$, stop.
4. Based on $s[v^{l-1}]$, update $B_k[h^{l-1}]$ and $S_k[h^{l-1}]$ for all $k \in K_{rs}^l$. $B[h]$ is the diagonal of the Jacobian of S evaluated at h^l .
5. Solve Eq. (3.19) and let h^l be its solution.

$$\min_{h \in \Omega} (h - h^l)^T S[h^l] + \frac{1}{2\gamma} (h - h^l)^T B[h^l] (h - h^l) \quad (3.19)$$

where γ is a positive parameter related to step size.

6. Compute $v^l = \delta h^l$ and update $s[v^l]$. Let $l = l + 1$ and go to Step 2.

Outputs: Link flows v

To illustrate this algorithm, let us take a look at the result of [Exercise 3.9](#). This is illustrated in [Exercise 3.10](#).

Exercise 3.10

Using the results of [Exercise 3.9](#) as the initialized feasible path and link flows, assuming link cost functions shown in [Fig. 3.11](#), update the link cost and hyperpath cost, and explain why the GAP function is zero.

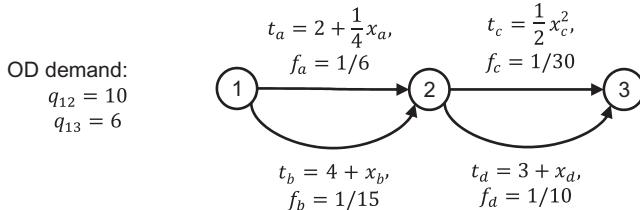


Fig. 3.11 Sample network with updated link cost functions for [Exercise 3.10](#).

- Referring to the link-path incidence matrix in [Exercise 3.4](#), $v^0 = [11.429, 4.571, 1.5, 4.5]$, hyperpath 1 $\{[0.714, 0.286, 0, 0, 0, 0]\} \in K_{12}^0$, hyperpath 2 $\{[0, 0, 0.179, 0.536, 0.071, 0.214]\} \in K_{13}^0$, and hyperpath flow $h^0 = [10, 6]$.
- Update link costs: $t[v^0] = [4.857, 8.571, 1.125, 7.500]$. Use this and the frequency delays $w = [4.286, 11.786]$ for the two hyperpaths to compute new path costs $S[h^0] = [10.214, 23.610]$.
- Since the initial hyperpaths already cover the whole network and the frequencies do not change with respect to the flows, rerunning [Algorithm 3.5](#) would not generate any new hyperpaths. As a result, there is only a single hyperpath for each $(r, s) \in W$, and this is the equilibrium solution.

3.3.2 Schedule-Based Assignment

Frequency-based assignment is not the only approach to assign passengers to transit systems. The methodology ignores arrival time at a destination. Since fixed route transit systems operate on a timetable, for some policies and planning purposes it is important to assign passengers in a way to account for schedule delay costs. We discuss schedule delay in more detail in [Section 3.4](#), but suffice to say that a commuter that is late by 1 min may value that time by 1.9–11.1 times >1 min of travel time ([Small, 1982](#)). Additionally, frequency-based approaches with hyperpath modeling assume there is

limited information available to passengers, resulting in the probabilistic assignment onto common lines. In the case where traveler information systems are available with varying degrees of information, a schedule-based approach may be more appropriate (Nuzzolo et al., 2001).

Tong and Wong (1999) proposed a schedule-based assignment model in which passengers arrive in the system randomly based on temporal distributions, after which they are deterministically assigned onto the network as all-or-nothing assignment. Nuzzolo et al. (2001) employed a doubly dynamic setting (within day and day to day) to obtain a dynamic equilibrium assignment. Hamdouch and Lawphongpanich (2008) developed a schedule-based assignment model that also incorporated “travel strategies,” which are similar to the hyperpaths from the frequency-based assignment but are not required to end at the same destination node. Schedule-based assignment makes use of time-expanded networks so it is possible that a user may end up arriving at a location later or earlier than desired, reaching a different destination time-expanded node. At each node, there is a rule for a preferred downstream node for a user class. This type of schedule-based system is illustrated in Fig. 3.12.

In addition to the time-expanded network, each OD pair is also divided into groups with passengers having the same desired arrival time intervals. Network loading can occur assuming first-come-first-serve or by random loading. The latter can occur if there is a huge crowd without any queue organization. The equilibrium is expressed as a VI and solved by an MSA algorithm. Uniqueness is not guaranteed. An illustration of an equilibrated solution is shown in Fig. 3.13.

3.3.3 Transit Assignment Variants

In recent years a number of other variant transit assignment methods have been proposed, primarily to deal with complexities of ITS-oriented traveler information systems in a simulation-based setting, and considering integration with activity-based scheduling (which we will discuss more in Chapter 4). For example, Wahba and Shalaby (2009) developed an agent-based simulation of schedule-based transit assignment considering both within-day and day-to-day learning. The modeling framework, called MILATRAS, is shown in Fig. 3.14.

Khani (2013) developed another simulation-based scheduled-based transit assignment model called FAST-TrIPs. The code for this model is open source (<https://github.com/akhani/FAST-TrIPs>) and has been

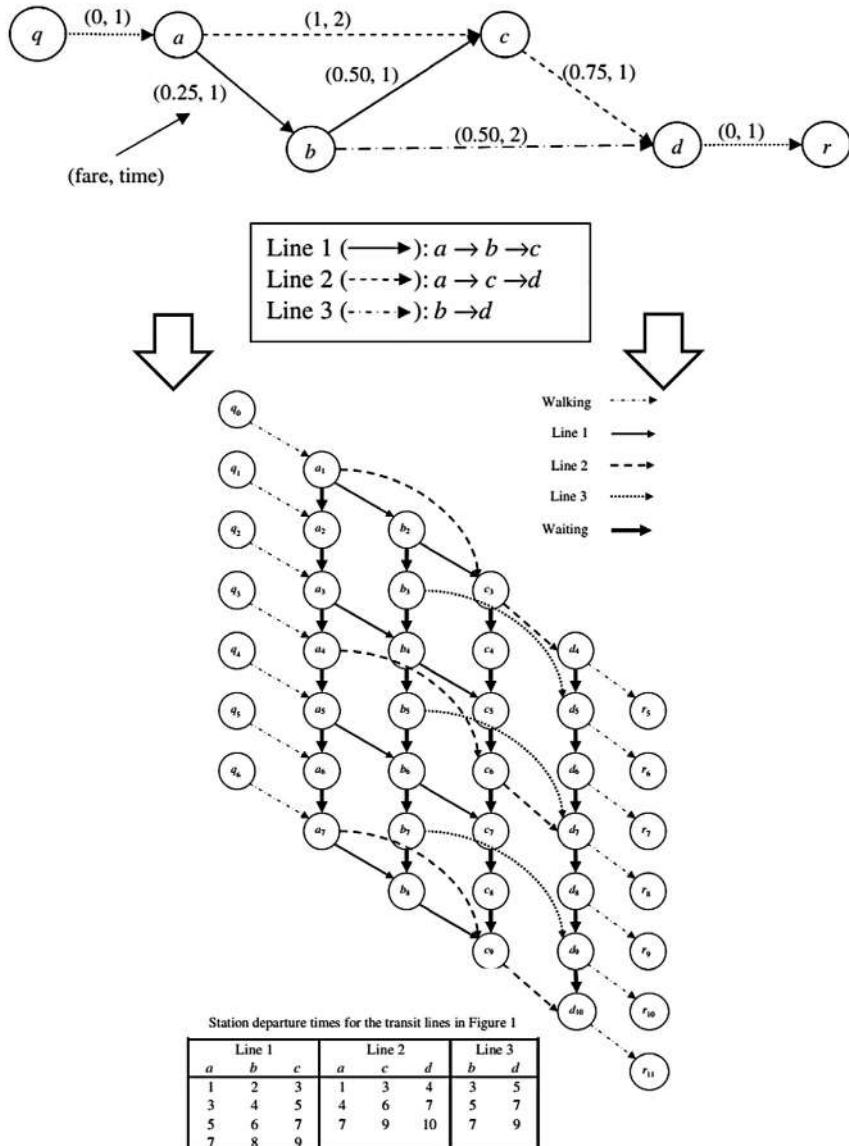


Fig. 3.12 Illustration of time-expanded network representation of a transit network with three lines. (Source: [Hamdouch and Lawphongpanich, 2008](#).)

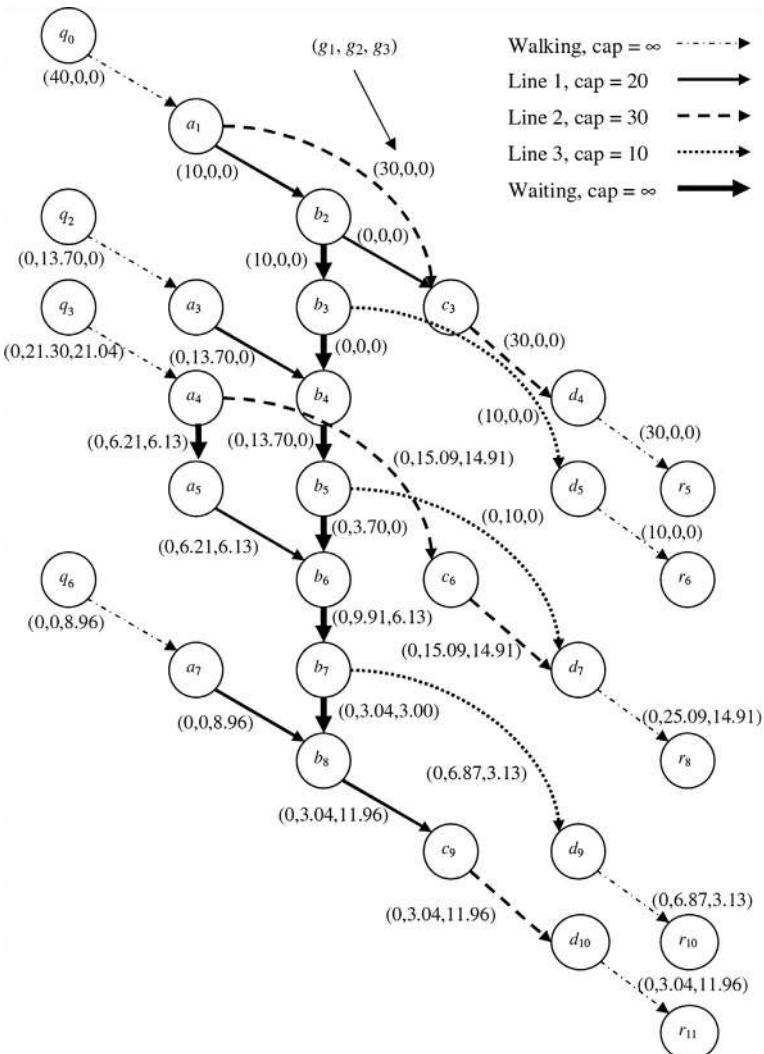


Fig. 3.13 Equilibrium link flows for each of three passenger groups. (Source: Hamdouch and Lawphongpanich, 2008.)

implemented by the Metropolitan Transportation Commission (MTC), the metropolitan planning organization (MPO) for the San Francisco Bay Area (<http://fast-trips.mtc.ca.gov/>). Verbas et al. (2016) proposed a simulation-based assignment algorithm that uses a hyperpath assignment in an upper level and a simulation of traveler patterns in the lower level, analogous to simulation-based dynamic traffic assignment. The comprehensive

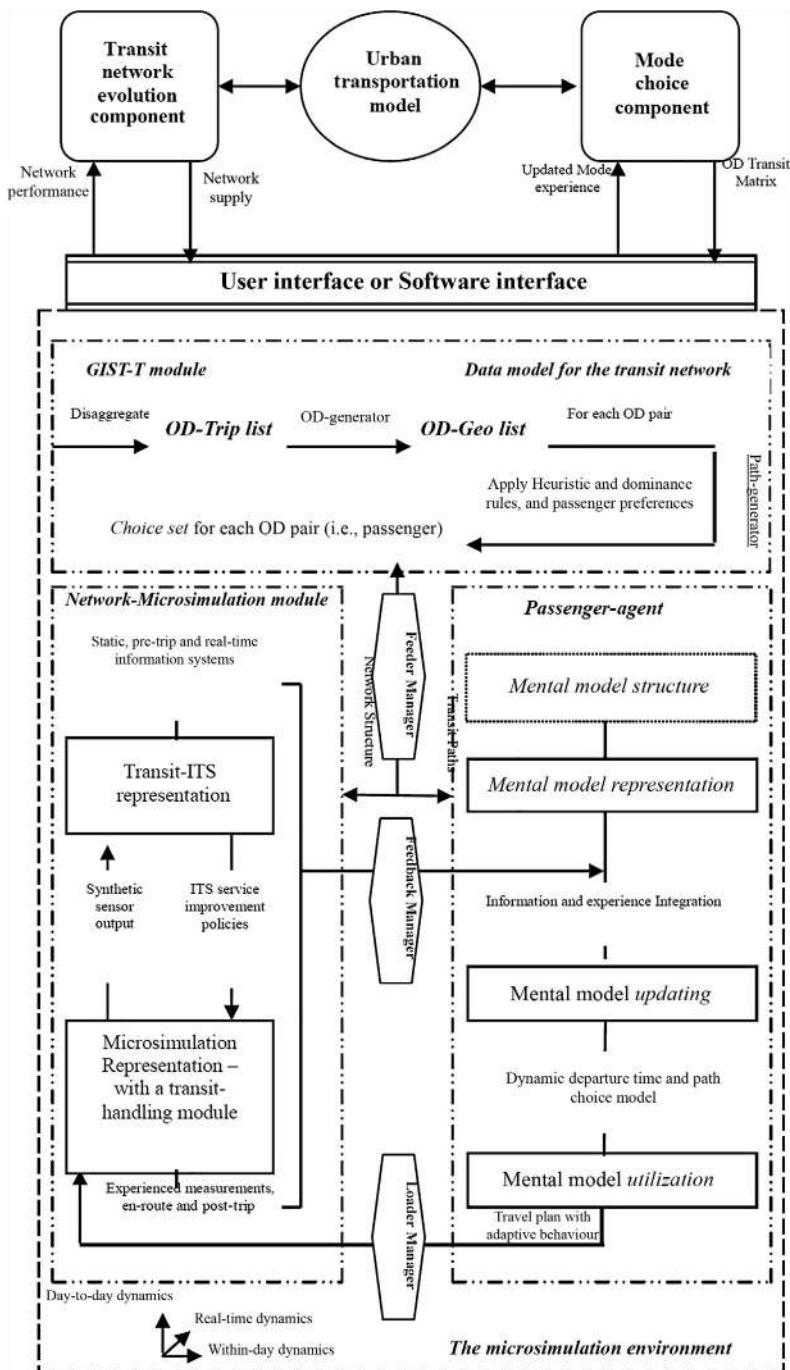


Fig. 3.14 MILATRAS model framework. (Source: Wahba and Shalaby, 2009.)

agent-based activity simulator MATSIM also has a transit assignment component (Rieser, 2010). The assignment is based on a simple shortest path without consideration of common lines or strategies.

3.4 OTHER EQUILIBRIA

While many transport policies relate to how people interact with the road infrastructure, there are also many policies that relate to other aspects of supply-demand equilibration in the MFG framework. Three of them are discussed in this section.

The first deals with how people choose departure time through a common bottleneck. In this setting, time of day choice is the behavioral consideration. This type of equilibrium has become more important in recent years because of the emphasis on activity scheduling and on real-time mobility applications that impact time use.

The second deals with equilibration of space usage for parking vehicles at a common destination. This problem has become more relevant in recent years with the increasing crowdedness of downtown areas and the arrival of new technologies and business models (autonomous vehicles, shared parking, dynamic pricing apps, and increasing usage of on-street parking spaces for truck and courier deliveries).

The third deals with equilibration of on-demand for-hire services like taxis and ridesourcing companies like Uber and Lyft. The equilibration in these cases deal with congestion effects that occur with matching drivers with passengers and is also gaining significant interest in recent years because of advances in information and communications technologies (ICTs) that popularize these mobility services.

3.4.1 Vickrey Morning Commute Problem

How people allocate their time is a fundamental economic question that has been studied since Becker (1965). In that work, time use is regarded as a constrained utility maximization problem in which people are limited by a time budget. Under this paradigm, travel preferences are not just guided by congestion effects (i.e., leaving earlier might avoid traffic and thus result in a shorter commute time) but are also influenced by users' scheduling preferences. Conducting one activity at one time can result in consumed utility that differs from conducting the same activity at a different time of day. This implies that schedule delay alone can influence departure time of travelers if travel flows are capacitated (Hendrickson and Kocur, 1981) and the effect can vary

(Newell, 1987; Arnott et al., 1988) from one individual to another due to heterogeneity in users' travel costs, activity agendas, and scheduling preferences (e.g., value of being late can vary over the population as shown by Small, 1982).

The analysis of departure time allocation for a population began with Vickrey (1969), who studied this problem as a single deterministic queue with flow capacity and has been called a “bottleneck model” (Arnott et al., 1990a) or “morning commute problem” (Arnott et al., 1990b). The existence (Smith, 1984b) and uniqueness (Daganzo, 1985) of a distribution through the single bottleneck have been proven. The case with elastic demand has also been formulated (Arnott et al., 1993). Because there are so many variants to the model, only the basic homogeneous version in Arnott et al. (1990a) is presented here with illustration.

There are N identical commuters traveling to work downtown via a common bottleneck in which at most s cars can traverse per unit time. The commuters have a desired arrival time to work, t^* , in which a penalty of β (γ) is incurred per unit time they are early (late). A penalty of α is incurred per unit travel time, where $\alpha > \beta$ (or people would simply leave very early when there is no bottleneck formed). When this flow capacity is exceeded at the bottleneck, a queue develops. The travel time is defined as shown in Eq. (3.20).

$$T(t) = T^f + T^w[t] \quad (3.20)$$

where T^f is a travel time when there is no queue and can be set to zero without loss of generality for the bottleneck analysis, $T^w[t]$ is the additional waiting time at the queue, and t is the departure time from home. The length of queue $D[t]$ is defined in Eq. (3.21).

$$D(t) = \int_{\hat{t}}^t r[\tau] d\tau - s(t - \hat{t}) \quad (3.21)$$

where \hat{t} is the time at which the queue was last zero and $r[t]$ is the departure rate at time t . The time derivative is $D'[t] = r[t] - s$ when $D[t] > 0$. Waiting time is defined in Eq. (3.22).

$$T^w[t] = \frac{D[t]}{s} \quad (3.22)$$

Under UE, the rate of arrival to work is fixed at the capacity s , whereas departures from work will distribute around a peak queue length that occurs at time \tilde{t} as shown in Fig. 3.15. The first and last departures face no queue but have essentially equivalent costs incurred in terms of being early or late to work. The times are defined as t_q (first departure) and t_q' (last departure).

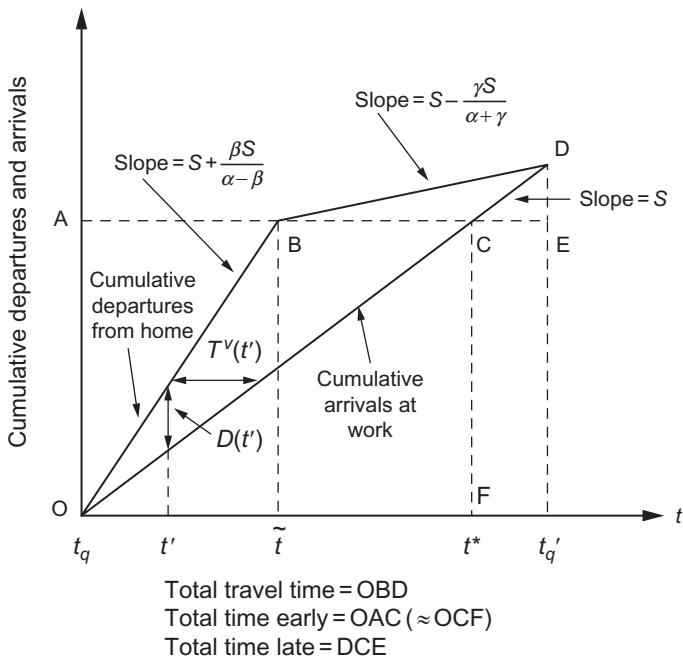


Fig. 3.15 Queue represented by cumulative departures from home and arrivals at work.
(Source: Arnott et al., 1990a.)

The equilibrium occurs when no traveler can unilaterally change their departure time without being worse off. This is interpreted as the marginal benefit from postponing departure by a unit of time being equal to the marginal cost. Based on this, the slopes of the home departure rate can be found to be Eq. (3.23).

$$r(t) = \begin{cases} s + \frac{\beta s}{\alpha - \beta}, & t \in [t_q, \tilde{t}] \\ s - \frac{\gamma s}{\alpha + \gamma}, & t \in (\tilde{t}, t_{q'}) \end{cases} \quad (3.23)$$

For N travelers, a system of three equations related to conservation of demand, boundary conditions, and peak queue length condition is solved to obtain the departure times for this piecewise linear queue distribution in Eq. (3.24).

$$t_q = t^* - \left(\frac{\gamma}{\beta + \gamma} \right) \left(\frac{N}{s} \right) \quad (3.24a)$$

$$t_{q'} = t^* + \left(\frac{\beta}{\beta + \gamma} \right) \left(\frac{N}{s} \right) \quad (3.24b)$$

$$\tilde{t} = t^* - \left(\frac{\beta\gamma}{\alpha(\beta+\gamma)} \right) \left(\frac{N}{s} \right) \quad (3.24c)$$

The total travel cost under equilibrium with no toll, TC^e , is derived by finding the average travel cost for one boundary condition and multiplying it by the whole population since they all have the same travel times. This is shown in Eq. (3.25), graphically represented by OBDEC in Fig. 3.15.

$$TC^e = \left(\frac{\beta\gamma}{\beta+\gamma} \right) \left(\frac{N^2}{s} \right) \quad (3.25)$$

This total cost TC^e is the sum of total travel time TTC^e (area of queue region, OBD) and total schedule delay SDC^e (OCF + CDE) and shown in Eq. (3.26).

$$TTC^e = SDC^e = \left(\frac{\beta\gamma}{2(\beta+\gamma)} \right) \left(\frac{N^2}{s} \right) \quad (3.26)$$

Arnott et al. (1990a) derived a single uniform toll ρ^* and time interval $[t^+, t^-]$ to apply to this population to minimize total costs. Assuming $\gamma > \alpha$ (which is empirically the case as shown by Small, 1982), the optimal toll and the time intervals in which it impacts the population are shown in Eq. (3.27) and in Fig. 3.16. The new start time for departures is t_q^* .

$$\rho^* = \frac{\beta\gamma}{2(\beta+\gamma)} \left(\frac{N}{s} \right) \quad (3.27a)$$

$$t_q^* = t^* - \frac{\gamma}{\beta+\gamma} \left(\frac{N}{s} \right) + \frac{(\gamma-\alpha)\rho^*}{(\beta+\gamma)(\alpha+\gamma)} \quad (3.27b)$$

$$t^+ = t_q^* + \frac{\rho^*}{\beta} \quad (3.27c)$$

$$t^- = t_q^* + \frac{N}{s} - \frac{2\rho^*}{\alpha+\gamma} \quad (3.27d)$$

By setting t^+ as the effective t_q , t^- as the t'_q , and the N as $N - \frac{2\rho^*}{\alpha+\gamma} - s(t^+ - t_q)$, one can obtain the curves using Eq. (3.24c).

Based on toll revenue of $R^* = \rho^* s(t^- - t^+)$, one can derive the SDC^* and TTC^* as shown in Eq. (3.28).

$$SDC^* = \left(1 + \frac{\gamma\beta(\gamma-\alpha)^2}{4(\beta+\gamma)^2(\alpha+\gamma)^2} \right) \left(\frac{\beta\gamma}{2(\beta+\gamma)} \right) \left(\frac{N^2}{s} \right) \quad (3.28a)$$

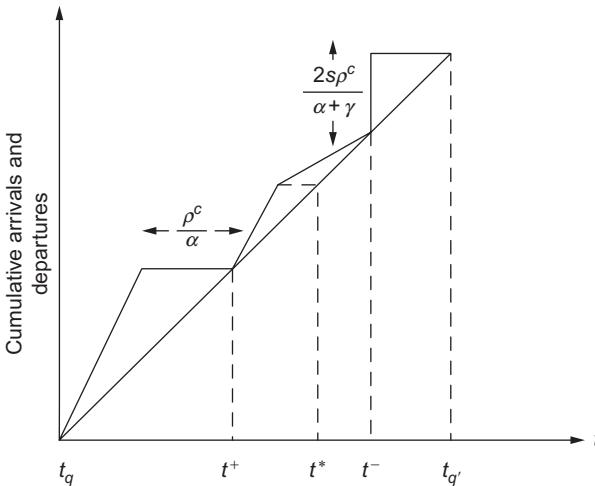


Fig. 3.16 Bottleneck queue with optimal toll applied across optimal time interval.
(Source: Arnott et al., 1990a.)

$$TTC^* = \left(\frac{1}{2} - \frac{(\gamma - \alpha)\beta}{2(\beta + \gamma)(\alpha + \gamma)} - \frac{\gamma\beta(\gamma - \alpha)^2}{4(\beta + \gamma)^2(\alpha + \gamma)^2} \right) \left(\frac{\beta\gamma}{2(\beta + \gamma)} \right) \left(\frac{N^2}{s} \right) \quad (3.28b)$$

To demonstrate this model, consider [Exercise 3.11](#).

Exercise 3.11

Given $\alpha = \$20/h$, $\beta = \$12.20/h$, $\gamma = \$47.50/h$ (based on [Small, 1982](#), and [Arnott et al., 1990a](#), adjusted to \$20/h value of travel time), a bottleneck with flow capacity of $s = 3000$ vph (e.g., a two-lane bridge), and a population of $N = 7500$ trying to get to work at 9 a.m. Plot the bottleneck queue that forms without toll and with a uniform optimal toll. What is the reduction in total travel cost?

For no toll equilibrium:

1. First compute the start time, end time, and peak queue length time:
 $t_q = 7:00$ a.m., $t'_q = 9:30$ a.m., $\tilde{t} = 7:47$ a.m.
2. The departure rates are $r[t < \tilde{t}] = 7692.308$, $r[t > \tilde{t}] = 888.889$. From this we can plot the no toll equilibrium queue.
3. Solving for the performance measures, we get $TTC^e = SDC^e = \$91,001.88$, $TC^e = \$182,003.77$, and total cost per trip is $\frac{TC^e}{N} = \$24.27$.

Under the optimal toll:

4. We compute optimal toll: $\rho^* = \$12.13$.

5. Based on the toll, we compute the new start time and the time range for pricing: $t_q^* = 7:05$ a.m., $t^+ = 8:05$ a.m., $t^- = 9:14$ a.m., $t_{q'}^* = 9:35$ a.m.
6. We compute the horizontal offset and vertical offsets: $\frac{\rho^*}{\alpha} = 0.607$, $\frac{2s\rho^*}{\alpha + \gamma} = 1078.541$.
7. Applying these, we can find the effective population ($N' = 3437.791$) in the subqueue for using Eq. (3.24c) to find the time of the max length in the subqueue: $\tilde{t}' = 8:26$ a.m.
8. Based on these, we can now construct the queue formation under the optimal toll. This is plotted in Fig. 3.17.

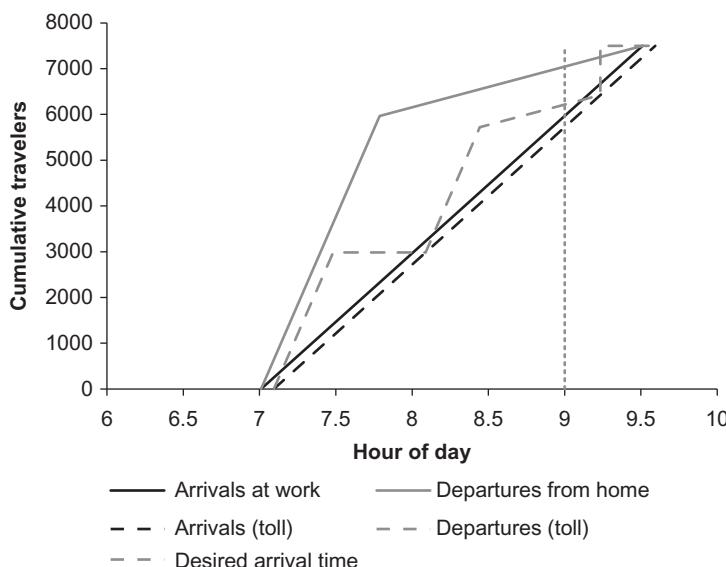


Fig. 3.17 Plot of queues formed from example in Exercise 3.11.

9. Lastly, we compute the performance measures: $C^* = \$41,098.75$, $SDC^* = \$91,615.86$, $TC^* = \$132,714.61$, $\frac{TC^*}{N} = \$17.70$. This is a reduction of 27.1% in average trip cost even after accounting for the cost of the tolls on each trip.

Based on these results, having no toll would lead to a maximum queue that occurs between 7:30 a.m. and 8 a.m. of 3640.075 travelers. By imposing a toll between 8:05 a.m. and 9:14 a.m. of \$12.13 for everybody crossing the bridge to get to downtown, the effect is a reduction of 27.1% in average trip cost. The percent of people who arrive late to work goes up from 20.4% in the no toll equilibrium to 23.7% in the optimal toll scenario. This does not take into account heterogeneity, elastic demand, presence of evening commute, availability of parking, multimodal options, or a host of other extensions that have since been applied to this rich theory.

3.4.2 On-Street Parking and Cruising

The basic decisions to consider in parking are the allocation and pricing of urban space. Only public on-street parking is considered for this section, although private garage parking has also been studied. Early on, Vickrey (1954) considered on-street parking to be a private good for marginal cost pricing purposes. Glazer and Niskanen (1992) demonstrated that this is not the case, as insufficient parking would lead to cruising behavior as a negative externality. Cruising vehicles circulate the neighborhood to find an open parking spot, but doing so increases the stock of vehicles on the street. This behavior can lead to a significant portion of downtown traffic based on data from several cities (Shoup, 2005).

In addition to congestion effects due to cruising, Arnott et al. (1991) used Vickrey's bottleneck model to show that competition for parking spaces can differ between free parking and priced parking. For example, they showed that under free parking drivers tend to naturally park "outwards" by occupying spaces in order of decreasing accessibility. However, this behavior is economically inefficient. Instead, by imposing a distance-based parking pricing it can induce a more efficient "inward" parking behavior. Several extensions of this model include evaluation of parking permits (Zhang et al., 2011a), parking clusters (Qian et al., 2012), parking reservations (Yang et al., 2013), and time-varying parking pricing (Fosgerau and de Palma, 2013).

While the bottleneck model was used to study spatial-temporal preferences of commuters, it lacked consideration for equilibrium with congestion effects of cruising. Arnott and Rowse (1999) introduced a circular city structure to model random parking availability, cruising behavior, and study dynamic parking pricing and information systems. The more complex model structure resulted in nonunique equilibria and ignored traffic congestion, however. Anderson and de Palma (2004) incorporated cruising in a simpler model to draw several conclusions. They noted that the socially optimal parking configuration is independent of the cost of cruising and that priced parking is always better off than unpriced parking. Calthrop and Proost (2006) studied the competition between on-street parking and off-street parking as a Stackelberg game in which the garage operator was a follower.

Arnott and Inci (2006) proposed a new parking model to overcome earlier shortcomings. They considered a downtown grid environment in which the stock of vehicles, parked cars, and cruising cars were all in a saturated steady-state equilibrium in which the balance between influx and outflux of vehicles reached a stable value. The existence and uniqueness

of such an equilibrium was proven under the condition that the system be saturated (or rather, to only apply this model to analyze saturated systems as nonsaturated systems would not have any cruising problems in the first place). The model can be used to analyze cruising, traffic congestion, competition with public transit and off-street parking (Arnott, 2006), differing commuter classes for parking durations and time limit regulations (Arnott and Rowse, 2013), and truck double-parking due to on-street deliveries (Amer and Chow, 2017). A detailed illustration of the model is presented by Arnott and Rowse (2009).

Since Amer and Chow's (2017) model is more generalized than the basic model from Arnott and Inci (2006) with truck deliveries, we present that version here with an example. The model applies to a downtown arterial network shown generically in Fig. 3.18A. The system is considered dynamic

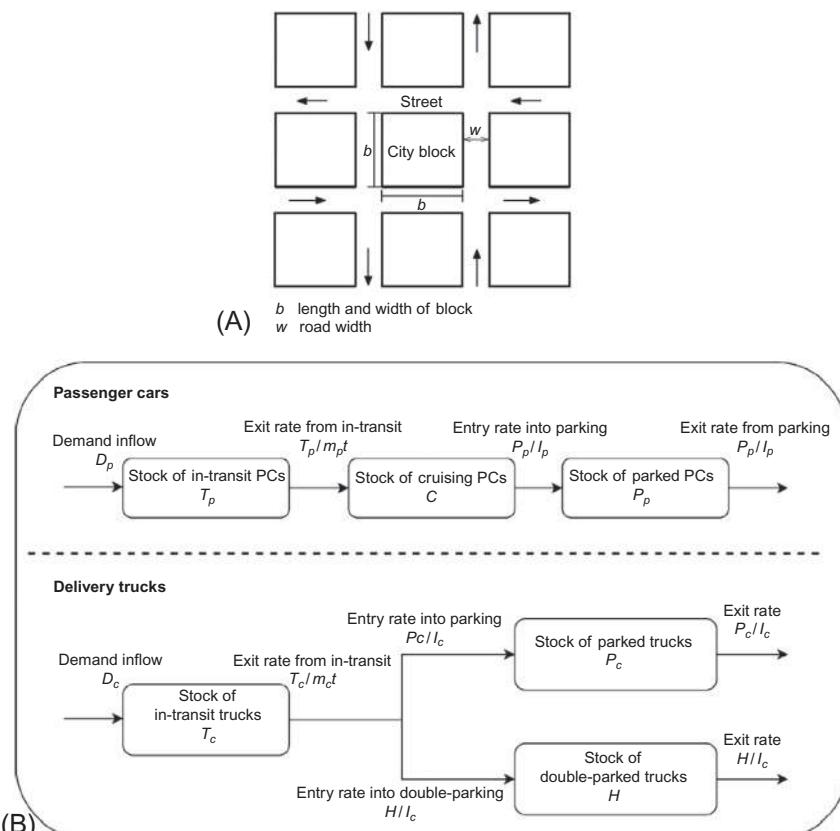


Fig. 3.18 (A) Assumed downtown city block setting and (B) dynamic flows under saturated condition. (Source: Amer and Chow, 2017.)

and analyzed during an increment in time during a saturated steady state. In that state, the inflows and outflows for each state variable shown in Fig. 3.18B need to be conserved.

The D_p is passenger car trip demand per unit time area, T_p is the stock of in-transit passenger cars per unit time area, C is the stock of cruising cars per unit area, P_p is the number of parking spaces (all occupied when saturated) allocated to passenger cars per unit area, D_c is the delivery truck trip demand per unit time area, T_c is the stock of in-transit delivery trucks per unit area time, P_c is the number of parking spaces allocated to delivery trucks per unit area, and H is the stock of double-parking delivery trucks per unit area time. Passenger cars tend to cruise for spaces when there is no on-street parking available, whereas delivery trucks have to stay in proximity to the delivery site due to the need to load and unload cargo, and end up double-parking when there are insufficient spaces.

Additional parameters include θ for the ratio of delivery van space over passenger car space, m_p (m_c) is the distance traveled by passenger car (truck) in downtown before arriving to the destination, l_p (l_c) is the parking duration of a passenger car (truck), ρ_p (ρ_c) is the value of time of the passenger car commuter (truck driver), v is the travel speed, v_0 is a free flow speed, t is travel time per unit distance, t_0 is free flow travel time, Ω is the jam density in the absence of curbside parking, P_{\max} is the maximum number of parking spaces that could be accommodated by the street per unit area, α is a conversion rate from cruising cars to in-transit passenger cars, β is a conversion rate from delivery truck stock to an equivalent in-transit passenger cars, γ is a conversion rate from double-parked trucks to in-transit passenger cars, f is an on-street parking fee per unit time, q is a double-parking fine per unit time, and e is the elasticity of demand with respect to trip price.

The steady-state equilibrium is different than the user equilibrium seen in the prior sections of this chapter. There is no behavioral principle; rather it is simply a stationary point in which change in flows is set equal to zero. Under this condition, a system of six equations needs to be solved as shown in Eq. (3.29).

$$D_p = \frac{T_p}{m_p t} \quad (3.29a)$$

$$\frac{T_p}{m_p t} = \frac{P_p}{l_p} \quad (3.29b)$$

$$D_c = \frac{T_c}{m_c t} \quad (3.29c)$$

$$\frac{T_c}{m_c t} = \frac{P_c}{l_c} + \frac{H}{l_c} \quad (3.29d)$$

$$D_p = D_0 \left(\rho_p m_p t + \rho_p C \left(\frac{l_p}{P_p} \right) + f l_p \right)^e \quad (3.29e)$$

$$t = \frac{t_0}{1 - \frac{T_p + \alpha C + \beta T_c + \gamma H}{\Omega \left(1 - \frac{(P_p + \theta P_c)}{P_{\max}} \right)}} \quad (3.29f)$$

Eqs. (3.29a), (3.29c) are state transition conservation conditions. Eqs. (3.29b), (3.29d) describe the steady-state equilibrium. $\frac{P_p}{l_p}$ describes the entry and exit rates from the parking pool. Eq. (3.29e) is the demand function for passenger cars. Truck demand is assumed to be inelastic. Eq. (3.29f) is the travel time function, which is based on Greenshield's traffic flow model. Uniqueness is guaranteed under three conditions: cruising contributes more to congestion than in-transit vehicles ($t_c > t_T$), throughput capacity is higher than steady-state throughput, and the level of demand T_p is intermediate.

A social optimum is obtained by solving a nonlinear optimization model with Eq. (3.29) as the constraints and objective in Eq. (3.30) to maximize the social surplus by adjusting parking fee f . When the total number of parking spaces P is allowed to change, the problem is a first-best allocation. When only a parking fee can be set, that is a second-best allocation.

$$\max \int_0^{P_p/l_p} D_p^{-1}[x] dx - \left(\rho_p T_p + \rho_p C + f P_p + \rho_p P_p + \rho_c T_c + f P_c + q H + \rho_c (P_c + H) \right) \quad (3.30)$$

[Amer and Chow \(2017\)](#) showed that the optimization model is concave if the inverse demand function is also concave. The equilibrium and social optimum model are illustrated with a case study in downtown Toronto illustrated in [Fig. 3.19](#).

After gathering data on the relevant parameters and fitting the model to the data, the following results are computed for the equilibrium under a current \$4/h parking fee policy with no truck parking allocation, a second-best social optimum in which truck allocations are allowed, and a first best social optimum in which total parking spaces can be increased as well. This is shown in [Table 3.5](#) from [Amer and Chow \(2017\)](#).

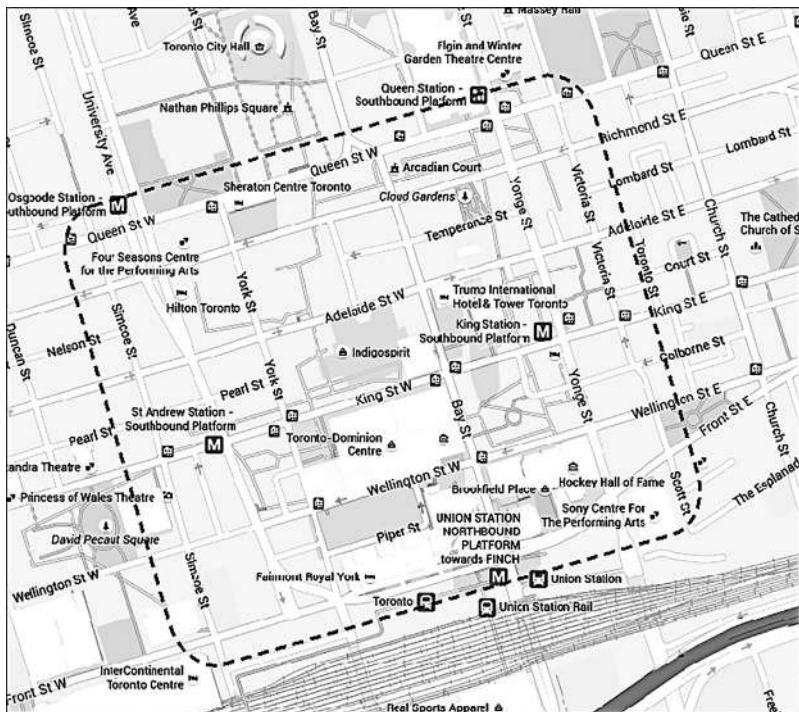


Fig. 3.19 Study area in downtown Toronto. (Source: Amer and Chow, 2017.)

In the base case, there is passenger trip demand $D_p = 2575 \text{ veh/h/mile}^2$ in the morning peak, truck trips $D_c = 865 \text{ veh/h/mile}^2$, average double-parking fine of $q = \$150/\text{h}$, in addition to the \$4 parking fee. In this equilibrium, there is cruising of $C^e = 429.5 \text{ veh/mile}^2$ and double-parking trucks of $H^e = 129.75 \text{ veh/mile}^2$. The average trip travel time per unit distance is 0.092 h/mi . There are $P = 5150$ parking spaces.

Under the second-best social optimum solution, some of the parking spaces are allocated for 130 truck parking spots, and the parking fee is increased to $f^o = \$7.85/\text{h}$. The result is that cruising goes down to $C^o = 0 \text{ veh/mile}^2$ and double-parking to $H^o = 0 \text{ veh/mile}^2$. Average trip travel time per unit distance decreases by 23% to $t^o = 0.071$, resulting in an increase in social surplus of \$14,304 per h-mi^2 compared to the equilibrium.

Under the first-best social optimum, the number of parking spaces increases up to $P^* = 6077$ spaces, parking fee goes back down to \$2.47/h, passenger car trip demand goes down 6% from 2575 in equilibrium to 2427 trips. Cruising and double-parking are still zero, while travel time

Table 3.5 Comparison of equilibrium and social optimal scenarios for downtown Toronto

	Equilibrium	Social optimum Parking fixed	Social optimum Parking variable
Inputs			
m_p (mile)	2		
l_p (h)	2		
ρ_p (\$/h)	20		
t_0 (h/mile)	0.067		
D_0 (constant)	4426		
P_{\max} (space/mile ²)	15,452		
Ω (veh/mile ²)	8252		
e (dimensionless)	-0.2		
α (dimensionless)	1.5		
β (dimensionless)	1.8		
γ (dimensionless)	4.4		
q (\$/h)	150		
m_c (mile)	0.181		
l_c (h)	0.15		
ρ_c (\$/h)	110		
D_c (veh/h/mile ²)	865		
θ (dimensionless)		1.64	
$P = P_p + \theta P_c$ (space/mile ²)	5150	5150	—
P_p (space/mile ²)	5150	—	—
P_c (space/mile ²)	0	—	—
f (\$/h)	4	—	—
Solution			
P_p^* (space/mile ²)	—	4937	5864
P_c^* (space/mile ²)	—	130	130
f^* (\$/h)	—	7.85	2.47
D_p^* (veh/h/mile ²)	2575	2469	2427
T_p (veh/mile ²)	473.6	353.1	424.2
C (veh/mile ²)	429.5	0	0
T_c (veh/mile ²)	14.4	11.1	11.3
H (veh/mile ²)	129.75	0	0
t (h/mile)	0.0920	0.07091	0.07234
v (mile/h)	10.8	14.1	13.8
ΔSS (\$/h-mile ²)		+\$14,304	+\$24,883

per unit distance goes back up to 0.072, which is still 21% decrease from equilibrium. The total increase in social surplus relative to the equilibrium is \$24,883 per h-mi², which is 74% higher than the second-best solution.

3.4.3 Taxi-Customer Matching

A last variant model of congestion is in the taxi market. What separates taxi operations from other types of congestion models is that the congestion effect is not just from the user side. It is also dependent on the service providers. The more taxis there are operating in a zone relative to the user demand, the quicker it would be to match a taxi to a user via some matching mechanism (e.g., hailing on the street, waiting at a taxi stand, calling a central dispatch, or e-hailing via mobile app). The more unbalanced the supply and demand are, the costlier it is to match them together. This cost is translated into waiting or planning time (see [de Borger and Fosgerau, 2012](#), for the substitutability between the two).

[Cairns and Liston-Heyes \(1996\)](#) provided an early study of the costs of searching and matching in the taxi industry. This concept was expanded to road networks to capture vacant taxis and fleet size ([Yang and Wong, 1998](#)). [Lagos \(2000\)](#) proposed a search friction model as a steady-state equilibrium. [Wong et al. \(2005\)](#) proposed a bilateral search behavior for taxis based on absorbing Markov chains in a double-ended queueing system.

Due to the entry of e-hailing companies and advances in mobile technologies, interest in bilateral search friction between taxis and passengers has increased in recent years. [Yang and Yang \(2011\)](#) proposed Cobb-Douglas-based search friction cost functions to determine market equilibrium. Several studies based on this bilateral search function have since been proposed. [He and Shen \(2015\)](#) and [Zha et al. \(2016\)](#) studied the equilibrium of ridesourcing and e-hailing taxi markets. [Zha et al. \(2017\)](#) studied temporal equilibrium of ridesourcing markets to evaluate surge pricing policies.

For taxi markets that cover heterogeneous demand patterns over a region, a spatial-temporal analysis framework may be more suitable. [Buchholz \(2015\)](#) proposed a spatial-temporal steady-state equilibrium model based on the search friction model from [Lagos \(2000\)](#), in which a region is divided into multiple zones and time intervals corresponding to the maximum wait time that passengers are willing to endure before turning to another alternative. The basis for the search friction is a dynamic system shown in [Fig. 3.20](#), where a stationary state is sought.

The number of matches $m[v_{it}, u_{it}]$ in a time interval t and zone i as a function of available taxis v_{it} and passenger demand u_{it} is defined in Eq. (3.31).

$$m[v_{it}, u_{it}] = v_{it} \left(1 - \left(1 - \frac{1}{\alpha v_{it}} \right)^{u_{it}} \right) \quad (3.31)$$

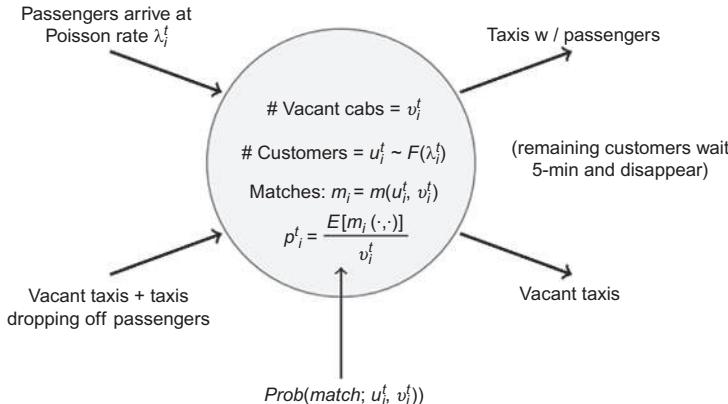


Fig. 3.20 Dynamics of search friction. (Source: Buchholz, 2015.)

Assuming a fixed probability for a passenger wanting to go from zone i to zone j at time t , M_{ijt} , the value function $V_{it}(S)$ for a particular state S is defined as Eq. (3.32).

$$V_{it}(S) = E_{p_i | \lambda_{it} S_t} \left[p_i(u_{it}, v_{it}) \left(\sum_j M_{ijt} (\Pi_{ij} + V_{j,t+\tau_{ij}}) \right) + (1 - p_i(u_{it}, v_{it})) \cdot E_{\epsilon_{a,j,t+1}} \left[\max_{j \in A(i)} \{ V_{j,t+\tau_{ij}} + \Pi_{[j=i]} \gamma - c_{ij} + \epsilon_{a,j} \} | \lambda_{it} \right] \right] \quad (3.32)$$

where

Π_{ij} is the revenue from serving the OD pair (i,j) defined by Eq. (3.33) with fixed fare b , unit distance fare rate π , distance δ_{ij} , and fuel cost c_{ij} ;

$$\Pi_{ij} = b + \pi \delta_{ij} - c_{ij} \quad (3.33)$$

λ_{it} is a latent passenger arrival rate that needs to be estimated;

γ is the extra payoff associated with staying put;

ϵ_{aj} is a Gumbel-distributed idiosyncratic shock to a taxi driver's perceived value of search in each alternative location j ;

$\sum_j M_{ijt} \cdot (\Pi_{ij} + V_{j,t+\tau_{ij}})$ is the expected value of the fare;

$E_{\epsilon_{a,j,t+1}} [\max_{j \in A(i)} \{ V_{j,t+\tau_{ij}} + \Pi_{[j=i]} \gamma - c_{ij} + \epsilon_{a,j} \} | \lambda_{it}]$ is the expected value of vacancy.

Idle taxis are modeled to choose a new destination zone to cruise to maximize expected destination value $W(j_a, S_t)$, as shown in Eq. (3.34).

$$P_i(j_a | S_t) = \frac{\exp\left(\frac{W(j_a, S_t)}{\sigma_\epsilon}\right)}{\sum_{k \in A(i)} \exp\left(\frac{W(j_k, S_t)}{\sigma_\epsilon}\right)} \quad (3.34a)$$

$$W_{it}(j_a, S_t) = E_{S_t + \tau_{ij_a}} [V_{j_a, t + \tau_{ij_a}}(S_t + \tau_{ij_a}, \cdot, \cdot) - c_{ij_a}] \quad (3.34b)$$

Buchholz (2015) proved the existence and uniqueness of an equilibrium of the model. Estimation of the parameters is based on finding a fixed point through an iteration of the equilibrium algorithm and inverting the matching function to infer the demand update as shown in Fig. 3.21.

A spatial-temporal taxi equilibrium model such as this one can be used to evaluate fare hikes, special events that draw large demand surges, and evaluate changes in service coverage by time of day and by region.

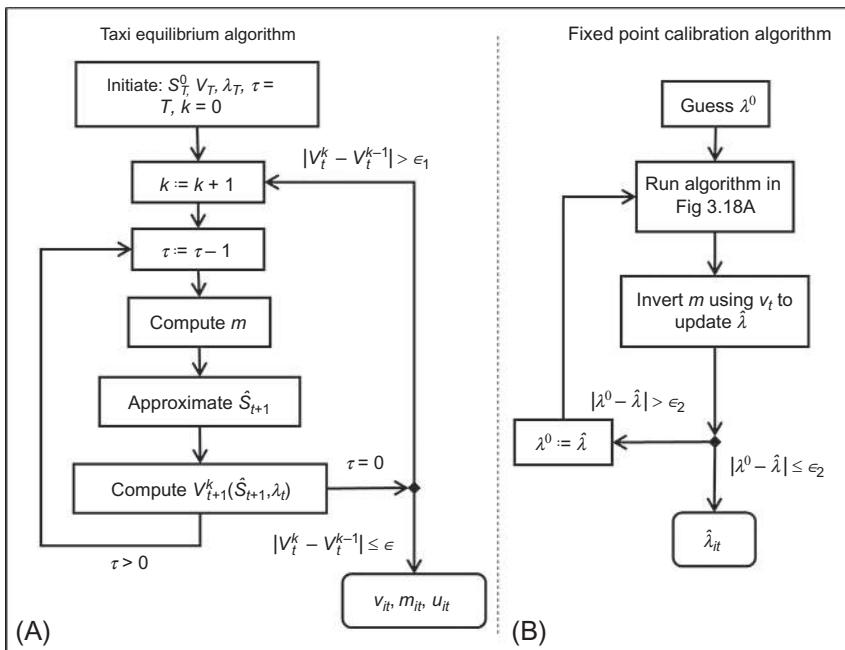


Fig. 3.21 Algorithms to (A) run and (B) calibrate the taxi equilibrium model.

3.5 TRANSPORT SYSTEMS AS TWO-SIDED MARKETS

3.5.1 Motivation

In a smart cities setting, the rise of the MaaS paradigm makes transport systems analysis much more complex. Performance is not just based on only user decisions. Operators' routes are dependent on user preferences, but requests may not be known until the moment they are made, leading to the need for real-time operating policies and control strategies. Worse yet, the methods discussed in [Sections 3.1–3.4](#) so far apply to this problem in principle but are not practically applicable.

For example, the taxi equilibrium model at least can model the joint decisions of users and operators to reach steady-state equilibrium. However, what happens when we allow for shared rides? Even allowing for only paired rides, it is possible to implement a host of different strategies to match passengers: they can be paired together on-site by having first passenger wait for a second to arrive, or they might be matched by a central dispatch such that the first passenger gets detoured to serve the second passenger. Constraints may be applied for wait time, detours, similarities in origin and destination, fare pricing allocations (see [Furuhashi et al., 2015](#)), and so on.

Consider the case of allowing taxi sharing for passengers going to the JFK International Airport in New York. Data from the Port Authority of NY and NJ is available for single ride taxis to the airport, allowing [Ma et al. \(2017b\)](#) to estimate a mode choice model. The model is applied to evaluate shared taxi service. Simply by substituting the matching mechanism from one extreme (only waiting) to the other (only detour), the model predicts vastly different spatial welfare effects, as illustrated in [Fig. 3.22](#). This suggests the equilibration aspect of the MFG framework when applied to MaaS is highly dependent on operators' policies.

New equilibrium models are needed to evaluate these systems at their operating policy level. Equilibrium models evaluated from a steady-state setting may not be able to capture the stochastic dynamics that feature so prominently in many MaaS configurations. Examples are shown in [Table 3.6](#), where policies are separated between “within-day” and “day-to-day” sensitivities.

Furthermore, the interplay between private operators, public agencies that maintain the built environment, and the users of the system in the smart cities setting are not explicitly characterized in any of the prior models in this chapter.

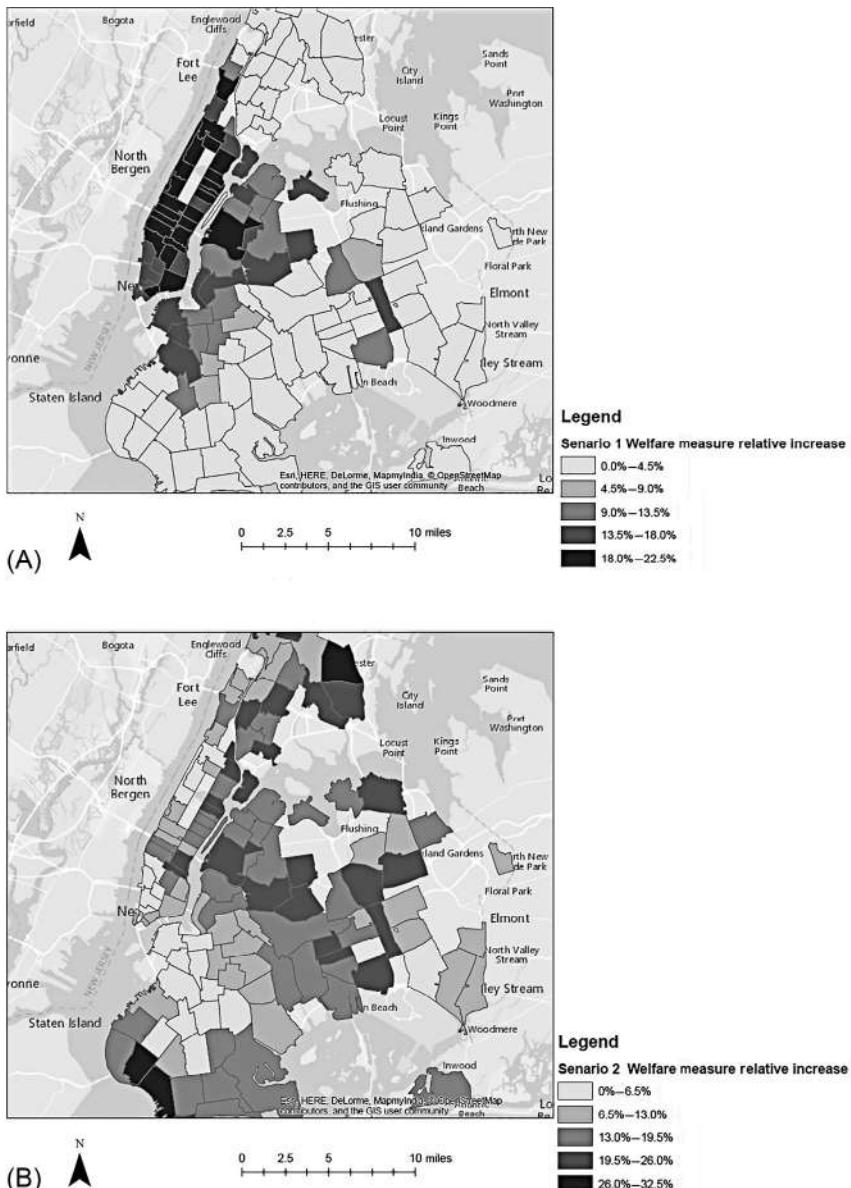


Fig. 3.22 Relative spatial changes in consumer surplus from single-ride taxi market to shared taxi via (A) matching by waiting and (B) matching by detour. (Source: Ma et al., 2017b.)

Table 3.6 Different types of mobility as a service

Service type	Typical day-to-day policies	Notes
Fixed route public transit	N/A	Fares, service coverage are decisions made centrally, so no day-to-day dynamic
Private fixed route transit operators	Fares, service coverage	Competition between operators will lead to dynamic changes
Price-regulated taxis	Service coverage	Matching and routing are within day; service coverage is a decentralized decision
Ridesourcing	Fleet size, fares	Matching and routing are within day, fares and fleet size can also be within day
Peer-to-peer ridesharing	Pricing, matching	As a decentralized system, most decisions will be day-to-day dependent
Microtransit	Service coverage, fares	Matching, routing are within-day policies
Car/bike sharing	Service coverage, fares	Vehicle balancing is a within-day policy

3.5.2 Two-Sided Market Framework

To capture the interplay between private operators, public travelers, and a public agency acting as the facilitator, consider a two-sided market framework. A two-sided market is one in which one or several platforms enable interactions between end users and try to get the two (or multiple) sides ‘on board’ by appropriately charging each side.

Definition 3.1 (Rochet and Tirole, 2006). *Consider a platform charging per interaction charges p^B and p^S to the buyer and seller sides. The market for interactions between the two sides is one sided if the volume of transactions realized on the platform depends only on the aggregate price level $p = p^B + p^S$. If by contrast the volume of transaction varies with p^B while p is kept constant, the market is said to be **two sided**.*

Examples of two-sided markets include Uber, Airbnb, videogame platforms like Sony PlayStation that support games from third-party developers, and more. In that case, one can argue that the IoT infrastructure as shown in Fig. 2.4 is representative of a two-sided market. More specifically, looking at the roles of public transportation agencies, we can describe the interactions and control strategies as a two-sided market as shown in Fig. 3.23.

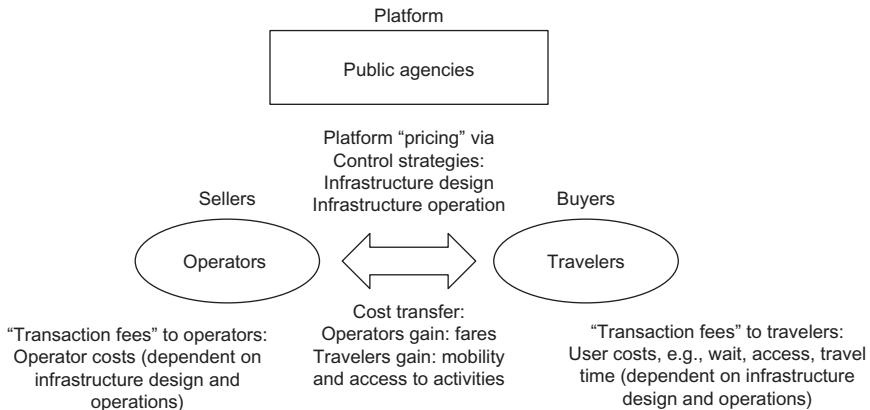


Fig. 3.23 Smart cities mobility provision represented as a two-sided market.

The public agency is the platform. It “sets transaction prices” by determining infrastructure designs (road alignments, parking, rail alignments (if publicly owned), stations and ports, etc.) and operations (tolls, operating hours, maintenance, etc.). Based on these “prices,” operators and travelers join the platform as sellers and buyers, respectively, to exchange costs: travelers pay the operators fares in exchange for mobility and access to activities. The transaction fees set by the public agency are translated into costs: route-based operating costs for the operators based on how they respond to the infrastructure designs and wait/access/travel times as a result of the infrastructure design/operations.

Viewed in this light, we can align the design and operation of infrastructure in a smart city toward optimizing the interactions of the seller and buyer markets. For example, for a given “price” decision of the platform, it results in a certain amount of sellers and buyers joining the platform as illustrated in Fig. 3.24.

When an operator or set of operators provide mobility services according to a specific operating policy, and users respond to the service according to a specific user behavior, a public agency needs to measure how well their control strategies (infrastructure design/operations) impact the social welfare of the system. As a two-sided market, the social welfare can be measured in Eq. (3.35).

$$H^B = N^S h^B[p^B] = D^S[p^S] h^B[p^B] \quad (3.35a)$$

$$H^S = N^B h^S[p^S] = D^B[p^B] h^S[p^S] \quad (3.35b)$$

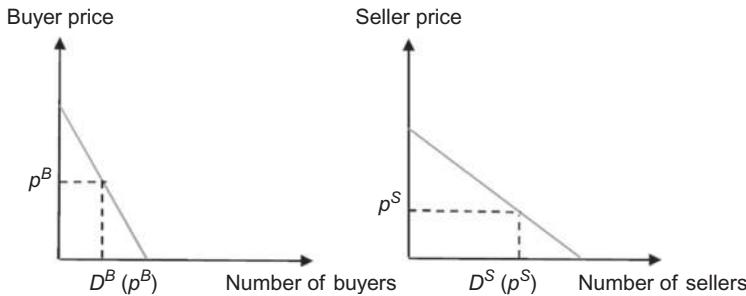


Fig. 3.24 Illustration of the sensitivity of two-sided market to platform pricing.
(Source: Djavadian and Chow, 2017a.)

where $H^B (H^S)$ is the total surplus for the travelers (operators), $N^B (N^S)$ is the number of travelers (operators) participating in the market, p^B is the generalized user cost of travel, p^S is the generalized operating cost, $h^B (h^S)$ is the net surplus for each traveler (operator), and $D^B (D^S)$ is a log-concave demand function that relates number of travelers (operators) to the cost of travel (operation). The social welfare Φ is defined in Eq. (3.36), where c is some constant.

$$\Phi = D^B[p^B] \int_{p^S}^{\infty} D^S[w] dw + D^S[p^S] \int_{p^B}^{\infty} D^B[w] dw, p^B + p^S = c \quad (3.36)$$

The social optimum can be obtained based on Ramsey pricing criterion (Rochet and Tirole, 2003) by equating the first-order condition with respect to the elasticities of demand $\eta^B(\eta^S)$ for the travelers (operators) as shown in Eq. (3.37).

$$\frac{p^B}{\eta^B D^B} \left[\int_{p^B}^{\infty} D^B[w] dw \right] = \frac{p^S}{\eta^S D^S} \left[\int_{p^S}^{\infty} D^S[w] dw \right], p^B + p^S = c \quad (3.37)$$

As discussed in Djavadian and Chow (2017a), the gap between the two terms in Eq. (3.37) is used to measure suboptimality. This can also be used to guide agency decision support; if the gap is negative (positive), it means that traveler (operator) cost may need to be reduced relative to operator (traveler) cost by means of policies to transfer cost from user to operator since elasticities tend to be negative. Examples of cost transfer from user to operator are to reposition stations/stops in a way that it reduces access cost at the expense of the route operations, to allow for route deviations to drop passengers closer to home, increase fleet size, or to simply reduce the fare or increase

service coverage region. Such a policy would benefit a negative gap (but would worsen a positive gap). An illustration of a policy analysis using the 2-sided market measures of social optimum is provided in [Exercise 3.12](#).

$$G = \frac{p^B}{\eta^B D^B} \left[\int_{p^B}^{\infty} D^B[w] dw \right] - \frac{p^S}{\eta^S D^S} \left[\int_{p^S}^{\infty} D^S[w] dw \right] \quad (3.38)$$

Exercise 3.12

A city agency has designed and operated the city infrastructure in such a way (density, road space usage, tolls, land use, etc.) that average travel cost per trip is \$10 and average operating cost per trip is \$14. Based on the underlying operating policy of an MaaS operator, its demand function is $D^S = 120 - 2p^S$. At the same time, the underlying travel behavior leads to a demand function of $D^B = 800 - 16p^B$. Determine the social welfare suboptimality of this scenario and a strategy to explore to improve the optimality.

1. First we use Ramsey pricing to determine the optimal prices for social optimality. Knowing the current prices set, the value of total price is set to $c = \$10 + \$14 = \$24$ per trip.
2. We also need to determine the elasticities of demand:

$$\eta^B = -\frac{p^B}{D^B} \left(\frac{\partial D^B}{\partial p^B} \right) = -\frac{16p^B}{D^B}, \eta^S = -\frac{2p^S}{D^S}$$

Since max price allowed under the demand function is $p_{\max}^B = \frac{800}{16} = \50 and $p_{\max}^S = \frac{120}{2} = \60 , the integrals are:

$$\begin{aligned} \int_{p^B}^{50} D^B[w] dw &= 800(50) - 8(50)^2 - 800p^B + 8[p^B]^2 \\ &= 20,000 - 800p^B + 8[p^B]^2 \end{aligned}$$

$$\int_{p^S}^{60} D^S[w] dw = 3600 - 120p^S + [p^S]^2$$

3. Then we have two equations and two unknowns:

$$\begin{aligned} -\frac{1}{16}(20,000 - 800p^B + 8[p^B]^2) &= -\frac{1}{2}(3600 - 120p^S + [p^S]^2) \\ p^B + p^S &= 24 \end{aligned}$$

Solving for them we get:

$$[p^B]^* = \$7, [p^S]^* = \$17, [D^B]^* = 688, [D^S]^* = 86, \Phi^* = \$2,544,224$$

4. Now we compare our current equilibrium. At $p^B = \$10$, $p^S = \$14$, the demand is $D^B = 640$, $D^S = 92$, and the social welfare is $\Phi = \$2,531,840$. This is suboptimal by 0.5%, so it's not too bad.

Note if, on the other hand, $p^B = \$20$, $p^S = \$4$, this could have been suboptimal by 9.1%. Another way to compare different states relative to the social optimum is to compare the left- and right-hand sides of the equality in Eq. (3.37). In the social optimum, they are equal. In the case

where $p^B = \$10$, $p^S = \$14$, $\frac{p^B}{\eta^B D^B} \left[\int_{p^B}^{\infty} D^B[w] dw \right] = -800$ and

$$\frac{p^S}{\eta^S D^S} \left[\int_{p^S}^{\infty} D^S[w] dw \right] = -1058. \text{ The gap between them is } -258, \text{ which}$$

implies strategies that reduce traveler cost relative to operator cost would help reach social optimum.

3.5.3 Day-to-Day Adjustment Processes of Two-Sided Markets

Although there is a measure to evaluate the suboptimality of a two-sided platform, for MaaS it is also difficult to predict the equilibrium of the system. For this purpose, the system can be represented by a day-to-day adjustment process instead of a steady-state equilibrium. Beckmann et al. (1956) have discussed the dynamics of these systems and noted that an equilibrium will not necessarily be reached from an arbitrary initial state.

Smith (1979) conducted one of the first rigorous studies to examine the stability of traffic equilibria under deterministic route assignment and found that a deterministic user equilibrium may not necessarily be stable (user optimal, per Smith, 1984a) from a dynamic adjustment perspective. Horowitz (1984) investigated the dynamic properties of networks operating under stochastic assignment behavior and found that even when there is a unique equilibrium, day-to-day dynamic link volumes may either converge to that state, oscillate about it perpetually, or converge to a different state depending on the route choice decision-making process. This additional dependency on route choice behavior and day-to-day learning has also been experimentally verified (Mahmassani and Chang, 1986; Mahmassani, 1990). These findings suggest that while a network equilibrium can be represented in a dynamic systems approach, learning and behavioral characteristics need to be taken into account carefully.

The work of Smith (1984c) showed using Lyapunov functions that dynamic adjustment processes converge to a nonempty set of equilibria as long as the cost-flow function is monotone and smooth. Friesz et al. (1994) described the adjustment process under information provision using an economic tatonnement concept. Cantarella and Cascetta (1995) provided a unified theory of dynamic equilibria in transportation networks in which deterministic processes always have at least one fixed point. Friesz et al. (1996) established mathematical axioms that distinguished between fast and slow dynamic processes. Zhang et al. (2001) proved that a stationary link flow pattern is a necessary and sufficient condition for user equilibrium path flow. Yang and Zhang (2009) summarized five types of deterministic day-to-day adjustment processes and showed that they all belong under a general class of rational behavior adjustment processes.

In addition to deterministic processes, there are stochastic day-to-day adjustment processes. Deterministic processes are known to exhibit separable basins of attraction. Stochastic processes can provide ergodic probability distributions even for examples with nonunique deterministic equilibria. However, the stable set may not be separable. Smith et al. (2014) considered new processes that combined features of both.

Djavadian and Chow (2017b) proposed an adjustment process that also combined features from both types of processes. It uses simulation to generate different populations, and for each population it attains a stable state via deterministic adjustment. The method was applied to evaluate two-sided markets by Djavadian and Chow (2017a) where both travelers and operators adjust their within-day decisions for different simulated populations to obtain a probability distribution over a stable set. The model is illustrated in Fig. 3.25.

RESEARCH AND DESIGN CHALLENGES

The following are open-ended challenges that sometimes refer to real data sets. Readers are encouraged to try to tackle the problems and post their designs or solutions on the following GitHub page, <https://github.com/BUILTNYU/Elsevierbook/tree/master/Chapter%203>.

- (3.1) Go to a local busy coffee shop and collect data over a half hour of people getting on the queue and getting served. Keep track of the number of servers (s) and estimate the arrival (λ) and service rates (μ) for an M/M/ s model per Appendix C. Using the average queue delay W_q as the average cost function, estimate the MECC.

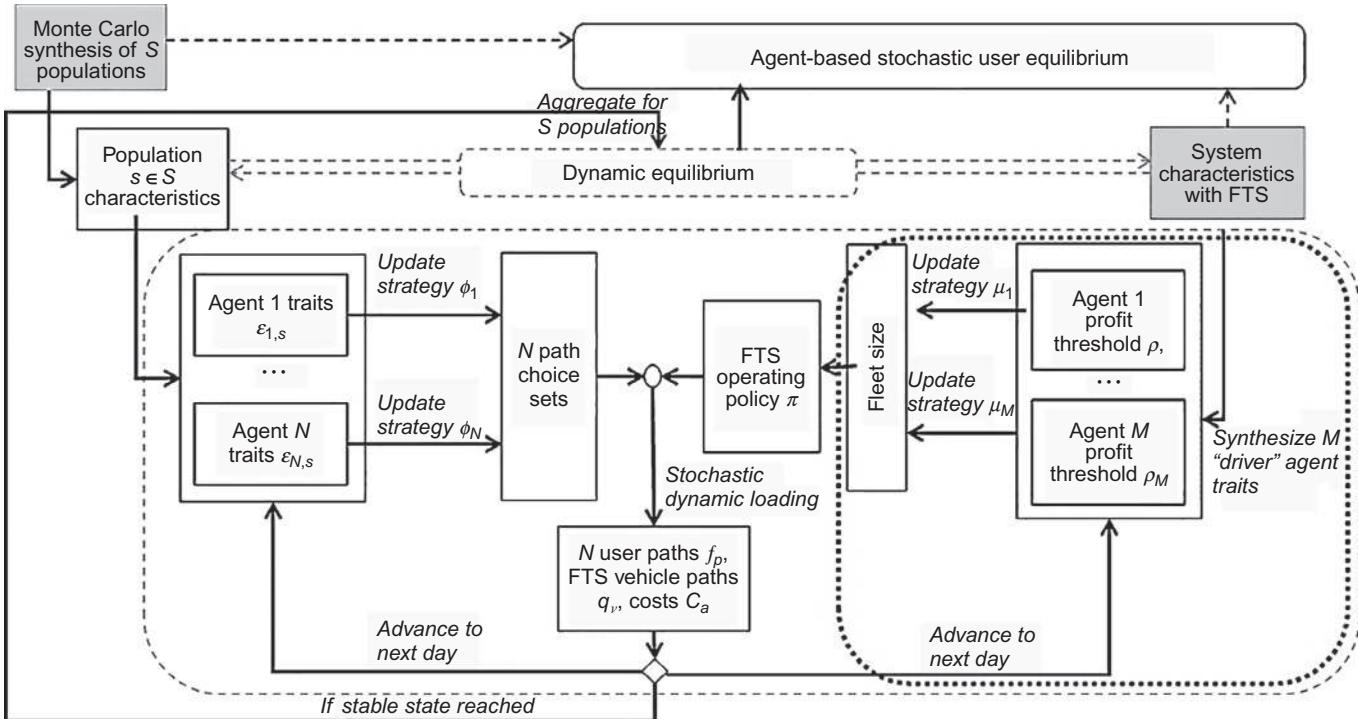


Fig. 3.25 Components of agent-based stochastic user equilibrium obtained by day-to-day adjustment of user and operator decisions as a two-sided market. (Source: Djavadian and Chow, 2017a.)

- (3.2) Go to <http://pems.dot.ca.gov/> and create a free account. Click on one of the Vehicle Detector Stations (VDS) at the Bay Bridge in San Francisco. Sample observations of the flow every 5 min for 2 h. Also sample the corresponding real time travel time of crossing the Bay Bridge on Google Maps at the same times. Use the sampled data to calibrate the parameters of the BPR function for the Bay Bridge:
- $$t = t_0 \left(1 + \alpha \left(\frac{x}{C} \right)^\beta \right)$$
- where x is the observed flow, t is the travel time, and the parameters are t_0 , C , α , and β . If the Bay Bridge toll price is assumed optimal (<http://baybridgeinfo.org/tolling-information>), what would be an appropriate linear demand function?
- (3.3) Implement the Frank-Wolfe algorithm in a language of choice and use it to find a UE and SO assignment on the Sioux Falls network. Compare the results and computation time to results on <https://github.com/bstabler/TransportationNetworks>. Modify the algorithm to solve for an elastic demand objective, where the OD demand is linear with double the original demand value when travel time is zero and zero when travel time is 90 min.
- (3.4) Go to <https://www.openstreetmap.org> and export a shapefile for a road network in one city of your choice (you may need to make a “capacity” field for the link performance function). Create two zones (A, B) located at different ends of the network as centroids.
- a. Use <http://aequilibrae.com/> to link the two centroids to the network and then gradually load demand going from A to B and from B to A: 1000, 2000, 3000, and so on and observe how the flows get loaded under UE assignment.
 - b. Consider two alternative links for 50% capacity parameter increase. Compare total system travel times of the network for each link when demands are 5000 versus when they are 10,000. How much does the comparison depend on the demand?
- (3.5) Create a simplified highway network model of NYC for the primary purpose of modeling circulation patterns for vehicles entering and exiting the city, or crossing different boroughs. Calibrate a multiclass UE model with elastic demand to include passenger vehicles and trucks. Calibrate the model to include the current toll charges for passenger vehicles and trucks for a weekday a.m. peak. Test out different policies for congestion pricing (like the \$12 NYC

congestion charge: <http://www.nydailynews.com/new-york/group-final-push-12-nyc-congestion-charge-article-1.3888806>) as well as for tolling trucks entering and exiting the city.

- a. How effective is the \$12 congestion charge? Is there a better design?
 - b. How effective is the current truck tolling policies for entering trucks? What happens if all the gateways into NYC included truck tolls?
- (3.6) Write a program that searches each link in a network to identify one that can demonstrate Braess' Paradox (removing it would improve total system travel time) for a given set of OD demand. Test it on the Sioux Falls network.
- (3.7) For the network in Fig. 2 of [Chin et al. \(2016\)](#), solve the stochastic assignment with Dial's algorithm using $\theta=10$ and $\theta=0.1$. Interpret the consumer surplus of the solutions. Now implement Spiess and Florian's algorithm and apply it to the same network. Compare results; is there a θ where the results can be equivalent? Or is the presence of unattractive lines a problem?
- (3.8) Implement MSA with Dial's algorithm for SUE and apply it to the Sioux Falls network. Compare the objective portion $\frac{1}{\theta} \sum_{(r,s) \in W} \sum_{k \in K} f_{rsk}(\ln f_{rsk})$ in Eq. (3.12) between the SUE solution with the first iteration solution.
- (3.9) For the Sioux Falls network, consider altering the link performance function of each link to be based on upstream link flows instead of the same link flows. Try to solve one iteration of Frank-Wolfe algorithm (Challenge 3.3) and discuss what happens.
- (3.10) For the freight assignment model in Eq. (3.15), formulate an equivalent VI.
- (3.11) For the freight assignment model in Eq. (3.15), either an SO or a UE objective is assumed. But in the case of freight, the decision-makers are not single individuals but are large companies with fleets of vehicles that form alliances with other companies even as they compete. If a weighted sum between SO and UE objectives were used, can the solution be a better fit of observed flows? Test this with commodity flow survey data in the Freight Analysis Framework (https://ops.flhwa.dot.gov/freight/freight_analysis/faf/#faf4) for a region.
- (3.12) Create a program for [Algorithm 3.6](#) from [Wu et al. \(1994\)](#). Apply it to solve the transit network example from Fig. 4 in [De Cea and Fernández \(1993\)](#).

- (3.13) Apply the program for [Algorithm 3.6](#) created in Challenge 3.12 to evaluate a simplified subway network of NYC Transit with the proposed Regional Unified Network from ReThinkNYC (<http://www.rethinknyc.org/the-plan/>).
- (3.14) Congestion pricing for access into Manhattan can be modeled using the Vickrey morning commute model. Using the 2010/2011 NYMTC Household Travel Survey (<https://www.nymtc.org/DATA-AND-MODELING/Travel-Surveys/2010-11-Travel-Survey>), obtain the demand for commuters from outside Manhattan into Manhattan for work, including volume and distribution. Calibrate a bottleneck capacity that would best fit the observed distribution. Then propose a pricing scheme that would optimize the social surplus. Compare the distribution of departure and arrival times between pricing with no pricing.
- (3.15) Collect data from OpenData NYC (<https://opendata.cityofnewyork.us/data/>) to calibrate a downtown parking model for Manhattan with truck double-parking. Determine space allocation and pricing strategies to maximize social surplus.
- (3.16) Simplify Buchholz's model in [Fig. 3.21](#) for the special case of a two-zone system. Consider parameters of the two zones such that one tends to have more density of pickups. What is the preference to stay in the same dense zone compared to serving interzonal trips?
- (3.17) For Challenge 3.5 (NYC simple model), evaluate the scenarios using the two-sided market framework in [Section 3.5.2](#). Which one has the smallest gap toward optimality? For the congestion pricing scenario, based on the sign of the gap, suggest an alternative policy and evaluate it.
- (3.18) In transit fleet operations, there are usually user and operator costs with weights assigned to them. An operator who reduces the user cost weight may lose user demand over time, but placing a high user cost weight might impose high operating costs. For the day-to-day adjustment process for two-sided transport markets, consider modifying the process to allow for learning of weights between user and operator cost. Demonstrate how this would work for a single service van with passenger capacity of six serving random customers in a Euclidean space to a terminal station day to day. For this van, assume it follows the nearest neighbor policy with a maximum detour multiplier θ which is initially set to 1.5 and updated each day.

CHAPTER 4

Market Schedule Equilibrium for Multimodal Systems

4.1 THE NEED TO EVALUATE ACTIVITY SCHEDULING BEHAVIOR

In [Chapter 3](#), the focus of user equilibrium traffic assignment is on the congestion externality that users impose on others when the system's resources are finite. That is, one basis for evaluation of user response to urban transport systems designs. A second fundamental basis for user response is the heterogeneity of our activity scheduling throughout the day. The economic vibrancy of a community is only possible through the diversity of preferences of its users in conducting different activities in different locations at different times of the day. In addition, the methods in [Chapter 3](#) assume that the underlying transaction for transportation is a "trip" from one location to another. While such an assumption may suffice in evaluating the effects that users have on each other when sharing routes in the system, it does not explain user response sensitivity to schedule-based transport systems like MaaS, on-demand mobility, shared vehicles, and so on.

The importance of user scheduling response cannot be ignored when evaluating mobility systems in a smart city. We call the equilibrium that occurs when users adjust their schedules in response to a transport system design as the "market schedule equilibrium." It is possible to show that even when we ignore congestion effects the design of a transport system can lead to counterintuitive user responses in a market schedule equilibrium that can only be explained with a user activity scheduling model. These paradoxical effects are illustrated here from examples in [Kang et al. \(2013\)](#).

Consider a 4-node network on which a user lives and works, as shown in [Fig. 4.1](#). The user lives at node 0, works at node 3, and does grocery shopping at node 1. The observed arrival times to each activity under the current transport system design as shown in [Fig. 4.1B](#) are assumed.

In the base case, the user needs to arrive at work between 9 a.m. and 10 a.m., and to the social activity between 6:15 p.m. and 6:30 p.m. with unit costs of schedule delay beyond those windows \gg unit travel and wait costs. Due to

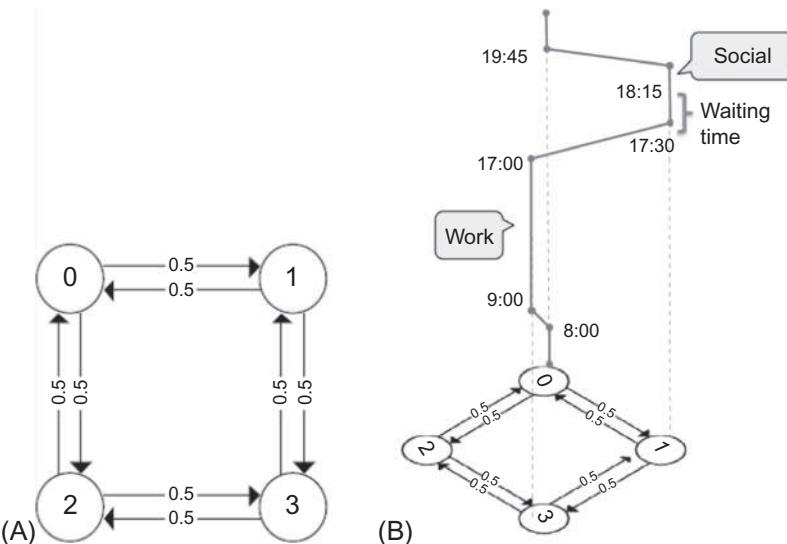


Fig. 4.1 A (A) 4-node network on which (B) a user conducts activities. (Source: Kang et al., 2013.)

the travel times of the network and the arrival time constraints for the user, they end up adding 45 min of wait time into their planned schedule. If every minute spent outside of home is treated equally as a cost regardless of whether it is travel or idle wait time (for illustration), then the total cost of the base schedule is $19.75 - 8 = 11.75$ h. Adding a new link from node 3 to node 0 with a travel time of 0.7 h would incentivize a traveler to make a trip back home after work before heading out to the social event as shown in Fig. 4.2, since the total incurred cost would now be lower: $(19.75 - 17.75) + (17.7 - 8) = 11.7$ h. In the new scenario, no new activities are conducted, and there are no congestion effects. Regardless, adding a new link to the network results in generating an *additional trip* to the user's schedule. This paradoxical phenomenon is noted as follows.

Definition 4.1 Kang-Chow-Recker (KCR) Trip Paradox. *A link investment to a network, even in the absence of congestion effects, can lead to a paradoxical increase in number of trips due to user scheduling preferences that do not generate any new economic activity.*

Now consider what happens if wait time is valued at 1.75 times that of travel time. In the base case, we can define the disutility as the weighted sum of travel time and wait time: $2 + 0.75(1.75) = 3.3125$ h. If the travel time on

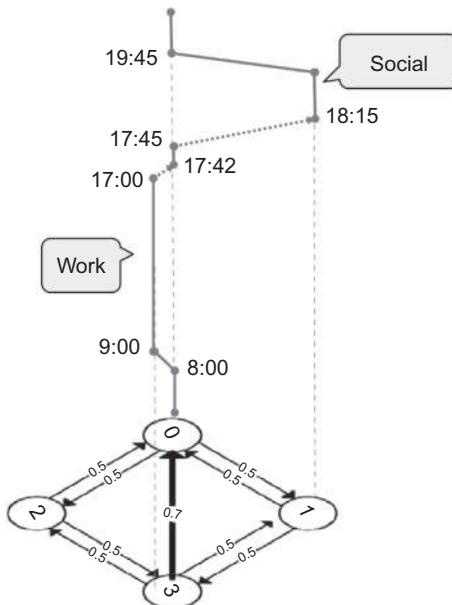


Fig. 4.2 Additional trip generated without any additional economic activity (Kang et al., 2013).

link $(3, 1)$ is improved from 0.5 to 0.25 h, then due to the activity arrival time constraints the user's schedule would not change from the base case. Instead, the 0.25 h difference would be converted from travel time to wait time, leading to a total disutility of $1.75 + 1(1.75) = 3.5$ h. Even though the network is improved, disutility for a user can worsen due to the combination of schedule parameters. This leads to a second paradox associated with activity scheduling.

Definition 4.2 KCR Utility Paradox. *A link investment to a network, even in the absence of congestion effects, can lead to a paradoxical increase in disutility to travelers when considering schedule constraints.*

These examples make two important points. First, the congestion effects and schedule effects can *independently* impact user responses in ways that cannot always be anticipated. Second, if schedule effects are not modeled it is possible not to know when network improvements can lead to additional vehicle miles traveled without contributing to economic output or reducing consumer surplus for some.

4.2 COMPLEXITY OF ACTIVITY SCHEDULING

Having established the importance of user scheduling, consider also the complexity of the problem. At its most basic level, users have incomes and time budgets from which to allocate to goods consumption to maximize utility, where the total time spent includes time at work to generate income to pay for utility-generating goods (Becker, 1965). An updated version of the model (see Small and Verhoef, 2007) is shown in Eq. (4.1).

$$\max_{G, T_w, \{T_k\}} \Lambda = U[G, T_w, \{T_k\}] + \lambda(Y + wT_w - G) + \mu \left(\bar{T} - T_w - \sum_k T_k \right) \\ + \sum_k \phi_k(T_k - \underline{T}_k) \quad (4.1)$$

where G is the amount of goods consumed, T_w is the time spent at work, T_k is the time spent at other activities k , Y is the unearned income, wT_w is the earned income with wage rate w , \bar{T} is the total time available, and \underline{T}_k is the minimum time needed to be spent on activity k . The objective Λ is a Lagrangian function in which the latter three terms are resource allocation constraints with corresponding Lagrange multipliers λ, μ, ϕ_k . The first constraint corresponding to λ requires costs spent on goods consumed not to exceed a total income that includes earned income from time spent working. The second constraint corresponding to μ requires that allocated time does not exceed total time available. The third constraint corresponding to the set of $\{\phi_k\}$ requires a minimum amount of time spent at an activity before utility can be attained. In many transport modeling applications, the time spent at work is held constant as a “compulsory activity” such that changes in transport system designs impact the allocation of time on activities around them.

The third constraint gets more complicated when the minimum time is not constant, but is rather dependent on accessibility of an activity among a choice set of activities distributed over space. This aspect of the problem is discussed by Hägerstrand (1970) and subsequent researchers in time geography. An activity k that is farther away from a user would essentially impose a higher value of T_k to access it. However, this cost can also depend on how a user sequences their activities. Activities sequenced in two different ways will impose different values of T_k as well. This consideration turns out to be very problematic because sequencing activities over space, known as a Traveling Salesman Problem (see Chapter 7), are computationally expensive to solve exactly (Held and Karp, 1962).

On top of this complexity, Section 3.4.1 discusses the costs associated with arriving at an activity late or early. These costs, as shown by Small

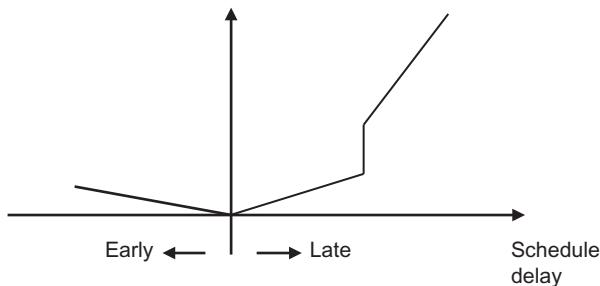


Fig. 4.3 Disutility of schedule delay depending on the flexibility of late arrival.

(1982), are not necessarily symmetric and do not extend linearly from a desired arrival time. An illustration of a nonlinear schedule delay cost function for work trips is shown in Fig. 4.3. Not only are the schedule delay costs asymmetric, but they also vary from one activity type to another and may also depend on the activity. For example, shopping may be highly flexible in schedule with little to no requirement on arrival time but have physical time windows for when the store is open. These would further impose additional constraints to the time allocation problem in Eq. (4.1), and the constraints would vary from person to person.

To evaluate the market equilibrium of an urban transport system design based on user activity scheduling, it is necessary to use a model of individual scheduling behavior that can take those designs into account. Such a model should allow the sequencing, departure/arrival times, selection of activities, and mode choice throughout the day to be sensitive to design variables of a transport system: routes covered, pricing, capacities, and service schedules for MaaS.

4.3 A MODEL OF USER ACTIVITY SCHEDULING BEHAVIOR

4.3.1 Literature Review

Activity scheduling is an especially difficult behavior to model. Many of the decisions, such as whether to trip chain, sequencing of certain trips, departure times, and mode choice, are highly interdependent. For example, people choosing to commute by park and ride may have to drive a car to a parking facility far from a downtown area and leave their car behind to take transit to work. At the end of the day, they would need to return to the parking facility to pick their car back up. These decisions are further impacted by

parking capacity, parking fees, train schedule, location of stops, and other infrastructure-related designs and operating policies.

Models of user activity scheduling in a spatial-temporal context emerged in the 1970s after the time-geographic theory proposed by Hägerstrand (1970). Jones (1979) specifically related travel demand to activity participation in time and space. By the 1980s, activity-based analysis had taken root with studies such as Recker et al. (1986a,b) and Kitamura (1988). Subsequent models (e.g., Axhausen and Gärling, 1992; Ettema et al., 1993; Gärling et al., 1994; Bhat and Koppelman, 1999; Bowman and Ben-Akiva, 2001; Bhat et al., 2004; Pendyala et al., 2005; Bellemans et al., 2010) expanded on the problem significantly. Many focus on such behavioral questions as explaining differences between travel behavior for work activities and social activities, or the need to drop off kids at school. There is less consideration of the impacts of system design factors such as transit schedules and facility capacities. Pinjari and Bhat (2011) provide a detailed survey of the history and synthesis of these models.

As an illustration, Bhat's CEMDAP model has a scheduling component as presented in Table 4.1 (number of stops, durations, locations, and departure times). One can draw two immediate conclusions. First, travel decisions are so complex that they require numerous models just to be able to fit to observed data. Second, by breaking up the joint travel decisions made throughout the day into all these components, direct relationships between travel decisions and multimodal system designs are difficult to pinpoint. The park-and-ride example is a nonlinear and highly structured interdependency between departure time, mode choice, activity destination choice, among others.

Studies such as Lam and Yin (2001), Lin et al. (2008), Ramadurai and Ukkusuri (2010), Konduri (2012), Liu et al. (2015a), Halat et al. (2016), and Liu et al. (2017) appear to offer solutions to this lack of structured interaction. Such models impose a Nash game upon travelers throughout the day under a setting where links in the infrastructure network, or in some cases even at the origin-destination (OD) pair level, exhibit congestion effects like in dynamic traffic assignment (DTA) models. However, unlike traditional DTA models, these activity-based network equilibrium models impose strong assumptions that trade-offs between one activity schedule versus another are primarily driven by congestion effects on the transportation network with other travelers at a 24-h decision-making timescale. For this to work effectively, congestion impacts on links in a network at a 5-min interval need to realistically measure up against other paths in the same time

Table 4.1 Modes used in scheduling portion of CEMDAP

Model	Econometric structure
Commute mode	Multinomial logit
Number of stops in work-home commute	Ordered probit
Number of stops in home-work commute	Ordered probit
Number of after-work tours	Ordered probit
Number of work-based tours	Ordered probit
Number of before-work tours	Ordered probit
Tour mode	Multinomial logit
Number of stops in a tour	Ordered probit
Home-work stay duration before a tour	Regression
Activity type at stop	Multinomial logit
Activity duration at stop	Linear regression
Travel time to stop	Linear regression
Stop location	Spatial location choice
Number of independent tours	Ordered probit
Decision to undertake an independent tour before pickup/joint discretionary	Binary logit
Decision to undertake an independent tour after pickup/joint discretionary	Binary logit
Tour mode	Multinomial logit
Number of stops in a tour	Ordered probit
Number of stops following a pickup/drop-off stop in a tour	Ordered probit
Home stay duration before a tour	Regression
Activity type at stop	Multinomial logit
Activity duration at stop	Linear regression
Travel time to stop	Linear regression
Stop location	Spatial location choice
Departure time from home	Regression
Activity duration at stop	Regression
Travel time to stop	Regression
Location of stop	Spatial location choice
School-home commute time	Regression
Home-school commute time	Regression
Mode for independent discretionary tour	Multinomial logit
Departure time from home for independent discretionary tour	Regression
Activity duration at independent discretionary stop	Regression
Travel time to independent discretionary stop	Regression
Location of independent discretionary stop	Spatial location choice

interval, then in neighboring 5-min intervals, 30-min intervals, hourly intervals, and so on, up to the tactical decision-making level by time of day. Realistically, the impact of a link delay due to congestion should be buried underneath a mountain of other response options before it reaches the sensitivity of a user's scheduling decision. However, such simplifications in network size or time interval to overcome computational challenges will artificially inflate the significance of link congestion effects on these tactical scheduling decisions.

In addition, congestion effects cannot be uniformly applied across all elements of the user's schedule. For example, at an activity level, not all congestion effects are negative externalities. A well-attended restaurant tends to be preferred over an empty one. A 5-min delay at a congested link in a network for a commuter on the way home should not have the same weight as a 5-min delay while waiting for a busy commuter train to arrive to get to work in the morning. Some congestion effects are more directly linked to physical capacities, such as parking capacity or transit station queue capacity. In these circumstances, congestion effects at the infrastructure link level are negligible compared to user activity scheduling effects as far as the user's tactical behavior is concerned.

[Recker \(1995\)](#) proposed a user activity scheduling model called the Household Activity Pattern Problem (HAPP) which focuses solely on scheduling decisions and effects, given an activity agenda. Variants of the model were shown to handle intrahousehold interactions. [Recker \(2001\)](#) drew parallels between the HAPP model and utility maximization forecast models. The model has been used to evaluate effects of household ridesharing policy on vehicle emissions ([Recker and Parimi, 1999](#)), measuring the inconvenience of alternative fuel vehicles with limited fueling infrastructure ([Kang and Recker, 2014](#)), inferring the desired arrival times and scheduling preferences for individuals in southern California ([Chow and Recker, 2012](#)), and measuring the capacity effects of a park and ride facility in the Greater Toronto Area ([Chow and Djavadian, 2015](#)).

Methodological extensions to this model have also been made. The model can handle destination choice through separate efforts by [Kang and Recker \(2013\)](#) and [Chow \(2014\)](#). In the latter study, a methodology is introduced to efficiently apply the HAPP model to analyze different scenarios based on reoptimization. [Gan and Recker \(2008, 2013\)](#) extended the model to consider dynamic rescheduling and stochastic scheduling. [Chow and Recker \(2012\)](#) addressed the parameter estimation problem to fit the

HAPP model with goal arrival times to observed user schedules as an inverse optimization model. This has since opened a new area of research in inverse transportation problems which is discussed in [Chapter 5](#). [Chow and Djavadian \(2015\)](#) extended the HAPP model to the multimodal household activity pattern problem (mHAPP) model, which accounts for capacitated multimodal infrastructure networks. [Chow and Nurumbetova \(2015\)](#) extended the HAPP model to multiple days (see [Lee and McNally, 2003](#); [Schlich and Axhausen, 2003](#)) using an inventory routing extension to capture the “needs” constraint ([Arentze and Timmermans, 2009](#)). [Liu et al. \(2017\)](#) linked the HAPP scheduling model with dynamic traffic assignment to capture operational congestion effects.

A similar type of model is also seen in the work of [Arentze and Timmermans \(2004\)](#) and [Liao et al. \(2010, 2013\)](#), although these models focus more on state decisions during a trip than on activity scheduling and sequencing.

The mHAPP model is adopted as a base model to be introduced and illustrated in the following sections.

4.3.2 Model Formulation and Analytical Properties

The following model is used to forecast the schedule chosen by a user given a set of compulsory activities, available transportation facilities, and desired arrival times and durations for each activity. It can serve as the scheduling model needed for different multiagent activity simulators like MATSim or as the sampled response of a model of a heterogeneous population.

An activity system G composed of zones $z \in G$ is populated by \mathbb{P} and a multimodal transport system V , which is composed of multiple modes $v \in \{0, V\}$. Walking mode is defined by a mode 0, driving is mode 1, and other types of modes are defined by $2, \dots, |V|$. A set of transport nodes $\Lambda = \{\Lambda_1, \dots, \Lambda_{|V|}\}$ is separated into individual modes. For example, Λ_1 is associated with the driving mode and represents a parking facility. $\Lambda_{v \geq 2}$ represents a stop or station associated with the corresponding mode. In this model, users traveling from one activity to another go from an activity zone $z(r) \in G_i$ to different transport nodes before ending at another activity zone $z(s) \in G_i$. Modes are defined by the types of nodes at each end. A link that connects a home zone or parking facilities to each other pertains to driving. Links between two nodes from the same mode represent routes covered by that mode. Links between two nodes from different modes belong to walking mode.

The mHAPP is a mixed integer linear programming problem that considers the scheduling of a single user $i \in S$. An individual resides in one zone

and conducts activities in a subset of zones $G_i \subset G$, where $z[u] \in G_i$ is a zone corresponding to an activity $u \in W$. The following parameters are defined for each user. Unlike links in a traffic assignment model in Chapter 3 that represent physical road segments, the links connecting the transport nodes and activity nodes together are directed links that represent either OD pairs, paths, or bottleneck links. The choice depends on the design of the model for different trade-offs between certain infrastructure capacities to activity scheduling.

Observable parameters:

O_i is the home node of individual i in a zone in G_i , which may be indexed as 0;

N_i is the set of activity nodes u of individual i , divided into N_i^+ for the activity participation and corresponding N_i^- for the return from the activity, where each $u \in N_i^+$ is located in a zone in G_i and each activity $(u + |N_i^+|) \in N_i^-$ is located at the individual's home zone;

D_i is the final return home node of individual i , which may be indexed as $|N_i| + 1$;

$W = NU \cup OUD$ is the set of all nodes in the population;

$t_{z(u), z(w)}$ is the travel time from one origin $u \in W$ to one destination $w \in W$;

$c_{z(u), z(w)}$ is the travel cost between nodes u and w ;

d_u is the duration at activity node u , or the average wait time at a transport station for a frequency-based transport node;

a_u, b_u are optional early and late arrival time windows for a node u for capturing operating hours or to include transit schedules;

p_u^1 is a variable parking fee for a node $u \in \Lambda_1$;

p_u^2 is a fixed parking fee for a node $u \in \Lambda_1$;

\mathcal{M} is a sufficiently large constant.

Latent parameters:

g_u is the goal arrival time of user i to a node $u \in N_i^+$;

$\beta_i = \{\beta_{i1}^0, \dots, \beta_{i,|K|}^0, \beta_{i1}^{T,e}, \dots, \beta_{i,|N|}^{T,e}, \beta_{i1}^{T,l}, \dots, \beta_{i,|N|}^{T,l}\}$ is a vector of weights for user i where β_i^0 is the set of weights for $|K|$ objectives, and $\beta_i^{T,e}, \beta_i^{T,l}$ are weights associated with early and late arrival relative to a goal arrival time.

Decision variables for individual i (index left out of these variables for convenience):

X_{uw}^{rs} is a binary variable that indicates a route chosen by a user from node $u \in \{r, \Lambda\}$ to $w \in \{s, \Lambda\}$, where the route originating or ending at an activity node would have $r = u$ or $s = w$;

T_u , $u \in N$, is a continuous nonnegative variable that indicates the time at which an activity node is started;

T_u^r , $u \in \Lambda$, is the start time at a parking or transport facility to travel from activity node r to activity node s ;

T_O , T_D are the initial departure time and final return home times;

Q_u^{rs+} is a continuous variable for the time that an individual picks up a vehicle from a parking facility u to go from node r to s ;

Q_u^{rs-} is a continuous variable for the time that an individual drops off a vehicle at a parking facility u to go from node r to s ;

P_u^e , P_u^l are nonnegative early and late arrival deviations from a goal arrival time g_u at node u .

The model framework considers general trade-offs between desires to get to different activities that are either required (long-term utility like work) or produce short-term utility for the user, subject to preferences to stay at home or to travel with minimal disutility.

The travel cost parameter $c_{z(u), z(w)}$ models capacity effects by adding their corresponding dual prices, that is, $c_{z(u), z(w)} = c'_{z(u), z(w)} + \lambda_{z(u), z(w)}$, where $c'_{z(u), z(w)}$ is the uncapacitated link cost and $\lambda_{z(u), z(w)}$ is the optimal dual price corresponding to a binding load capacity. The parameter $t_{z(u), z(w)}$ captures in-vehicle travel time. If used for fixed route transit systems, average wait time at a station is modeled using the activity duration parameter d_u . The transit schedule is explicitly modeled using the time windows at a station by duplicating a node for each run of interest.

The objective of a user varies from one to another; we model this by allowing the objective parameters to be correlated variates across the population. The objective weights are latent parameters but can be calibrated from observed activity scheduling data. A methodology for parameter estimation was developed by [Chow and Recker \(2012\)](#) and is discussed in more detail in [Chapter 5](#) on inverse optimization.

Consider the following weighted objective function in Eq. (4.2). For convenience, the i index is dropped out.

$$\min -U = \sum_{k \in K} \beta_k^0 Z_k + \sum_{u \in N^+} \beta_u^{T,e} P_u^e + \sum_{u \in N^+} \beta_u^{T,l} P_u^l \quad (4.2a)$$

where Z_k is represented by different objectives shown in Eqs. (4.2b)–(4.2g), with each objective having a different coefficient β_k^0 .

$$\sum_{r \in \{N, O\}} \sum_{s \in \{N, D\}} \sum_{u \in \{r, \Lambda\}} \sum_{w \in \{s, \Lambda\}} c_{uw} X_{uw}^{rs} \quad (4.2b)$$

$$\sum_{r \in \{N, O\}} \sum_{s \in \{N, D\}} \sum_{u \in \{\Lambda\}} \sum_{w \in \{s, \Lambda\}} t_{uw} X_{uw}^{rs} \quad (4.2c)$$

$$\begin{aligned} & \sum_{u \in \Lambda_1} \left(p_u^1 \left(\sum_{r \in \{N, O\}} \sum_{s \in \{N, D\}} Q_u^{rs+} - \sum_{r \in \{N, O\}} \sum_{s \in \{N, D\}} Q_u^{rs-} \right) \right. \\ & \quad \left. + p_u^2 \sum_{r \in \{N, O\}} \sum_{s \in \{N, D\}} \sum_{w \in \{r, \Lambda\}} X_{wu}^{rs} \right) \end{aligned} \quad (4.2d)$$

$$- Y_v, \quad \forall v \in V \quad (4.2e)$$

$$\sum_{u \in N^+} (T_{u+|N^+|} - T_u) \quad (4.2f)$$

$$T_D - T_O \quad (4.2g)$$

Eq. (4.2b) minimizes the modal travel cost, which includes fares and tolls. Eq. (4.2c) minimizes the travel time. Eq. (4.2d) is the parking cost. Eq. (4.2e) is the mode preference, defined to be nonpositive (the least preferred mode would have a $\beta_k^0 = 0$). Eq. (4.2f) minimizes the delay from returning home after conducting an activity. Eq. (4.2g) minimizes the length of the day spent outside of home. The objectives are subject to the following constraints.

Subject to

$$\sum_{s \in N} \sum_{w \in \{s, \Lambda\}} X_{nw}^{rs} = 1, \quad r \in \{O, N\} \quad (4.3)$$

$$\sum_{r \in N} \sum_{u \in \{r, \Lambda\}} X_{us}^{rs} = 1, \quad s \in \{D, N\} \quad (4.4)$$

$$\sum_{w \in \{\Lambda\}} X_{uw}^{rs} - \sum_{w \in \{r, \Lambda\}} X_{wu}^{rs} = 0, \quad u \in \Lambda, r \in \{O, N\}, s \in \{D, N\} \quad (4.5)$$

$$\sum_{r \in N} \sum_{u \in \{r, \Lambda\}} X_{uO}^{rO} + \sum_{s \in N} \sum_{w \in \{s, \Lambda\}} X_{Dw}^{Ds} = 0 \quad (4.6)$$

$$\sum_{r \in \{N, O\}} \sum_{s \in \{N, D\}} \sum_{w \in \{\Lambda_1, N^-, D\}} X_{uw}^{rs} - \sum_{r \in \{N, O\}} \sum_{s \in \{N, D\}} \sum_{w \in \{\Lambda_1, N^-, O\}} X_{wu}^{rs} = 0 \quad (4.7)$$

$$X_{wu}^{rs} = 0, \quad u \in \Lambda_1 \quad (4.7)$$

$$T_u + P_u^e - P_u^l = g_u, \quad u \in N^+ \quad (4.8)$$

$$T_u + d_u - T_{|N^+|+u} \leq 0, \quad u \in N^+ \quad (4.9)$$

$$-T_u^{rs} + Q_u^{rs+} \geq \left(\sum_{w \in \{\Lambda_1, N^-, D\}} X_{uw}^{rs} - 1 \right) \mathcal{M}, \quad u \in \Lambda_1, r \in \{O, N\}, s \in \{N, D\} \quad (4.10a)$$

$$-T_u^{rs} + Q_u^{rs+} \leq \left(1 - \sum_{w \in \{\Lambda_1, N^-, D\}} X_{uw}^{rs} \right) \mathcal{M}, \quad u \in \Lambda_1, r \in \{O, N\}, s \in \{N, D\} \quad (4.10b)$$

$$T_u^{rs} - Q_u^{rs-} \geq \left(\sum_{w \in \{O, \Lambda_1, N^-\}} X_{wu}^{rs} - 1 \right) \mathcal{M}, \quad u \in \Lambda_1, r \in \{O, N\}, s \in \{N, D\} \quad (4.10c)$$

$$T_u^{rs} - Q_u^{rs-} \leq \left(1 - \sum_{w \in \{O, \Lambda_1, N^-\}} X_{wu}^{rs} \right) \mathcal{M}, \quad u \in \Lambda_1, r \in \{O, N\}, s \in \{N, D\} \quad (4.10d)$$

$$T_r + d_r + t_{rs} - T_s \leq (1 - X_{rs}^{rs}) \mathcal{M}, \quad r, s \in N \quad (4.11a)$$

$$T_r + d_r + t_{rw} - T_w^{rs} \leq (1 - X_{rw}^{rs}) \mathcal{M}, \quad r \in N, s \in \{D, N\}, w \in \Lambda \quad (4.11b)$$

$$T_u^{rs} + t_{us} - T_s \leq (1 - X_{us}^{rs}) \mathcal{M}, \quad r \in \{N, O\}, s \in N, u \in \Lambda \quad (4.11c)$$

$$T_u^{rs} + d_u + t_{uw} - T_w^{rs} \leq (1 - X_{uw}^{rs}) \mathcal{M}, \quad u, w \in \Lambda, r \in \{O, N\}, s \in \{N, D\} \quad (4.11d)$$

$$T_O + t_{Os} - T_s \leq (1 - X_{Os}^{Os}) \mathcal{M}, \quad s \in N \quad (4.11e)$$

$$T_r + t_{rD} - T_D \leq (1 - X_{rD}^{rD}) \mathcal{M}, \quad r \in N \quad (4.11f)$$

$$T_O + t_{Ow} - T_w^{Os} \leq (1 - X_{Ow}^{Os}) \mathcal{M}, \quad w \in \Lambda, s \in N \quad (4.11g)$$

$$T_u^{rD} + t_{uD} - T_D \leq (1 - X_{uD}^{rD}) \mathcal{M}, \quad u \in \Lambda, r \in N \quad (4.11h)$$

$$\mathcal{M} \sum_{r \in \{N, O\}} \sum_{s \in \{N, D\}} \sum_{w \in \Lambda_v} X_{rw}^{rs} \geq Y_v, \quad v \in V \quad (4.12)$$

$$T_r \geq a_r, \quad r \in N^+ \quad (4.13a)$$

$$T_r \leq b_r, \quad r \in N^+ \quad (4.13b)$$

$$T_u^{rs} \geq a_u, \quad u \in \Lambda, r \in \{O, N\}, s \in \{N, D\} \quad (4.13c)$$

$$T_u^{rs} \leq b_u, \quad u \in \Lambda, r \in \{O, N\}, s \in \{N, D\} \quad (4.13d)$$

$$Q_u^{rs-} \leq \mathcal{M} \sum_{w \in \{s \in N^+, \Lambda\}} X_{uw}^{rs}, \quad u \in \Lambda, r \in \{O, N\}, s \in \{N, D\} \quad (4.14a)$$

$$T_u^{rs} \leq \mathcal{M} \sum_{w \in \{s, \Lambda\}} X_{uw}^{rs}, \quad u \in \Lambda, r \in \{O, N\}, s \in \{N, D\} \quad (4.14b)$$

$$X_{uw}^{rs}, Y_v \in \{0, 1\} \quad (4.15a)$$

$$T_u, T_u^{rs}, T_O, T_D, P_u^e, P_u^l, Q_u^{rs+}, Q_u^{rs-} \geq 0 \quad (4.15b)$$

The flow conservation constraints in Eqs. (4.3)–(4.6) generally apply to all nodes. Eq. (4.7) requires visits to a parking node to be done in pairs for picking up and dropping off vehicles. This condition also allows parking durations to be obtained with Eq. (4.10). Eq. (4.8) is the goal arrival constraint that allows a user to get to a destination early or late at a penalty. Eq. (4.9) sets the arrival time for returning home from an activity to occur after conducting the activity. Eq. (4.11) ensures that arrival times are connected in relation to routes chosen to eliminate subtours and update arrival times. Eq. (4.12) is used to determine whether a mode is used that day, which would impart a utility to the user based on the objective in Eq. (4.2e). The remaining constraints are optional time window constraints and conditional constraints (Eq. 4.14) to keep the arrival time and parking time values to zero if not visited.

As shown in Chow (2014) and Chow and Nurumbetova (2015), the HAPP model and its extensions can be modified to include destination choice and multiple day scheduling based on needs satisfaction. For simplicity, only the base mHAPP model is presented here.

The optimization model in Eqs. (4.2)–(4.14) is a mixed integer linear programming problem that can be solved using algorithms designed for those problems such as branch and bound algorithms. As a variant of the pickup and delivery problem (see Solomon and Desrosiers, 1988), the mHAPP model is an NP-hard problem for which the optimality of a solution cannot be verified in exponential time. Effective algorithms have been developed to solve these problems (see Cordeau and Laporte, 2003). Kang and Recker (2013) developed a solution algorithm based on Dumas et al. (1991) to solve these problems exactly. Chow (2014) developed insertion algorithms and a genetic algorithm, both heuristics, to address large-scale problems in reasonable computation time. These are discussed in more detail in Section 4.3.3.

Consider an illustrative example of the mHAPP model in Exercise 4.1.

Exercise 4.1

For the activity system shown in Fig. 4.4 and two users' observed parameters, design values of objective weights β and goal arrival times g for three scenarios: User 1 drives, User 1 takes transit, and User 2 takes transit. Summarize their performances and draw out their time-space trajectories.

User 1 would cross the bridge by car if the utility for automobile is set high enough, that is, if the only objective is $-\beta_{mode,1}^0 Y_1$, where $\beta_{mode,1}^0 \gg 0$. To ensure a unique solution, cost minimization is also considered, $\beta_{cost}^0 \sum_{r \in \{N, O\}} \sum_{s \in \{N, D\}} \sum_{u \in \{r, A\}} \sum_{w \in \{s, A\}} c_{uw} X_{uw}^{rs}$. If User 1 has $g_1 = 9$ a. m. and $g_2 = 6$ p. m., the output of mHAPP is:

Total in-vehicle travel time: 1 h 30 min

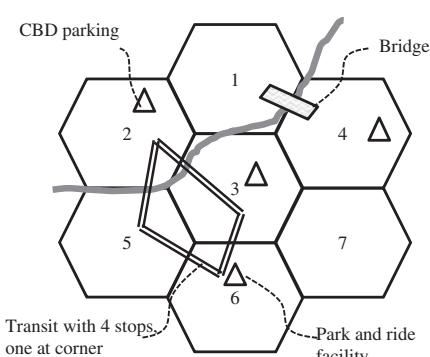
Total access/egress time: 20 min

Total time outside of home: 11 h 10min

Total cost: \$12.67 (\$8.17 CBD parking, \$4.50 gas)

Total idle/wait time: 20 min

Total time vehicle parked: 9 h 40 min (8 h 10 min in CBD parking, 1 h 30 min in zone 4 parking)



All driving (transit) times from zone centroid to a neighboring zone centroid is assumed to be $\frac{1}{4}$ h ($\frac{1}{3}$ h), auto cost is \$3/h, all walking access between two facilities in the same zone is set to $\frac{1}{12}$ h, all arrival penalties are $\beta_u^{T,e} = 0.70$, $\beta_{T,l}^{T,l} = 11$.

Transport facilities

CBD parking: $p_1^2 = \$1/\text{hr}$

Zone 3 parking: $p_2^1 = \$0$

Zone 4 parking: $p_3^1 = \$0$

PnR parking: $p_4^1 = \$5$

Transit station at all zones: $d_5(z = 2) = d_6(z = 3) = d_7(z = 6) = d_8(z = 5) = \frac{1}{12}$ h

User 1

Home zone: 7, owns a car

Activity 1 (work) – zone: 2, $d_1 = 8$ l

Activity 2 (dinner) – zone: 4, $d_2 = 1$ h

Pays transit fares \$2/trip

User 2

Home zone: 5, owns a car

Activity 1 (work) – zone: 2, $d_1 = 6$ km

Activity 2 (shopping) – zone: 3, $d_2 = 2$ h
Fare card costs \$4 for unlimited use all day if used that day

Fig. 4.4 Activity system to illustrate flexibility of model to capture different scheduling preferences

For User 1 to take transit, they need to get to the park-and-ride facility. Therefore having an objective $-\beta_{mode,2}^0 Y_2$ where $Y_2=1$ if transit is taken, with $\beta_{mode,2}^0 \gg 0$, would switch the user to this mode. Under the same cost minimization objective and goal arrival times, $g_1=9$ a. m. and $g_2=6$ p. m., the output of mHAPP is:

```

Home (8 a.m.) → PnR parking (8 : 15 a.m.)
    → Transit zone 6 (8 : 20 a.m. – 8 : 25 a.m.)
    → Transit zone 2 (8 : 45 a.m.)
    → Activity 1 (8 : 50 a.m. – 4 : 50 p.m.)
    → Transit zone 2 (4 : 55 p.m. – 5 : 00 p.m.)
    → Transit zone 6 (5 : 20 p.m.) → PnR parking (5 : 25 p.m.)
    → Zone 4 parking (5 : 55 p.m.) → Activity 2 (6 p.m. – 7 p.m.)
    → zone 4 parking (7 : 05 p.m.) → Home (7 : 20 p.m.)

```

Total in-vehicle travel time: 1 h 40 m.

Total access/egress time: 30 m.

Total time outside of home: 11 h 20 m.

Total cost: \$12 (\$4 transit fares, \$5 PnR parking fee, \$3 gas).

Total idle/wait time: 10 m

Total time vehicle parked: 10 h 20 m (9 h 10 m in PnR, 1 h 10 m in zone 4 parking).

Because of preference to take transit, User 1 actually shifts their schedule earlier in the day to arrive at work 10 min earlier than desired. This assumes that the benefit of taking transit compared to taking auto is more beneficial than the cost of arriving earlier to work.

User 2 would take transit if the transit mode objective is sufficiently high, that is, if it is $-\beta_{mode,2}^0 Y_2$ where $Y_2=1$ if transit is taken. A smart card is used to pay \$4 to allow unlimited transit trips in a day. The cost is not included in c_{inv} for the transit trips, but instead included in the $\beta_{mode,2}^0$ for this user, that is, $\beta_{mode,2}^0 = \beta_{mode,2}^{0'} - \4 if the coefficients are calibrated to be in units of dollars and $\beta_{mode,2}^{0'}$ is the original parameter. Under the goal arrivals of $g_1=8$ a.m. and $g_2=5$ p.m. and a cost minimization objective, the following is a solution of mHAPP:

```

Home (7 : 25 a.m.) → Transit zone 5 (7 : 30 a.m. – 7 : 35 a.m.)
    → Transit zone 2 (7 : 55 a.m.)
    → Activity 1 (8 a.m. – 2 p.m.)
    → Transit zone 2 (2 : 05 p.m. – 2 : 10 p.m.)
    → Transit zone 5 (2 : 30 p.m.)
    → Home (2 : 35 p.m. – 4 : 05 p.m.)
    → Transit zone 5 (4 : 10 p.m. – 4 : 15 p.m.)
    → Transit zone 3 (4 : 55 p.m.)
    → Activity 2 (5 p.m. – 7 p.m.)
    → Transit zone 3 (7 : 05 p.m. – 7 : 10 p.m.)
    → Transit zone 5 (7 : 50 p.m.) → Home (7 : 55 p.m.)

```

Total in-vehicle travel time: 2h.

Total access/egress time: 40m.

Total time outside of home: 11 h.

Total cost: \$0 (\$4 transit fare but it is in the disutility objective, not the cost).

Total idle/wait time: 20m

Note that User 2 could have also taken a car for Activity 2, but that would cost an additional \$1.50 gas for the round trip. If minimizing time spent outside of home is a strong enough objective relative to cost minimization, the user would switch from this schedule to use the car.

These patterns are illustrated in Fig. 4.5, showing how the same mHAPP model, with different parameters even for the same user can exhibit very different outcomes that interact with the transport system.

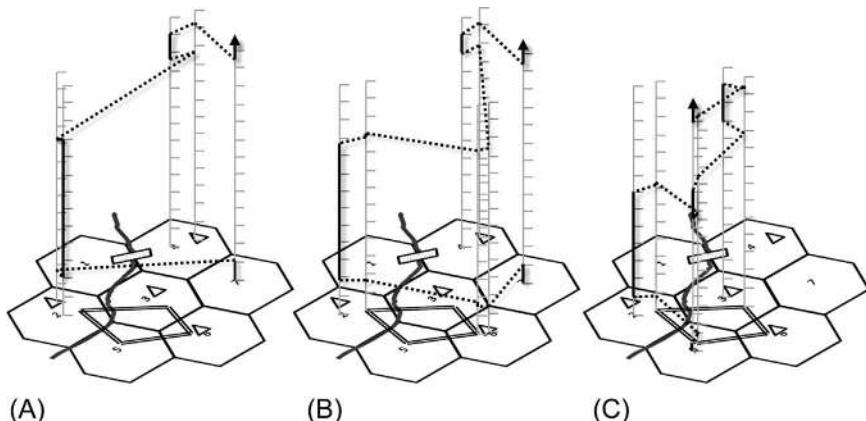


Fig. 4.5 Three solutions of mHAPP based on different parameters for user 1 (A, B) and user 2 (C).

From the perspective of the transport system agency or operator, this model framework allows one to design different control variables: facility load capacity (reflected in a dual price added to c_{uv}), pricing fares or parking fees, setting transit headways impacting time windows or average wait times, updating the travel times with congested travel times, location of stations, or provision of routes. In return, running the mHAPP model on each user of a full population or a sampled subpopulation provides a modeler with information about heterogeneous scheduling responses.

4.3.3 Solution Algorithms

The family of HAPP models belong to the pickup and delivery problem, which is a mixed integer linear programming problem that is NP-hard. In practice, effective algorithms are needed to solve the optimization model. For small problems involving one to four activities (10 nodes), commercial integer programming solvers (e.g., CPLEX, GAMS, the `intlinprog` function in MATLAB) using branch and bound algorithms are adequate. In most basic activity scheduling problems without destination choice or multimodal network interaction, there are typically four or less activities conducted in a day (Golob, 2000).

A branch and bound algorithm (Land and Doig, 1960) is a well-known solution method for solving integer programming problems. The intuition behind the algorithm is to iteratively add constraints to the integers (branch), and relax the integral constraint to solve the resulting linear program (LP), which can be solved efficiently (Dantzig and Thapa, 2003). By testing the relaxed LP, the algorithm gains information from which it can close certain branches without having to explore them further (bound). A version of it shown in Bradley et al. (1977) is presented in Algorithm 4.1.

Algorithm 4.1 (Based on Bradley et al., 1977). Branch and Bound Algorithm for Integer Programming

Inputs: Integer programming parameters c , A , b , and decision variables $X \in \mathbb{Z}$, structured as a maximization problem: $Z = \{\max c^T X : AX \leq b, X \in \mathbb{Z}\}$

1. Initialize. Set lower bound $Z^* = -\infty$ and upper bound \bar{Z} from associated LP. Apply steps 2–4 to whole problem. If fathomed, stop.
2. Branching. Among unfathomed subproblems, select one most recently created and create 2 branches by inserting constraints to the parent associated LP: $x_j \leq \lfloor x_j^* \rfloor$ for one and $x_j \geq \lfloor x_j^* \rfloor + 1$ for other.
3. Bounding. For each new subproblem, solve associated LP; if upper bound can be updated, do so.
4. Fathom. For each subproblem, apply 3 fathom tests:
 1. Test 1. If new upper bound $\leq Z^*$
 2. Test 2. If no feasible solution.
 3. Test 3. If solution is a feasible integer problem. In this case, if the bound $> Z^*$, update Z^* .
5. Optimality test. Stop when there are no remaining subproblems, or if $\bar{Z} - Z^*$ is within tolerance.

Outputs: X^*

The algorithm is illustrated in [Exercise 4.2](#).

Exercise 4.2

(From [Bradley et al., 1977](#)). Solve the following program using [Algorithm 4.1](#).

$$\max Z = 5x_1 + 8x_2$$

Subject to

$$x_1 + x_2 \leq 6$$

$$5x_1 + 9x_2 \leq 45$$

$$x_j \in \mathbb{Z}^+, j=1,2$$

1. Initialize: $Z^* = -\infty$.
2. First we relax the integral constraint by converting $x_j \in \mathbb{Z}^+$ to $x_j \geq 0$. The solution of the associated LP to this relaxation is $x_1 = 2.25$, $x_2 = 3.75$, $\bar{Z} = 41.25$. Since there are noninteger solutions that should be integer, we keep going.
3. Pick one noninteger variable that should be an integer (x_2) and branch it: $x_2 \leq 3$ [L_2], $x_2 \geq 4$ [L_1].
4. Solve L_1 : $x_1 = 1.8$, $x_2 = 4$, $Z = 41$. Since we have a noninteger solution, we branch to [L_3] $x_1 \geq 2$ and [L_4] $x_1 \leq 1$.
5. L_3 is not feasible so we fathom it. Solving L_4 leads to: $x_1 = 1$, $x_2 = 4.444$, $Z = 40.55$. We need to branch to form [L_5] $x_2 \leq 4$ and [L_6] $x_2 \geq 5$.
6. Solve L_5 (which, at this point, looks like this: $Z = \{\max(5x_1 + 8x_2) : x_1 + x_2 \leq 6, 5x_1 + 9x_2 \leq 45, x_2 \geq 4, x_1 \leq 1, x_2 \leq 4, x_j \geq 0\}$) to obtain $x_1 = 1$, $x_2 = 4$, $Z^* = 37$. The objective value is bounded here because the solution is feasible and is better than the prior bound. If we stop the algorithm here, we know our solution has an optimality gap of $\frac{(41-37)}{41} = 10.8\%$. We still have two open branches.
7. Solve L_6 and get $x_1 = 0$, $x_2 = 5$, $Z^* = 40$. The bound has been updated with the higher value. Stopping the algorithm now would result in a 2.5% optimality gap.
8. Solve L_2 and get $x_1 = x_2 = 3$, $Z = 39$. Since this objective value is lower than the current bound, we can fathom it. As there are no more unfathomed branches, the algorithm terminates.

For larger instances, or for mHAPP and selective HAPP with destination choice ([Kang and Recker, 2013](#); [Chow, 2014](#)), customized algorithms are needed. As noted in [Kang and Recker \(2013\)](#), numerous algorithms have been developed over the years to solve large-scale pickup and delivery problems with time windows, as reviewed in [Cordeau and Laporte \(2003\)](#). For example, problems with up to 2500 locations have been successfully solved

in the literature. For problems with destination selection, a set of zones accessible to a household within their space-time prism (Miller, 1991) may range up to 100 zones or so in practice. Therefore this is well within the range to solve.

Dumas et al. (1991) proposed a dynamic programming-based solution method intended to solve capacitated fleet routing. It decomposes the routing problem into a restricted master problem and a shortest path subproblem that uses the dual variables from the restricted master problem. The HAPP model class, and particularly the mHAPP model, cannot be solved exactly using this approach because the objective function may include objectives that are nonadditive across links (Eqs. (4.2d)–(4.2g)) with soft arrival times having early and late penalties. As such, the dynamic programming approach would not be able to account for these trade-offs when recursively adding new links to a sequence. The method in Kang and Recker (2013) is designed for hard time windows with only link-additive costs (Eqs. (4.2b)–(4.2c)) in the objective. Finding an effective exact algorithm for large-scale mHAPP (and its selective version, mSHARP (Chow, 2014)) remains an open research challenge.

We turn to heuristics for now. Chow (2014) proposed a genetic algorithm (GA) to obtain a solution for these types of problems. A GA is a type of stochastic search algorithm that randomly generates new solutions based on pruning “generations” of solutions based on fitness and evolution. It has been shown to converge as the number iterations approach infinity, but the rate of convergence is highly dependent on the problem structure, dimensionality, and parametric design of the algorithm. As such, it is not possible to ascertain the optimality of a solution after a finite number of iterations.

Nonetheless, it has been shown to be effective in generating good quality vehicle routing solutions (Baker and Aye chew, 2003) and is adopted here as the de facto algorithm for large-scale instances. GA is also used in MATSim to find good fitting schedules for individuals in the synthetic population. The GA from Chow (2014) is shown in Algorithm 4.2.

Algorithm 4.2 Genetic Algorithm for Selective HAPP (G-SHARP) for Single Individual Household

Input: HAPP model parameters, GA parameters: population size E , number of generations G , initial population activity choice threshold Δ , mutation rate θ , infeasible solution penalty λ .

Initialization

1. Generate $E \times N^+$ percentiles μ using Latin Hypercube Sampling (see Chow et al., 2010a)

2. For $p \leq E$ and $u \in N^+$, if $\mu[p, u] > \Delta$, $Y_{mu}[p] := 1$

3. For $u \in N^-$, if $Y_{m,u-|N^+|}[p]$, $Y_{mu}[p] := 1$

Generate random feasible sequences for binary variables

4. $\Omega_p := \text{randperm}[N^+]$ ($\Omega_p = p^{th}$ sequence in E)

5. For $u \in N^-$, if $\Omega_p[u] < \Omega_p[u - |N^+|]$, randomly reassign to occur after $\Omega_p[u - |N^+|]$

6. For $1 \leq i \leq |\Omega_p|$,

a. $X_{\Omega_p[i], \Omega_p[i+1]}[p] := 1$, $X_{O, \Omega_p[1]}[p] := 1$, $X_{\Omega_p[|\Omega_p|], D}[p] := 1$

Optimize the continuous variable subproblem given a set of feasible binary variables

7. $T[p] := LP[X[p]]$

8. $\text{penalty}[p] := \lambda \left[\sum_{u \in N^+} \max[T_u - b_u, 0] + \sum_{u \in N^+} \max[a_u - T_u, 0] \right]$

9. Update objective value $-U[X, T, Y, \text{penalty}]$ for each $p \leq E$

GA updates: population crossbreed, mutate, selection via fitness, and repeat

10. For $2 \leq g \leq G$,

a. For $p \leq E$, $\Omega_{p,g} := \text{crossbreed}[\Omega_{p1,g-1}, \Omega_{p2,g-1}]$ as per Tagsetiren and Smith (2000)

b. For $m \in M$ and if $q_m > 0$,

i. Randomly select $u \in N_m^+$ so that $\sum_{u \in N_m^+} Y_{mu}[p, g] = q_m$

c. For $u \in N^+$,

i. if $Y_{mu}[p, g] = 1$ and $\text{rand} < \frac{\theta}{\sum q_m + 1}$, $Y_{mu}[p, g] := 0$

d. Randomly select $w \in N_m^+$, $Y_{mw}[p, g] := 1$

e. For $u \in N^+$,

i. If $Y_{mu}[p, g] = 1$, $Y_{m,u+|N^+|}[p, g] := 1$, else $Y_{m,u+|N^+|}[p, g] := 0$

f. For $u \in N^-$, if $\Omega_{p,g}[u] < \Omega_{p,g}[u - |N^+|]$, randomly reassign to occur after $\Omega_{p,g}[u - |N^+|]$

g. For $1 \leq i \leq |\Omega_{p,g}|$,

i. $X_{\Omega_{p,g}[i], \Omega_{p,g}[i+1]}[p, g] := 1$, $X_{O, \Omega_{p,g}[1]}[p, g] := 1$, $X_{\Omega_{p,g}[|\Omega_{p,g}|], D}[p, g] := 1$

h. $T[p, g] := LP[X[p, g]]$

i. $\text{penalty}[p] := \lambda \left[\sum_{u \in N^+} \max[T_u - b_u, 0] + \sum_{u \in N^+} \max[a_u - T_u, 0] \right]$

j. Update objective value $-U[X, T, Y, \text{penalty}]$ for each $p \leq E$

k. Merge best E values from sort of $\{-U[1:E, g]\} \cup \{-U[1:E, g-1]\}$

Output: X , T , Y

The binary variable Y_{mu} indicates whether an activity u is visited for activity type m . The parameter q_m is a quota of how many activities of type m need to be visited. The *randperm* term in step 4 refers to random sequence generation. The *LP* term in step 8 refers to solving the linear program for

arrival times T under goal arrival constraints, travel times, and fixed route sequence X . This overcomes the objectives that are not additive across links. The *penalty* term in step 9 stores the penalty for exceeding the hard time windows, as activities can have both goal arrivals and hard time windows.

Although this algorithm is designed for the G-SHARP model in [Chow \(2014\)](#), it can be readily adapted to handle the mHAPP model and its selective variant as well. The objective needs to be modified and the LP also needs to be modified to include the additional constraints for the transport nodes and the continuous parking duration variables Q . The random selection must include the transport nodes.

The algorithm has been tested in the following computational example. For a single user household lives in zone 81 of a 100-zone grid shown. Based on the parameters for the population and for the household, its activity schedule is obtained with [Algorithm 4.2](#) with $G=40$. Schedules for two scenarios are solved for: a base scenario (A) and a scenario (B) where the utility of discretionary activities is doubled. These are shown in [Fig. 4.6](#).

In Scenario A, the user chooses to work (in the steady state) at zone 16. The user goes to zone 28 for lunch at 12:31 p.m. before heading back to work at 1:36 p.m. The lunch arrival time is earlier than her desired time of 12:36 p.m. due to scheduling conflicts. She returns home by 6:11 p.m. without visiting any discretionary activity. In Scenario B, the increased utility for discretionary activities affords the user a visit to zone 36 for a discretionary activity after work.

4.4 MARKET SCHEDULE EQUILIBRIUM FOR A TRANSPORT SYSTEM

So far in this chapter, the modeling of activity schedules focuses on individual decision-making. When considering the interaction of a market of users with the mobility providers, it is necessary to evaluate the market schedule equilibrium. We define market schedule equilibrium in this context as follows.

Definition 4.3 *The **market schedule equilibrium** for a multimodal transport system is a steady state in which the market aggregation of constrained choices, $\mathcal{P}[i|C]$, for a schedule i among schedule choice set C does not exceed link load capacities in the system, $\sum_{i \in J} \delta_{i,uv} \mathcal{P}[i|C] \leq k_{uv}$, where $\delta_{i,uv}$ is a variable indicating schedule i contributes to the maximum load on link (u,w) associated with facilities $u \in \{O, D, \Lambda\}$, $w \in \{O, D, \Lambda\}$, and k_{uv} is the link capacity.*

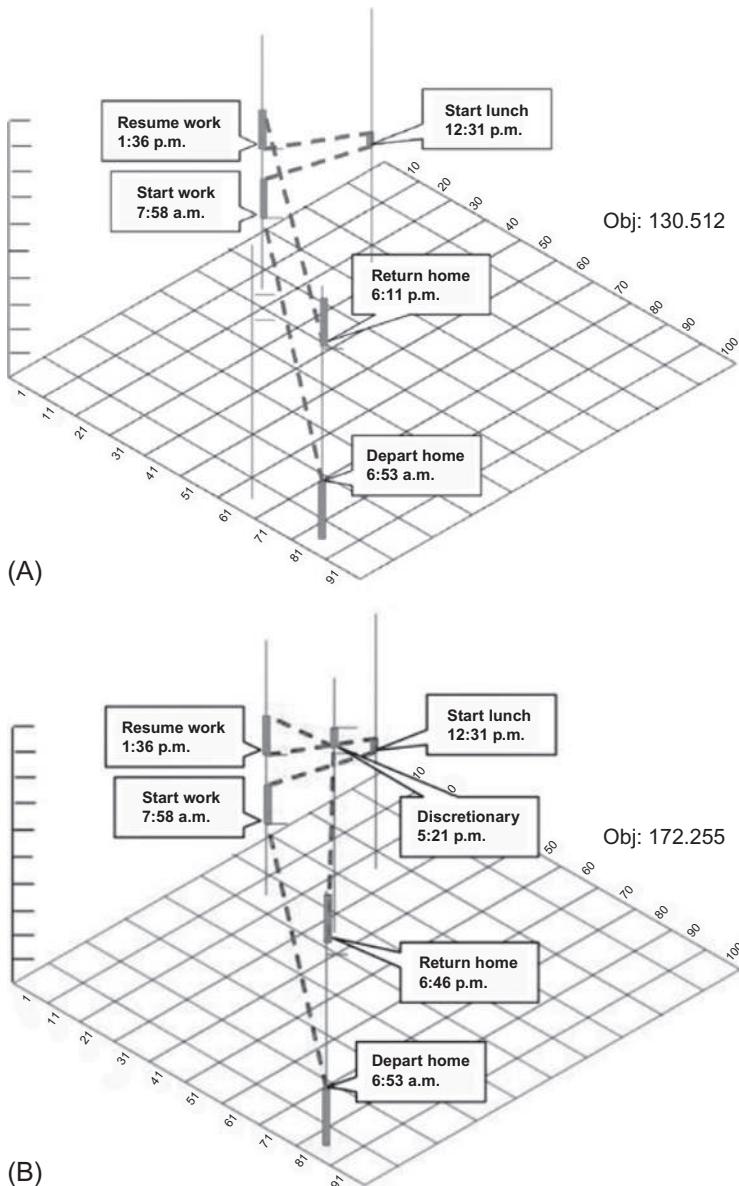


Fig. 4.6 Solutions of example in [Chow \(2014\)](#) for a single household using [Algorithm 4.2](#), for (A) a baseline scenario and (B) a scenario where discretionary activity utility is doubled. (Source: [Chow, 2014](#).)

Two challenges need to be addressed to obtain the market schedule equilibrium. First, a method is needed to aggregate individual schedule choices into market flows. Second, the outcome of these flows needs to reflect the effects of binding capacities in the time-space network. A method developed by [Chow and Djavadian \(2015\)](#) is described in [Section 4.4.1](#) to address these two challenges. Lastly, the practical application of these concepts is captured by MATSim's use of a day-to-day adjustment process (see [Djavadian and Chow, 2017b](#)) to model equilibration of scheduling decisions (through a genetic algorithm rescheduling procedure) and within-day traffic dynamics using cellular automata. An illustration of the tool in evaluating a change in headway for a transit line in New York City is shown in [Section 4.4.2](#).

4.4.1 The Aggregation Problem

The choice of each user in the market is dependent on network attributes such as travel time, travel cost, schedule delay, and so on, as shown earlier in Eqs. (4.2)–(4.15). In theory, the utility maximization in the HAPP model is equivalent to the utility maximization in a schedule choice model ([Recker, 2001](#)). Indeed, the pickup and delivery problem itself can be formulated as a set covering problem in which the master problem is to choose a schedule from a set of implicitly enumerated schedules ([Dumas et al., 1991; Kang and Recker, 2013](#)), which further reinforces this connection. The market schedule equilibrium problem can be decomposed into sampling-based mHAPP subproblems and a restricted master problem as illustrated in [Fig. 4.7](#). The components are explained in more detail later.

In the mHAPP models, users change schedules when the weights of their objectives, β , and arrival time constraints, g , are varied. As a result, the schedule choice problem for a population cannot be modeled with utility functions with fixed objective coefficients and arrival time constraints across the population and only random unobserved disturbances like in multinomial logit models. Two different users are distinguished by different values of β and g , which means a schedule model needs to exhibit random coefficients across the population. This can be addressed using a mixed logit model structure ([Hensher and Greene, 2003](#)). In the mixed logit model, the utility is defined by Eq. (4.16).

$$U_{ji} = \beta_i' Z_{ji} + \epsilon_{ji} \quad (4.16)$$

where U_{ji} is the utility of user $i \in \mathbb{P}$ choosing schedule $j \in C_i$, C_i is the schedule choice set available to user $i \in \mathbb{P}$, $Z_{ji}[X, T, Y, Q]$ is a vector of objective values

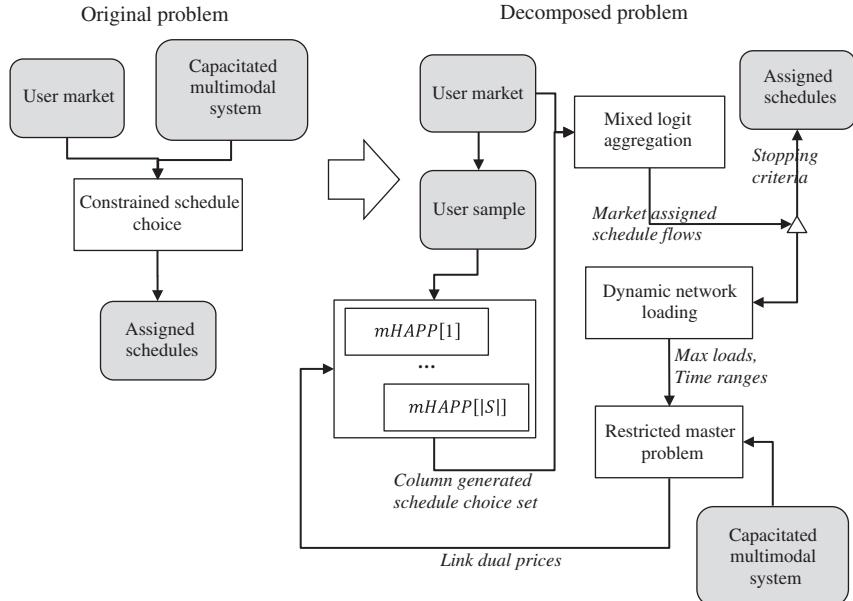


Fig. 4.7 Overview of decomposition approach to obtaining market schedule equilibrium.

described in Eq. (4.2) pertaining to a specific schedule j , β is a vector of normally distributed variates pertaining to the objectives, and ε_j is a Gumbel variate.

The probability of a user i choosing an alternative j is determined by Eq. (4.17).

$$P_i(j \mid C_i) = \int_s P_i(j \mid C_i, \beta_s) ds \quad (4.17)$$

where $P_i(j \mid C_i, \beta_s) = \frac{\exp(\beta_s' Z_{ji})}{\sum_l \exp(\beta_s' Z_{li})}$. The integral in Eq. (4.17) can be

numerically approximated by sampling values of β_s for a set S , $s \in S$, and taking the sample average of the simulated probability shown in Eq. (4.18). Normally for K -dimensional vectors of β_s with covariance matrix, sampling can be accomplished using Cholesky decomposition (see Train, 2009, and Appendix D). In the case of mHAPP models, the parameters of each agent's utility function are estimated using inverse optimization, which is discussed in Chapter 5. Since we have estimated values for each agent, they can be used directly as the sample draws without having to newly sample using the Cholesky decomposition method.

$$P_i(j \mid C_i) \approx \frac{1}{|S|} \sum_{s \in S} P_i(j \mid C_i, \beta_s, X_{ji}) \quad (4.18)$$

This model approach is illustrated in a generic example in [Exercise 4.3](#).

Exercise 4.3

Consider a mode choice model in which the utility for mode j for user i is defined as $U_{ji} = \theta(\beta_{cost} \times Cost_j + \beta_{time} Time_j) + \epsilon_{ji}$, where ϵ_{ji} is Gumbel distributed, $\theta = 0.2$, and $\beta_{cost} \sim N[-3, 0.2^2]$ and $\beta_{time} \sim N[-1, 0.1^2]$, where β_{cost} and β_{time} are independent from each other. Compare the average probability of choosing alternative A (with 95% confidence interval) for this model against an MNL model among the choices:

Alternative A: cost = \$5, time = 30 min

Alternative B: cost = \$15, time = 15 min

Based on a simulation of 500 independent samples, the average probability of choosing alternative A lies between [94.1%, 94.6%] with 95% confidence. An MNL model based on average values of the coefficients has the following probability:

$$P_i(A) = \frac{\exp(0.2(-3 \times 5 - 1 \times 30))}{\exp(0.2(-3 \times 5 - 1 \times 30)) + \exp(0.2(-3 \times 15 - 1 \times 15))} = 95.3\%$$

Using the mixed logit modeling framework, the market demand for a schedule is $\mathcal{P}[i| C] = P_i[j| C_i]|\mathbb{P}|$. Link load capacity (not to be confused with flow capacity) constraints are based on the maximum load of a link facility (u, w) , which is derived according to dynamic network loading. This is determined by first defining the start and end times at which the maximum link load occurs: φ_{uw}^a and φ_{uw}^b . These times are dependent on the aggregated population flows assigned to each schedule according to cumulative arrivals and departures of all scheduled populations. For example, arrival of flows assigned to schedule j occurs at time T_{ju} (the arrival time of schedule j at node u). Departure occurs at time $T_{ju} + t_{uw}$. Based on these cumulative arrivals and departures, the start of the time range where the load is maximum is denoted φ_{uw}^a , and the end is φ_{uw}^b . Once this range $[\varphi_{uw}^a, \varphi_{uw}^b]$ is known, a dependent indicator function $\delta_{uwj}[\varphi_{uw}^a, \varphi_{uw}^b]$ (mentioned in [Definition 4.3](#)) is defined to equal 1 if a schedule j falls within the time range for link (u, w) .

Definition 4.4 The **indicator function** $\delta_{uwj}[\varphi_{uw}^a, \varphi_{uw}^b]$ $\delta_{uwj}=1$ if $T_{ju} + t_{uw} < \varphi_{uw}^b$ and $T_{ju} > \varphi_{uw}^a$.

This concept is illustrated in [Exercise 4.4](#).

Exercise 4.4

Three schedules make use of a link (u, w) with travel time $t_{uw} = 30 \text{ min}$: $j=1$ with flow of 300 arriving at time $T_{1u}=8 \text{ a.m.}$, $j=2$ with flow of 400 arriving at time $T_{2u}=8:15 \text{ a.m.}$, and $j=3$ with flow of 500 arriving at time $T_{3u}=8:30 \text{ a.m.}$. Determine the maximum load on this link, the time range of max load, and indicate which of the three schedules contribute to that max load.

Noting that each set of flows that arrive stay on the link for 30 min due to the travel time, a cumulative diagram can graphically show the period of maximum load. This is presented in [Table 4.2](#) and [Fig. 4.8](#).

Table 4.2 Cumulative arrivals and departures on link (u, w) for [Exercise 4.4](#)

Time	Arrival load	Departure load	Load
7:45 a.m.	0	0	0
8:00 a.m.	0	0	0
8:00 a.m.	300	0	300
8:15 a.m.	300	0	300
8:15 a.m.	700	0	700
8:30 a.m.	700	0	700
8:30 a.m.	1200	300	900
8:45 a.m.	1200	300	900
8:45 a.m.	1200	700	500
9:00 a.m.	1200	700	500
9:00 a.m.	1200	1200	0
9:15 a.m.	1200	1200	0

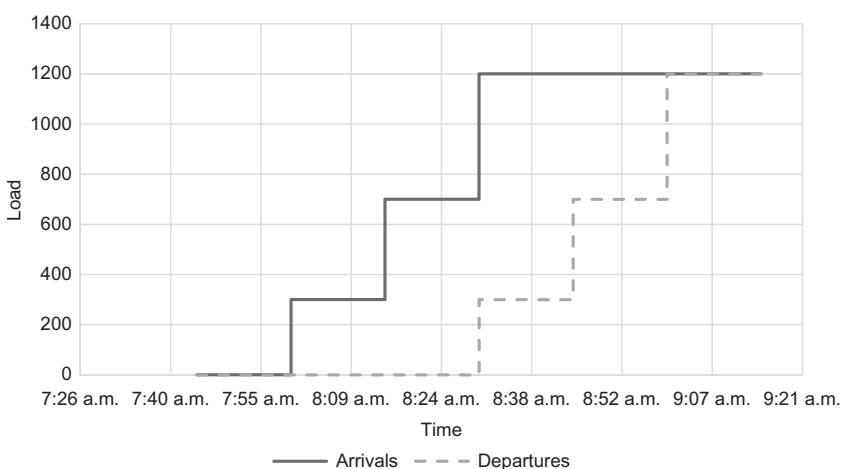


Fig. 4.8 Cumulative diagram of arrivals and departures onto link (u, w) for [Exercise 4.4](#).

The maximum load is 900 and occurs between 8:30 a.m. and 8:45 a.m. This means we set $\varphi_{uw}^a = 8:30$ a. m. and $\varphi_{uw}^b = 8:45$ a. m. For this range, only schedules $j = \{2, 3\}$ contribute to the load. Therefore $\delta_{uw1} = 0$, $\delta_{uw2} = 1$, $\delta_{uw3} = 1$.

The indicator function is used to determine the feasibility of a schedule assignment for a population. The constraint $\delta_{i,uw}\mathcal{P}[i|C] \leq k_{uw}$ has a corresponding capacity dual price λ_{uw} where the following complementary slackness condition exists: if $\delta_{i,uw}\mathcal{P}[i|C] < k_{uw}$, then $\lambda_{uw} = 0$, and if $\lambda_{uw} > 0$, then $\delta_{i,uw}\mathcal{P}[i|C] = k_{uw}$.

Keeping λ_{uw} , φ_{uw}^a , φ_{uw}^b fixed, the effect of binding capacity is used to solve the set of mHAPP subproblems for each sampled individual. For these cases, the travel cost objective $\sum_{(u,w)} c_{uw} X_{uw}$ in Eq. (4.2) is replaced with $\sum_{(u,w)} c'_{uw} X_{uw}$, where $\sum_{(u,w)} c'_{uw} X_{uw} = \sum_{(u,w)} c_{uw} X_{uw} + \sum_{(u,w)} \lambda_{uw} Y_{uw}^c$. The binary variable Y_{uw}^c is used to apply the dual price for a link if the mHAPP solution traverses link (u,w) during the time interval $[\varphi_{uw}^a, \varphi_{uw}^b]$. This is accomplished by adding the following constraints in Eq. (4.19)–(4.22) to each mHAPP model.

$$T_u^{rs} \leq \varphi_{uw}^a - c_{uw} + \mathcal{M} Y_{uw}^a + (1 - X_{uw}^{rs})\mathcal{M}, \quad \forall (r,s), (u,w) \quad (4.19)$$

$$T_u^{rs} + (1 - X_{uw}^{rs})\mathcal{M} \geq \varphi_{uw}^b - \mathcal{M} Y_{uw}^b, \quad \forall (r,s), (u,w) \quad (4.20)$$

$$Y_{uw}^c + 1 \geq Y_{uw}^a + Y_{uw}^b, \quad \forall (u,w) \quad (4.21)$$

$$Y_{uw}^a, Y_{uw}^b, Y_{uw}^c \in \{0, 1\}, \quad \forall (u,w) \quad (4.22)$$

Finally, because the indicator variable $\delta_{i,uw}$ is endogenous to the market schedule equilibrium problem, convergence cannot be guaranteed by simply iterating the procedure shown in Fig. 4.7. Instead, a fixed point algorithm is needed to assure a fixed schedule choice set, assigned choices, time ranges for maximum loads, and link capacity dual prices. We adopt a Method of Successive Averages for convenience, although more efficient algorithms can also be used as discussed in Chapter 3. Algorithm 4.3 is shown as follows and illustrated in Exercise 4.5 from Chow and Djavadian (2015).

Algorithm 4.3 (Chow and Djavadian, 2015). Decomposition Algorithm for Mixed Logit Market Schedule Equilibrium With Sampled mHAPP Choice Set Generation

Input: calibrated mHAPP models (\mathcal{H}_i) for $i \in S$, population activity demand characteristics W

1. Initialize with $n=1$, restricted choice set $C'_{in} = \{\operatorname{argmin}_{\lambda_{uv}} \mathcal{H}_i[\lambda_{uv} = 0]\}_{i \in S}$.
2. Aggregate using Eq. (4.18) to obtain market schedule assignment f_{ji}^n , perform dynamic loading, and update time range $[\varphi_{uv}^{an}, \varphi_{uv}^{bn}]$ for each link with binding capacity.
3. Solve restricted master problem, $\max_{\lambda} \left(\sum_{(u, v)} \lambda_{uv} (\mathcal{P}_{uv}[C'] - k_{uv}) \right)$ to update link capacity dual prices λ_{uv} .
4. Update $C'_{i,n+1} = C'_{in} \cup \{\operatorname{argmin}_{\lambda_{uv}} \mathcal{H}_i[\lambda_{uv} = 0]\}_{i \in S}$.
5. Determine auxiliary primal population flows ζ_{ji} .
 - a. Identify all schedules impacted by the updated time range of binding capacity, $\delta_{uvj}[\varphi_{uv}^{an}, \varphi_{uv}^{bn}]$.
 - b. Fit the flows of the impacted schedules of individuals such that the total flow is equal or less than the link capacities, using the mixed logit distribution in Eq. (4.18).
 - c. Fit the remaining flows according to the mixed logit distribution among the remaining schedules not impacted by the capacity effect, if any.
6. Perform MSA update of the schedule assignments: $f_{ji}^{n+1} = \frac{1}{n+1} \zeta_{ji} + \frac{n}{n+1} f_{ji}^n$.
7. If MSA stopping criteria reached, stop, else let $n=n+1$ and go to step 2.

Output: implicitly enumerated schedule choice set C'_i , assignment onto these schedules, f_{ji} , and link capacity dual prices λ_{uv} .

Exercise 4.5

(Chow and Djavadian, 2015). Obtain the market schedule equilibrium for the following system with population activity scheduling. An activity and transport system is shown in Fig. 4.9, Table 4.3, and Table 4.4 representing a simplified view of 1000 individuals living in suburban residences (zone 0) accessing a central business district for work (zone 1). There are four transport nodes: node 6 and node 7 are parking facilities; node 8 and node 9 are transit stations. Transit line headways are set at 10 min; parking facilities charge \$4 plus \$0.01 per minute. There are three classes of users: class 1 (250 people) has objective coefficients $\beta_1 = \{3, 1, 3, 0, 0, 0.8, 0.2\}$ corresponding to Eqs. (4.2b)–(4.2g) (note that Eq. 4.2e has 2 terms, first for automobile mode and second for transit mode, both set to 0), goal arrival for activity 1 is $g_{11}=540$ min, and for activity 2 it is $g_{12}=720$ min. For class 2 (250 people), the objective coefficients are the same $\beta_2=\beta_1$, but goal arrival to activity 1 is $g_{21}=510$ min. Class 3 (500 people) has $\beta_3=\{3, 1, 3, 0, 200, 0.8, 0.2\}$ (which implies a high preference for transit) and the same goal arrival times as class 1. Link (0, 6) has a load capacity of 200 users.

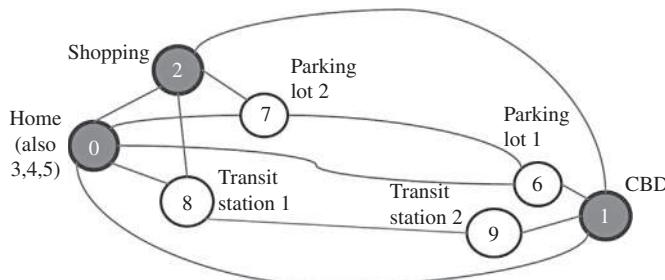


Fig. 4.9 Activity and transport system for [Exercise 4.5](#) (driving: {(0,7),(0,6),(6,7)}, transit: (8,9), walking: {(0,2),(0,8),(2,8),(2,7),(1,6),(1,9)}}). (*Source: Chow and Djavadian, 2015.*)

Table 4.3 Activity data for [Exercise 4.5](#)

Activity node	Duration (min)	Time to open (min)	Time to close (min)	Early arrival penalty	Late arrival penalty
1	480	360	720	0.4	2.4
2	30	600	1230	0	0

Table 4.4 mHAPP network data for [Exercise 4.5](#)

Link ID	r	s	u	w	Cost	Time	Link ID	r	s	u	w	Cost	Time
1	0	1	0	1	0	500	28	1	3	1	6	0	7
2	0	2	0	2	0	200	29	1	3	6	3	7.2	45
3	1	2	1	2	0	350	30	1	3	1	9	0	12
4	1	3	1	3	0	500	31	1	3	9	8	3	50
5	1	4	1	4	0	500	32	1	3	8	3	0	6
6	2	1	2	1	0	350	33	1	4	1	6	0	7
7	2	3	2	3	0	200	34	1	4	6	4	7.2	45
8	2	4	2	4	0	200	35	1	4	1	9	0	12
9	3	2	3	2	0	200	36	1	4	9	8	3	50
10	3	4	3	4	0	0	37	1	4	8	4	0	6
11	3	5	3	5	0	0	38	2	1	2	7	0	5
12	4	1	4	1	0	500	39	2	1	7	6	4.8	30
13	4	3	4	3	0	0	40	2	1	6	1	0	7
14	4	5	4	5	0	0	41	2	1	2	8	0	10
15	0	1	0	6	7.2	45	42	2	1	8	9	3	50
16	0	1	6	1	0	7	43	2	1	9	1	0	12
17	0	1	0	8	0	0	44	2	3	2	7	0	5
18	0	1	8	9	3	50	45	2	3	7	3	3.2	20
19	0	1	9	1	0	12	46	2	4	2	7	0	5
20	0	2	0	7	3.2	20	47	2	4	7	4	3.2	20
21	0	2	7	2	0	5	48	3	2	3	7	3.2	20
22	1	2	1	6	0	7	49	3	2	7	2	0	5

Table 4.4 mHAPP network data for [Exercise 4.5](#)—contd

Link ID	<i>r</i>	<i>s</i>	<i>u</i>	<i>w</i>	Cost	Time	Link ID	<i>r</i>	<i>s</i>	<i>u</i>	<i>w</i>	Cost	Time
23	1	2	6	7	4.8	30	50	4	1	4	6	7.2	45
21	1	2	7	2	0	5	51	4	1	6	1	0	7
25	1	2	1	9	0	12	52	4	1	4	8	0	6
26	1	2	9	8	3	50	53	4	1	8	9	3	50
27	1	2	8	2	0	10	54	4	1	9	1	0	12

Initially solving the mHAPP model for the three classes of users without consideration of capacity, we get the following routes (arrival time) sequences:

Class 1: 0 (488) – 6 (533) – 1 (540) – 6 (1027) – 7 (1057) – 2 (1062) – 7 (1097) – 4 – 3 – 5 (1117)

Class 2: 0 (458) – 6 (503) – 1 (510) – 6 (997) – 7 (1027) – 2 (1032) – 7 (1067) – 4 – 3 – 5 (1087)

Class 3: 0 (462) – 8 (468) – 9 (523) – 1 (540) – 9 (1032) – 8 (1087) – 3 (1098) – 7 (1118) – 2 (1123) – 7 (1158) – 4 – 5 (1178)

Based on this assignment, however, the dynamic network loading shows that link (0, 6) load capacity is violated because a maximum load of 500 users occurs between 8:08 a.m. and 8:23 a.m. Based on [Algorithm 4.3](#) with a stopping tolerance of 0.001, it converges after 79 iterations as shown in [Fig. 4.10](#). Note that this objective value is the capacity-relaxed lower bound.

The final equilibrium market flows based on mixed logit assignment are presented in [Table 4.5](#).

The presence of the load capacity redistributes the travel costs of the population in this example. The total disutility of the market schedule equilibrium increases from 671,010 (when uncapacitated) to 675,455, which is equivalent to 1350.91 disutility per user. The cumulative diagram for link (0, 6) is shown in [Fig. 4.11](#).

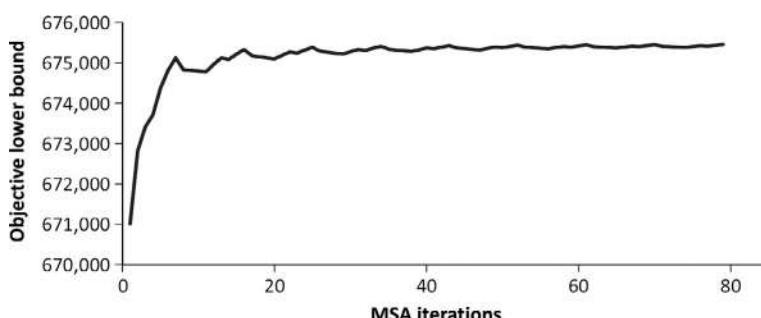


Fig. 4.10 Algorithm 4.3 convergence for [Exercise 4.5](#). (Source: *Chow and Djavadian, 2015*.)

Table 4.5 Market schedule equilibrium assignment

Class	Schedule	Route	Arrival times (min)	Flows
1	1 (initial)	0-6-1-6-7- 2-7-4-3-5	488-533-540-1027-1057- 1062-1097-1117	102.137
	2	0-6-1-6-7- 2-7-4-3-5	413-533-540-1027-1057- 1062-1097-1117	34.199
	3	0-6-1-6-7- 2-7-3-4-5	443-533-540-1027-1057- 1062-1097-1117	32.300
	4	0-8-9-1-9- 8-3-7-2- 7-4-5	462-468-523-540-1032- 1087-1098-1118-1123- 1158-1178	66.484
	5	0-6-1-6-7- 2-7-3-4-5	368-533-540-1027-1057- 1062-1097-1117	14.880
	1 (initial)	0-6-1-6-7- 2-7-4-3-5	458-503-510-997-1027- 1032-1067-1087	97.803
2	2	0-6-1-6-7- 2-7-3-4-5	413-503-510-997-1027- 1032-1067-1087	66.494
	3	0-6-1-6-7- 2-7-3-4-5	368-503-510-997-1027- 1032-1067-1087	66.463
	4	0-6-1-6-7- 2-7-3-4-5	443-503-510-997-1027- 1032-1067-1087	19.239
	1 (initial)	0-8-9-1-9- 8-3-7-2- 7-4-5	462-468-523-540-1032- 1087-1098-1118-1123- 1158-1178	500

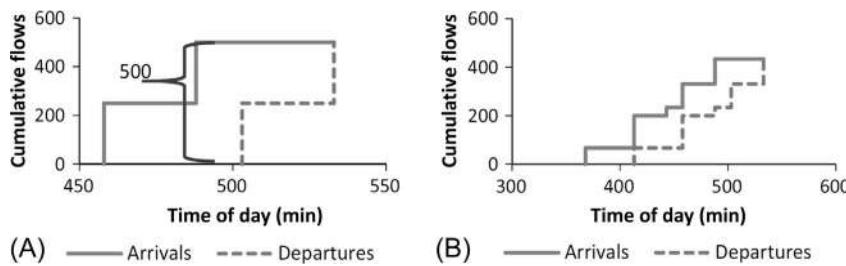


Fig. 4.11 Cumulative diagram (A) without link capacity and (B) with link capacity.
(Source: [Chow and Djavadian, 2015](#).)

4.4.2 Integration With MATSim

The prior sections provide a theory of heterogeneous market activity scheduling behavior in response to different multimodal systems designs. Activity scheduling modeling of market schedule equilibrium is available in practice

using tools like MATSim. MATSim uses agent simulation of day-to-day adjustments to capture the equilibrium interaction (shown by Djavadian and Chow, 2017b, to be an agent-based stochastic user equilibrium) between a system design and market activity scheduling. The drawback is that sensitivity analyses of the market schedule equilibrium performance measures with respect to different system or agent parameters cannot be quantified analytically.

MATSim has been used in recent research studies to evaluate the market schedule equilibrium for different MaaS technologies. Neumann and Nagel (2013) evaluated the effect of changes in public transit service frequencies on user activity scheduling. Boesch et al. (2016), Bischoff and Maciejewski (2016), and Hörl et al. (2017) used MATSim to evaluate autonomous taxis. Balac et al. (2017) modeled the effect of parking pricing policies on a free-floating car-sharing system in Zurich. The tool is illustrated in [Exercise 4.6](#).

Exercise 4.6

Use MATSim to evaluate the effect of a 50% headway reduction for the downtown “A” line to Lefferts Blvd on riders in New York using input data files located on GitHub <https://github.com/BUILTNYU/Elsevierbook/tree/master/Data%20Files>.

The data on GitHub includes the following files:

- *finalnetwork.xml*: This consists of the road network for the five boroughs in New York, obtained from OpenStreetMap.
- *config.xml*: The config file provides the basic parameters of the scenario used to run the simulation. 10 days are simulated.
- *population3.xml*: The population file contains the travel plans for each agent and includes activity types, locations, and end times. We generated this file from NYMTC’s 2010/2011 Household Travel Survey by filtering the trips that started and ended in one of the five boroughs, leading to 8487 agents in total.
- *vehicle.xml*: The vehicle file is also generated from the historical GTFS data and used along with the schedule file. It can be used to set the vehicle capacities, for example. In this example, we did not modify the capacity, so it remains at 50 persons per vehicle (although trains should have much higher capacity).
- *finalschedule.xml*: This includes the subway schedule data obtained from GTFS. It includes transit stops, routes, and departure times of each subway line in both weekday and weekend, with a total of 26 lines.
- *finalschedule-half.xml*: This is the modified schedule based on half the headway of the original schedule for line A to Lefferts Blvd.

The network and population data are visualized in [Fig. 4.12](#).

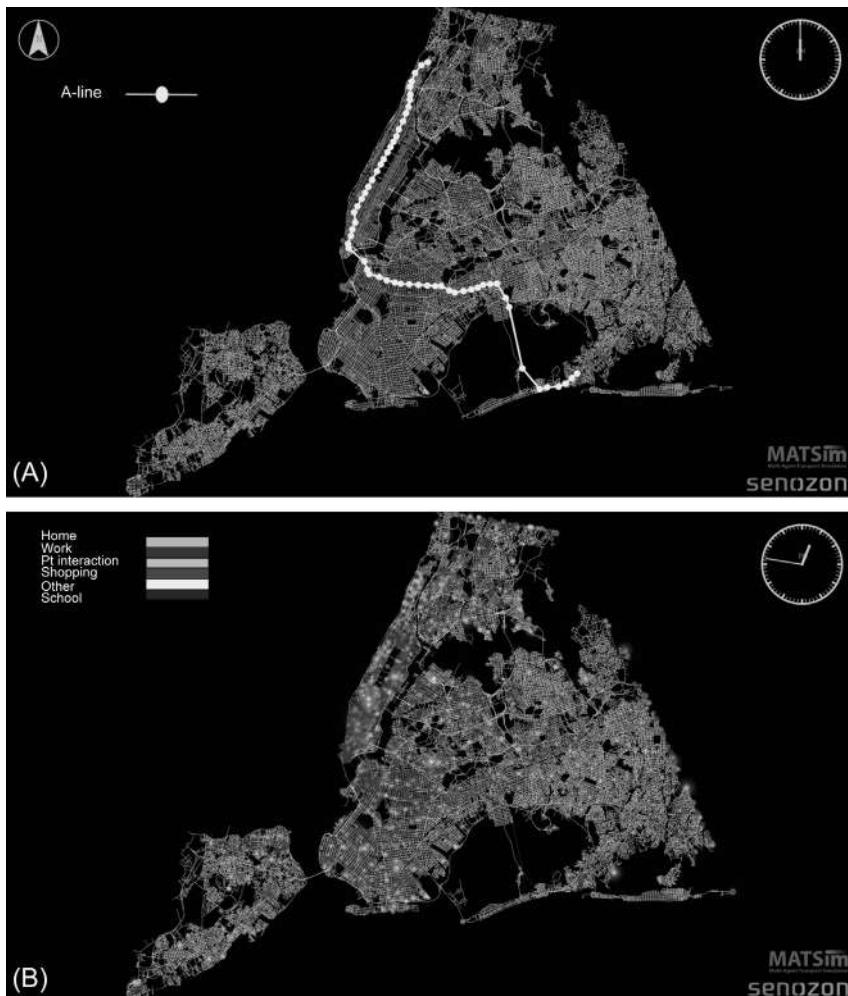


Fig. 4.12 MATSim input data: (A) transit line schedule from GTFS overlaid on a road network from OpenStreetMaps and (B) activity zones corresponding to NYMTC household travel survey.

Two scenarios need to be run, taking an average of 16 min on an Intel Core i7-6700 CPU with 3.40 GHz running on a 64-bit Windows 10 operating system with 16 GB RAM to run 10 days of iterations. In the first scenario, the following script is run in MATSim shown in Fig. 4.13. For the second scenario, the “finalschedule.xml” is replaced with the “finalschedule-half.xml.”

The schedule file includes all the subway lines in NYC. The A line to Lefferts Boulevard is examined. The line has 30 stations as presented in Table 4.6.

```

<module name="plans" >
  <param name="inputPlansFile" value="population3.xml" />
</module>
<module name="network" >
  <param name="inputNetworkFile" value="finalnetwork.xml" />
</module>
<module name="transit">
  <param name="transitScheduleFile" value="finalschedule.xml" />
  <param name="vehiclesFile" value="vehicle.xml" />
  <param name="transitModes" value="pt" />
  <param name="useTransit" value="true" />           Change to "finalschedule-half.xml"
</module>

```

Fig. 4.13 Schedule file that needs to be modified for scenario analysis.

Table 4.6 Stations on the A-Lefferts Blvd line

ID	Station name	ID	Station name	ID	Station name
1	Inwood—207 St	11	34 St—Penn Station	21	Utica Av
2	Dyckman St	12	14 St	22	Broadway Jct
3	190 St	13	W 4 St	23	Euclid Av
4	181 St	14	Canal St	24	Grant Av
5	175 St	15	Chambers St	25	80 St
6	168 St	16	Fulton St	26	88 St
7	145 St	17	High St	27	Rockaway Blvd
8	125 St	18	Jay St—MetroTech	28	104 St
9	59 St— Columbus Circle	19	Hoyt—Schermerhorn Sts	29	111 St
10	42 St—Port Authority Bus Terminal	20	Nostrand Av	30	Ozone Park— Lefferts Blvd

Without changing the default parameters in MATSim, the social welfare values before and after the headway reduction are shown in Fig. 4.14. MATSim generates multiple plans per agent and assigns them stochastically to each plan via a discrete choice model. The “EXECUTED” trajectories in Fig. 4.14 indicate the average score overall in all the plans available to each agent, whereas “BEST” trajectories indicate the maximum utility plans. Ten days are simulated for each scenario. The headway reduction for the one downtown A line to Lefferts Boulevard has minimal impact on the social welfare of the population.

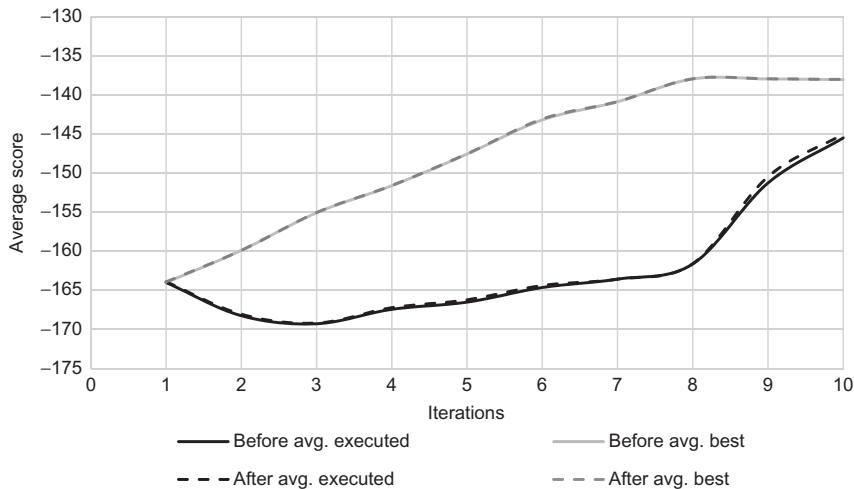


Fig. 4.14 Comparison of distribution of welfare measure for users of the system before and after headway reduction.

Furthermore, intuition suggests that reducing headway should improve level of service by reducing average wait time, leading to an increase of ridership on the altered line. This is not the case here, however. A plot of the ridership from 6:59 a.m. to 10:02 p.m. is obtained before and after the headway reduction shown in Fig. 4.15. It shows the number of passengers onboard the line when measured after departing each station. The multiagent simulation can reveal an impact that is not entirely trivial

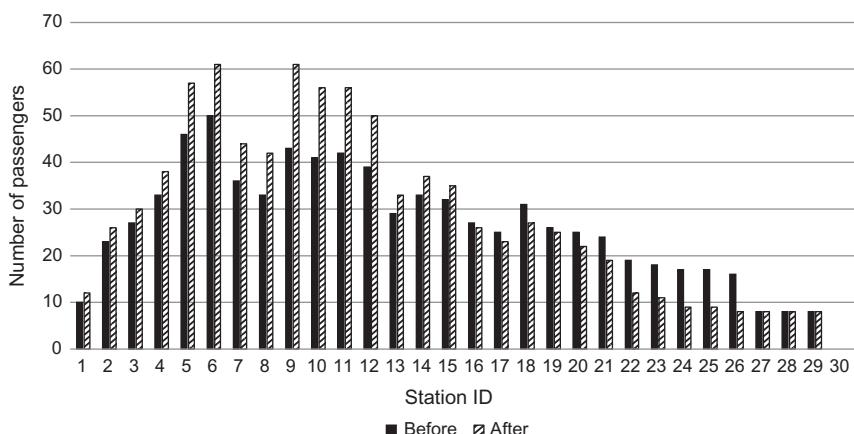


Fig. 4.15 Comparison of distribution of 7 a.m.–10 p.m. ridership on the A line with Regular (before) and Reduced (after) headway.

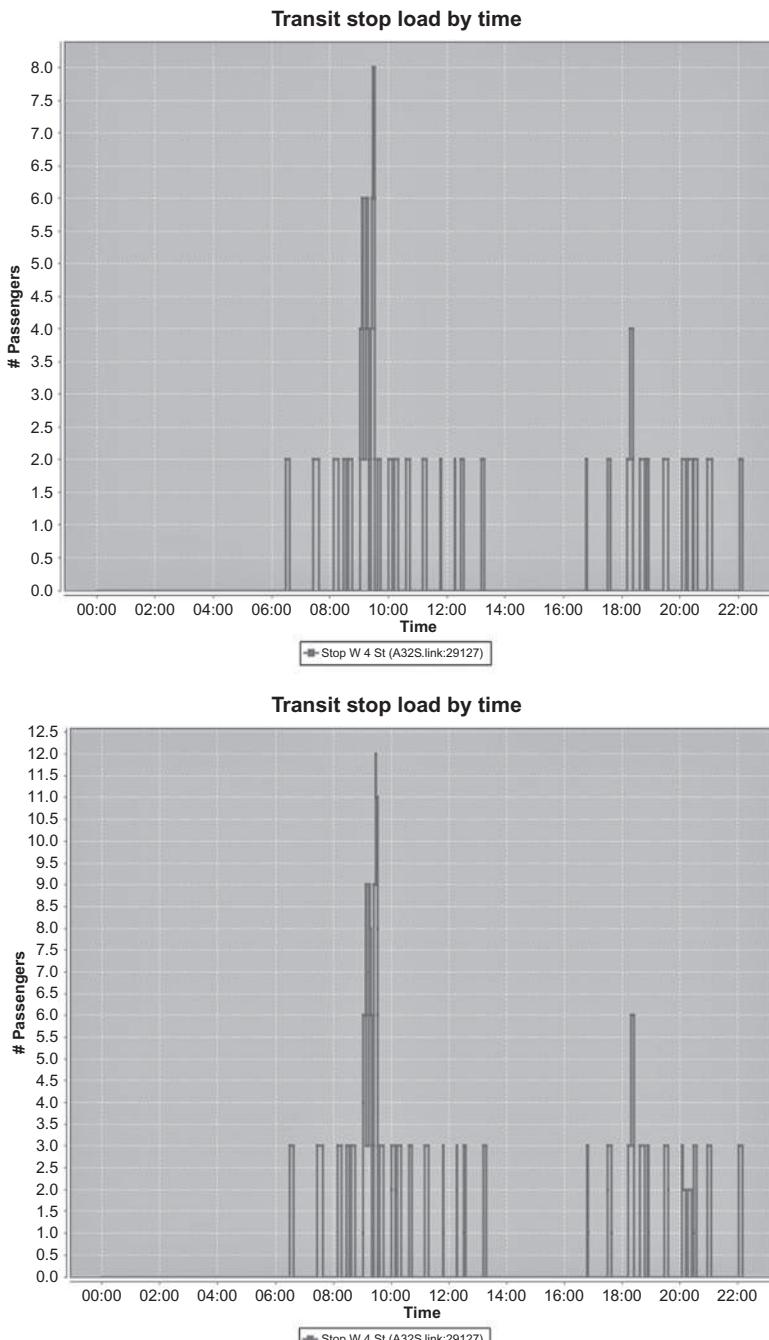


Fig. 4.16 Comparison of distribution of time of day arrivals onto the A line at the W. 4th Station (Station 13) before and after headway reduction.

to predict. While stations 2–15 have an increase in ridership, stations 16–30 see a reduction in ridership.

A deeper analysis of the results suggests the A line shares some of the same routes as other subway lines particularly around the W. 4th Station (station 13). The result is that reducing headway can cause the subways to overcrowd, leading to more delays in some sections. The additional delays cause the ridership demand to decrease. Such a result would not be anticipated using a simple analytical model.

Lastly, we can observe a change in the distribution of the passenger load on the line by time of day, made possible because of the passenger scheduling consideration in the simulation. This is shown in [Fig. 4.16](#).

MATSim and mHAPP share some commonalities: they are both based on utility maximization and they both have GA-based solution algorithms to find schedules. The mHAPP modeling framework represents the analytical approach while MATSim represents the simulation-based approach to assigning market schedules onto a transport system. Despite the differences in implementation, there are advantages to integrate both models together. Several use cases are discussed here.

Use Cases

- (1) **Parameter estimation:** MATSim is based on having a synthetic population with activity schedule preferences. To estimate the parameters for MATSim activity scheduling, mHAPP models for a sample of users can be calibrated as discussed in [Chapter 5](#) and used to then create a consistent synthetic population.
- (2) **Generating initial schedules for new scenarios:** While MATSim can use the surveyed schedules from the data as a starting point for the GA to analyze local perturbations in the system, it may not be able to identify good initial solutions for new scenarios in which data is unavailable. For example, having an autonomous vehicle ridesharing feeder for last mile service may result in significant changes in schedule that are not apparent (e.g., leaving the car at home, visiting different destinations, changing trip chain patterns) if applying a GA from the base scenario schedules. In those cases, mHAPP can be used to optimize schedules for a user sample that take system capacities into account. These schedules can serve as the initial population for the GA to cross-breed to obtain better solutions.

- (3) **Sampling elasticity:** One of the drawbacks of an agent-based simulation is that elasticities cannot be analytically derived. In this case, mHAPP model framework can be used to fit to a sample of users to obtain sampled elasticities for those users, while relying on MATSim to provide population-level metrics.
- (4) **Sampling suboptimality of MATSim solutions:** Another drawback of MATSim is the reliance on a heuristic (GA) to obtain new schedules for the population. A modeler cannot tell how suboptimal is a solution. The mHAPP model can be used in parallel to solve the schedules for a user sample to compare to the GA solution of the same sample of users. That way, it is possible to approximate the suboptimality of the population-level solutions.

The benefits of integration go both ways. The mHAPP framework can benefit from the MATSim outputs by using the simulated travel times between different activity nodes to capture congestion effects. The use cases presented here have not yet been tested and remain open research items.

4.5 URBAN FREIGHT ACTIVITY ANALYSIS

While the activity scheduling argument has been made primarily for passenger travel, it is even more important for evaluating urban freight delivery demand. Urban freight is important because it plays such a significant role in economic and environmental sustainability (Steenhof et al., 2006; Lee et al., 2009; Chow et al., 2010b). In many ways, the argument for evaluating scheduling response over congestion response is even more pertinent to urban freight because the trucks typically represent only a subset of the whole population, so they do not contribute as significantly to congestion effects of shared road space. In addition, their decisions tend to be at a tactical level for one part of a long chain of trips. As a result, their route level decisions based on congestions may not be very elastic. On the other hand, scheduling changes can contribute significantly to the decisions of the trucks, as illustrated empirically with off-hour delivery policy tests in New York (Holguín-Veras et al., 2011).

Urban public agencies have placed more emphasis on designing systems (e.g., Crainic et al., 2004) and policies to facilitate urban freight. Similar to MaaS, cities are seeing a growth in a number of urban delivery service strategies: truck on-street delivery policies (Amer and Chow, 2017), allocation of downtown hubs (Muñozuri et al., 2012), provision of real-time information systems (Taniguchi and Shimamoto, 2004), same day deliveries (Voccia

et al., 2017), crowdshipping (Miller et al., 2017), use of public lockers to reduce return trips (Morganti et al., 2014), drone deliveries (Murray and Chu, 2015), and more.

All these operations require changes in how urban space is shared with passenger mobility and have environmental and social impacts on the built environment. Policymakers therefore need to evaluate the effects of urban freight systems on users of those systems: shippers, carriers, receivers, and end users. A review of urban freight demand modeling is provided by Comi et al. (2012). Early models of urban freight were purely statistical truck count or truck trip forecasts (Chow et al., 2010b). However, urban freight deliveries are not simply OD trips like commuter trips. Instead, vehicles tend to conduct trip chains to drop off goods to multiple destinations (Langevin et al., 1996). Boerkamps et al. (2000) and Wisetjindawat et al. (2006) developed some of the first models that incorporate these more complex shipper behavior and spatial distribution patterns. Several studies have since proposed choice models for different components of the truck activity scheduling: Hunt and Stefan (2007), Russo and Comi (2010), and Ruan et al. (2012). More recent efforts use multiagent simulations to capture the complex and heterogeneous interactions between multiple agents: Holmgren et al. (2012), Van Heerden and Joubert (2014) (as an extension of MATSim).

The mHAPP framework is also applicable to urban freight in terms of evaluating market schedule equilibrium using analytical utility maximization modeling. Since the original HAPP model was derived from logistics routing discipline, application of mHAPP framework to urban freight is straightforward. An example is presented here from You et al. (2016) to illustrate its applicability.

A hypothetical freight activity system is shown in Fig. 4.17 with symmetric travel times for all OD pairs shown. Durations of activities (goods pickup, drop-off) at zone 2, 3, and 4 are 30 min. A market of 1000 single-truck firms serve this network, each belonging to one of two classes. Class 1 (600 firms) trucks are observed to take the following node sequence (and arrival time): 1(8:30 a. m.) – 2(9 a. m.) – 3(10 a. m.) – 1(11:30 a. m.). Class 2 (400 firms) trucks are observed to take: 1 (8:15 a. m.) – 2 (8:45 a. m.) – 4(10 a. m.) – 1(10:30 a. m.).

The truck firms are assumed to behave according to vehicle routing problems with the following objective function:

$$\max U = \sum_{u \in N} \beta_u \sum_{w \in N} X_{wu} - \beta_{tt}(T_D - T_O) - \sum_{u \in N} \beta_u^e P_u^e - \sum_{u \in N} \beta_u^l P_u^l$$

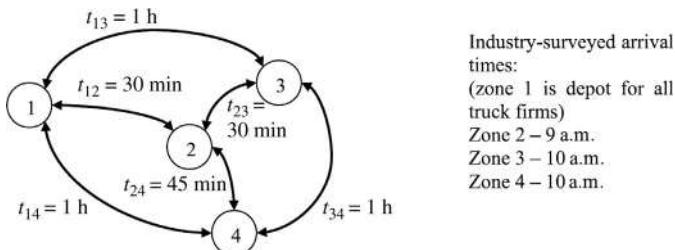


Fig. 4.17 Freight activity system. (Source: You et al., 2016.)

where the variables and parameters are defined earlier in the chapter. Both classes have the following homogeneous parameters: $g_2 = 9 \text{ a. m.}$, $g_3 = g_4 = 10 \text{ a. m.}$, $\beta_2^e = \beta_3^e = \beta_4^e = 1$, $\beta_2^l = \beta_3^l = \beta_4^l = 4$. The remaining parameters are calibrated to have minimum variance from a common prior, resulting in $\beta_2 = \beta_4 = 15$ for both classes. The trade-off is that trucks choosing zone 4 earn the additional utility of 15 but suffer from being 15 min early to the location and waiting.

Substitution effects can be measured just like random utility models. For example, visiting zone 2 and zone 4 is equivalent to arriving early by 15 min or late by 3.75 min.

Spatial-temporal scenarios can be analyzed. If link (2, 3) travel time increases from 30 to 45 min, the model predicts that 60% of the trucks would reschedule to the following sequence: 1(8:15 a.m.) – 2(8:45 a.m.) – 3(10 a.m.) – 1(11:30 a.m.).

RESEARCH AND DESIGN CHALLENGES

The following are open-ended challenges that sometimes refer to real data sets. Readers are encouraged to try to tackle the problems, and post their designs or solutions on the following GitHub page, <https://github.com/BUILTNYU/Elsevierbook/tree/master/Chapter%204>.

- (4.1) Turn on Location History on your Google Maps (or equivalent service) (<https://www.androidcentral.com/how-view-your-location-history-google-maps>). Track your movements for a week or longer. Based on the activity patterns on there, estimate models of departure time choice, “next destination” choice, and activity duration based on methods summarized in Table 4.1. How accurate is the model?

- (4.2) Write a program to construct the mHAPP model as a mixed integer linear program and test it on the different users in the example in [Exercise 4.1](#). Use a commercial solver of your choice (CPLEX, Gurobi, Julia, MATLAB, etc.).
- (4.3) For Challenge 4.2, write a branch and bound script in your language of choice to solve the mHAPP model instead of using a commercial solver. Compare objective values, solution variables, and computation times.
- (4.4) For Challenge 4.2, now implement the GA in [Algorithm 4.2](#). Compare objective values, solution variables, and computation times.
- (4.5) Using one of the programmed solution algorithms, revisit your collected data from Challenge 4.1. Try to see if you can calibrate the parameters of the mHAPP model such that your observed movements on a particular day would be optimal. Does the solution match?
- (4.6) Download the schedule of a bus route (<https://transitfeeds.com/>) and daily ridership (http://web.mta.info/nyct/facts/ridership/ridership_busMTA.htm) in NYC. Download the traffic analysis zones from which the route passes through. Using the 2010/2011 NYMTC Household Travel Survey (<https://www.nymtc.org/DATA-AND-MODELING/Travel-Surveys/2010-11-Travel-Survey>), create a stop-level OD matrix by time of day that fits the daily ridership. Knowing the bus capacity (or assuming a value of 75 passengers per bus), design a multimodal market schedule equilibrium model for this system and predict the change in arrival time distribution when:
 - a. Ridership demand is doubled;
 - b. Travel time increases by 50%.
- (4.7) Gomentum Station (<http://gomentumstation.net/>) is testing the use of autonomous vehicle fleet (EZ10) to provide first and last mile service to residents of San Ramon, California, to a local BART transit station: <http://gomentumstation.net/autonomous-vehicles-now-being-tested-on-san-ramon-streets-danville-sanramon/>.

Put together a plan to collect data and design an mHAPP model for such a system. What are the key activity scheduling elasticities of interest?
- (4.8) Create a MATSim model of NYC using the 2010/2011 NYMTC Household Travel Survey (<https://www.nymtc.org/DATA-AND-MODELING/Travel-Surveys/2010-11-Travel-Survey>)

data and the network files from <https://github.com/BUILTNYU/Elsevierbook/tree/master/Data%20Files>.

- a. Use the model to evaluate the effect of the congestion pricing plan proposed by Fix NYC: <https://ny.curbed.com/2018/3/14/17117204/new-york-congestion-pricing-cuomo-subway-uber>.
 - b. Use the model to evaluate the effect of closing the subways between midnight to 5 a.m. and replacing it with a fleet of on-demand shuttle buses: <https://www.linkedin.com/pulse/how-close-down-nyc-subway-from-12am-5am-without-shutting-joseph-chow/>
- (4.9) Construct a synthetic region with zones. Synthesize a population of 1000 individuals residing in this region, including their parameters to solve an mHAPP model. Solve the mHAPP model for the base case for these 1000 individuals and obtain measures such as departure time distributions, number of trips, modes used, travel disutilities, and so on. Now randomly sample 100 individuals from this group. Use the mixed logit model to forecast the population behavior based on the 100-person sample. How accurate is it? Create a new scenario where a new mode option is made available. Solve the population-level market schedule equilibrium and compare that to the prediction using only the 100-person sample.
- (4.10) Create an mHAPP extension in MATSim that would have mHAPP parameters defined for a sample of the population. This sample would be solved to obtain a distribution of responses in a scenario. The distribution can then be used to feed the initial solutions for MATSim in evaluating the new scenario. Evaluate the effectiveness of this approach.

CHAPTER 5

Inverse Transportation Problems

5.1 INTRODUCTION

As important as it is to model the mechanics of interaction between system operators, infrastructure, and travelers, it is just as important to accurately and precisely measure the attributes of the transport system. The models presented in [Chapters 3](#) and [4](#) are only as accurate as the calibration of the system attributes to properly reflect reality. Attributes are divided into two types: (1) parameters that differentiate one instance of a system from another, and (2) discrete operating states in the system derived under various ranges of parameters. For example, a system where line capacity is doubled would behave differently from the original system. Whether a service line is operating at capacity is a state that depends on the line flow and the capacity parameters.

Transport system attributes are particularly difficult to measure due to several reasons. Many of the parameters are highly dynamic and transient; the speeds and travel times of travelers on various roadways can fluctuate over time and depend on exogenous transient variables like weather conditions or whether an incident has occurred. Some parameters are defined by aggregation of individual agent parameters with speed and travel times as good examples. A population of 1000 people traveling on a road may each have different speeds with a population-level speed distribution across that population. Some system attributes are latent or unobservable. For example, travel demand from an origin zone to a destination zone is generally not directly observable; surveys may be conducted from a sample from which population values are then estimated. Similarly, utility that a traveler gains from making a trip is a latent attribute.

General inference of attributes is beyond the scope of this book. Instead, we are interested in estimating attributes (parameters and states) pertaining to the mechanics of an urban transport system, based on different types of information: different aggregations of those attributes, historical observations, and/or observations of other attributes related to those mechanics. Methods to infer attributes from observations of other variables fall under the science of inverse problems ([Tarantola, 2005](#)). If there is a model M that transforms a

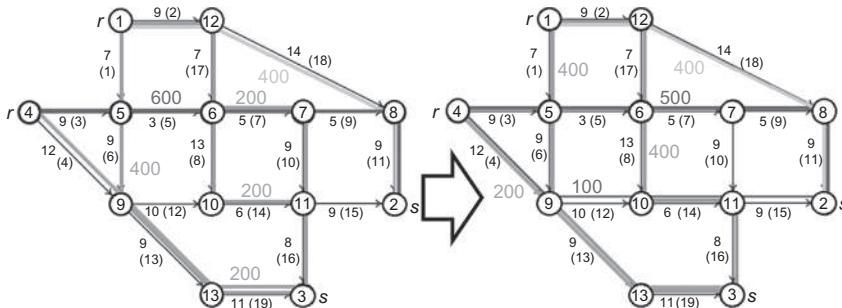


Fig. 5.1 Illustration of a state change in the Nguyen-Dupuis network that leads to a change in observed flows. (Source: Xu et al., 2017.)

set of parameters θ to a set of outputs X as $X = M(\theta)$, then the inverse model deals with estimating a set of parameter estimates $\hat{\theta}$ based on observed outputs x as $\hat{\theta} = M^{-1}(x)$.

To illustrate the significance of this problem, consider the state change in Fig. 5.1 of the classic Nguyen-Dupuis network (Nguyen and Dupuis, 1984) in which the set of link flows changes from the left to the right. The links are numbered in parentheses next to the link travel costs (in minutes). The larger size font values correspond to the path flow: 600 travelers along a path designated by nodes (4, 5, 6, 7, 8, 2) on the left network, for example. The change from one state on the left to the next on the right is due to a change in the link 7 capacity. Based on estimation from sample data (e.g., Alexander et al., 2015), we can estimate the total system travel times (68,400 on left, 70,000 on right) and observe that there is a difference in flow on links 3, 4, 5, 7, 8, 9, 10, 11, 12, 14, and 15. Estimating attributes that can explain *why* the state change occurs, however, requires more structured inference methods that consider the mechanics of the network flows.

As discussed in Chapter 2, the rise of Big Data and IoT has completely altered the ways we can measure attributes of the urban transport system. Information may be obtained from individual level and in much larger samples in a real-time setting. This means that the system attributes can be measured much more precisely and at a more disaggregate level over time.

Historically, inference models have been based on econometric methods that involve building up models from assumptions of relationships to identify cause/effect relationships for trade-off analysis. In machine learning, inference of system attributes is assumed to be a repeated procedure in which pattern recognition and classification of the system state and parameters evolve over time (hybrids that combine machine learning and econometrics

exist as well, and generally there are parallel methods used in both). This is suited to the availability of Big Data in which information can be obtained either in real time and/or from a significant portion of the population. Hence the name of Part C of the book (for this chapter and [Chapter 6](#)) includes learning.

The other major component of Part C is the reference to public information. Because learning techniques make use of more individual information than ever before, it also means that the issue of user and operator privacy will only heighten over time. The term privacy refers to the behavioral resistance to sharing information with others. For organizations, privacy is due to competition, whereas for individuals there can be many other reasons. [Chapter 6](#) deals with identifying methods to mitigate these concerns.

In this chapter, we start off with an overview of standard applications of machine learning in transport. We argue that generic machine learning techniques do not fully capture the unique structure of urban transport systems: the interaction between selfish users and operators, the dependency of users on system attributes and vice versa, and the dependency between different components of the transport system. This leads to the introduction of inverse transportation problems in which the parameters of systems governing the flow of people and goods on a network are estimated using inverse models.

5.2 MACHINE LEARNING APPLICATIONS IN URBAN TRANSPORT

Machine learning techniques have been applied to numerous urban transport problems. Comprehensively reviewing them all would be beyond the scope of this chapter. Illustrative examples from recent years are given to paint a picture of how extensively Big Data and IoT benefits have permeated transport analysis.

To guide this overview, a taxonomy of different types of machine learning methods from [Brownlee \(2013\)](#) is adopted and presented in [Table 5.1](#). Machine learning methods can also be classified by learning style: supervised learning, unsupervised learning, or semisupervised learning.

Regression algorithms refer to methods to fit relationships between variables allowing for a measure of error. Instance-based algorithms, also called memory-based learning or winner-take-all methods, classify based on historical data and compare new data using similarity measures. Regularization algorithms incorporate penalties (the regularization) to favor simpler models. Decision tree algorithms fit classifications to the data according to a tree

Table 5.1 Machine learning methods by similarity compiled by Brownlee (2013)

Method	Examples
Regression algorithms	Ordinary least squares regression (OLSR) Linear regression Logistic regression Stepwise regression Multivariate adaptive regression splines (MARS) Locally estimated scatterplot smoothing (LOESS)
Instance-based algorithms	k-Nearest neighbor (kNN) Learning vector quantization (LVQ) Self-organizing map (SOM) Locally weighted learning (LWL)
Regularization algorithms	Ridge regression Least absolute shrinkage and selection operator (LASSO) Elastic net
Decision tree algorithms	Least-angle regression (LARS) Classification and regression tree (CART) Iterative dichotomiser 3 (ID3) C4.5 and C5.0 Chi-squared automatic interaction detection (CHAID) Decision stump M5 Conditional decision tree
Bayesian algorithms	Naïve Bayes Gaussian naïve Bayes Multinomial naïve Bayes Averaged one-dependence estimators (AODE) Bayesian belief network (BBN) Bayesian network (BN)
Clustering algorithm	k-means k-medians Expectation maximization (EM) Hierarchical clustering
Association rule learning algorithms	Apriori algorithm Eclat algorithm
Artificial neural network algorithms	Perceptron Backpropagation Hopfield network Radial basis function network (RBFN)

Table 5.1 Machine learning methods by similarity compiled by [Brownlee \(2013\)](#)—cont'd

Method	Examples
Deep learning algorithms	Deep Boltzmann machine (DBM) Deep belief networks (DBN) Convolutional neural network (CNN) Stacked auto-encoders
Dimensionality reduction algorithms	Principal component analysis (PCA) Principal component regression (PCR) Partial least squares regression (PLSR) Sammon mapping Multidimensional scaling (MDS) Projection pursuit Linear discriminant analysis (LDA) Mixture discriminant analysis (MDA) Quadratic discriminant analysis (QDA) Flexible discriminant analysis (FDA)
Ensemble algorithms	Boosting Bootstrapped aggregation (Bagging) AdaBoost Stacked generalization (blending) Gradient boosting machines (GBM) Gradient boosted regression trees (GBRT) Random forest
Other algorithms	Support vector machines (SVM)

structure. Bayesian algorithms make use of prior distributions to obtain posterior distributions using Bayes' theorem. Clustering algorithms are used to identify the underlying structure of the data based on commonality. Association rule learning algorithms infer rules that would best explain observed relationships between variables in the data. Artificial neural network algorithms are variants of regression methods that connect different variables together in a way to mimic biological neural networks. Deep learning algorithms are more complex variations that include multiple layers. Dimensionality reduction algorithms are like clustering methods. They classify data by finding ways to reduce the dimensionality. Ensemble algorithms find ways to combine multiple other models to make use of their combined predictions. These algorithms are particularly effective for problems where certain models perform better under certain conditions than others, and automatically assigning the right model to attack a problem by recognizing the condition would improve the overall prediction. [Bishop \(2006\)](#) provides a good introduction to different machine learning methods as well.

Table 5.2 Machine learning applications in urban transport

Reference	Machine learning method	Urban transport application
Allahviranloo and Recker (2013)	Support vector machines	Activity pattern prediction
Cai et al. (2016)	k-nearest neighbor	Short-term traffic forecasting
Kumar et al. (2014)	Neural network	Traffic noise modeling
Li et al. (2014)	LASSO, ridge regression	Freeway traffic state estimation
Luque-Baena et al. (2015)	Self-organizing map	Vehicle detection in traffic monitoring
Lv et al. (2015)	Stacked auto-encoder	Traffic flow prediction
Ma et al. (2015)	Boltzmann machine	Network congestion prediction
Ma et al. (2017a)	Bayesian network	Mode choice
Özdamar and Demir (2012)	Hierarchical clustering	Disaster relief
Rashidi and Mohammadian (2011)	Decision tree	Household trip forecast
Regue and Recker (2014)	Gradient boosting machines	Bike-sharing demand prediction
Yu and Abdel-Aty (2014)	Random forest	Crash injury severity prediction

[Table 5.2](#) provides an illustrative list of studies conducted in urban transport using machine learning methods. Consider the case of Bayesian network modeling to predict mode choice in [Ma et al. \(2017a\)](#). In that study, 44% of the workforce in Luxembourg are cross-border commuters coming from France, Germany, or Belgium. To design transport alternatives for commuters, it is important to understand the explicit relationships between different factors. These relationships are highly nonlinear, which suggests a Bayesian network (BN) approach might be effective in teasing out their contributions to the choice. Based on a hill climbing algorithm and structural restrictions, the authors come up with a set of marginal probability tables for the BN model in [Fig. 5.2](#).

Machine learning techniques have been found to be generally more accurate predictors than econometric models, although they are more likely to run the risk of overfitting to a data set such that generalizations are harder to make. Given these trade-offs of learning the parameters and states of an urban transport system over time, these methods are effective.

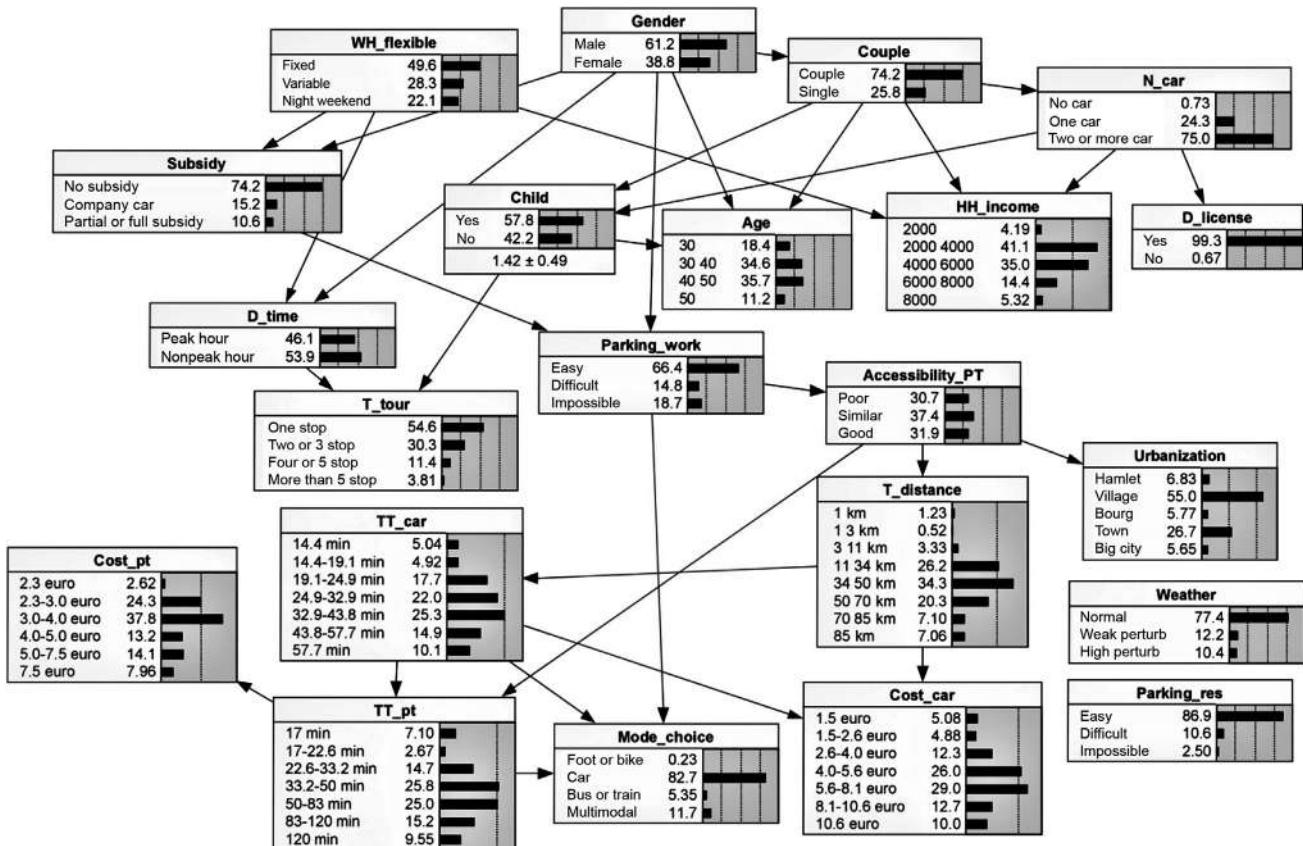


Fig. 5.2 Bayesian network model of Luxembourg commuter mode choice. (Source: Ma et al., 2017a)

Nonetheless, the question of effectiveness brings us back to the example shown in Fig. 5.1. Many transport system attributes are related to route decisions of travelers and logistics decisions by operators to minimize operating costs. One series of algorithms not included in Table 5.1 are mathematical programming algorithms, which refer to machine learning algorithms that assume the behavior of a system is governed by one or more mathematical programming models. It is particularly relevant to transport networks because of two qualities in network inference. First, state changes can involve system parameters like link capacities due to weather or incidents. Second, route decisions are governed by behavioral mechanisms like route choice behavior. If we can correctly blend the behavioral rules used by travelers and operators into the inference procedure, it would make the inference problem more efficient and able to predict effects of structural changes to the network (e.g., when a public agency sets up a new route to serve passengers).

Inference in a network context has a long history. The simplest network inference methods relate an observable set of attributes to the desired parameters directly through network structure, for example, link flow is the sum of the proportions of all OD flows that use that link (Van Zuylen and Willumsen, 1980), absent of any route decision sensitivity. Route sensitivity is modeled by explicitly mapping observed variables like link counts to latent path flows and mapping those path flows to desired parameters like OD flows (e.g., Vardi, 1996; Tebaldi and West, 1998). Other researchers modeled the route choice behavior by inserting it into the inverse models: Cascetta et al. (1996), Vovsha and Bekhor (1998), Srinivasan and Mahmassani (2000), Dia (2002), Frejinger and Bierlaire (2007), Ben-Elia and Shiftan (2010), Gao (2012), and Fosgerau et al. (2013). Markovian models (Akamatsu, 1996) have been introduced to overcome route enumeration. Nonenumerative route behavior mechanisms integrated with network structure have also been proposed (Yang et al., 1992; Ashok and Ben-Akiva, 2002).

One emerging machine learning technique is based on the notion of inverse optimization. In this method, the behavior of operators and users is assumed to follow a mathematical programming model in addition to the observed data. Based on observations of outcomes and prior distributions of the parameters, new posterior parameters are estimated in a Bayesian learning context. The next section delves into different types of inverse optimization models along with a review of recent developments in this area.

5.3 INVERSE TRANSPORTATION PROBLEMS

The first inverse optimization model was proposed by [Burton and Toint \(1992\)](#) for the inverse shortest path problem and was further generalized by [Ahuja and Orlin \(2001\)](#) to any inverse linear programming problem. Since many transportation problems can be classified as mathematical programming problems, specific applications of inverse optimization to transportation problems are regarded as inverse transportation problems.

There have been several advances and applications in inverse optimization (IO) since then. [Table 5.3](#), based on [Xu et al. \(2017\)](#), provides a summary of these advances. In terms of methodological advances, [Wang \(2009\)](#)

Table 5.3 Inverse optimization advances and applications

Methodological advances	New applications
Burton and Toint (1992)	Inverse shortest path Day et al. (2002) Network calibration
Ahuja and Orlin (2001)	Inverse linear program Burkard et al. (2004) Inverse median problem
Wang (2009)	Inverse integer program Agarwal and Ergun (2008) Mechanism design
Güler and Hamacher (2010)	Link capacities in minimum cost flow problem Brucker and Shakhlevich (2009) Inverse scheduling
Zhang and Zhang (2010)	Inverse nonlinear program Bertsimas et al. (2012) Financial portfolio management
Chow and Recker (2012)	Multiagent IO, inverse VRP with side constraints Birge et al. (2017) Electricity market structure
Aswani et al. (2015)	Noisy data bilevel problem Chow et al. (2014) Inverse traffic assignment
Bertsimas et al. (2015)	Inverse variational inequality Chan et al. (2014) Cancer therapy
Esfahani et al. (2015)	Incomplete info robust problem You et al. (2016) Urban truck forecasting
Chan and Lee (2017)	Multiobjective Pareto set Hong et al. (2017) Mixed logit estimation
Xu et al. (2017)	Latent link capacity effects

proposed a solution method based on a cutting plane algorithm to solve the inverse mixed integer linear programming problem. Güler and Hamacher (2010) proposed an IO model for estimating link capacities for minimum cost flow problems. Zhang and Zhang (2010) proposed an inverse quadratic program. Chow and Recker (2012) proposed an inverse vehicle routing problem that includes side constraint estimation. Bertsimas et al. (2015) proposed an inverse variational inequality to estimate the parameters that would lead to observed patterns under equilibrium. Aswani et al. (2015) proposed a bilevel model to estimate parameters allowing for noisy data. Esfahani et al. (2015) modeled the noisy information problem as a robust optimization model. Chan and Lee (2017) proposed the use of inverse optimization to estimate the Pareto set for a multiobjective convex optimization problem. Xu et al. (2017) proposed a multiagent IO framework to estimate heterogeneous parameters across a population and to infer latent network parameters like capacity effects.

Applications include network calibration (Day et al., 2002), facility location (Burkard et al., 2004), scheduling (Brucker and Shaklevich, 2009), mechanism design for forming alliances between multiple network operators in a multicommodity flow problem (Agarwal and Ergun, 2008), financial portfolio management (Bertsimas et al., 2012), electricity market inference (Birge et al., 2017), inverse traffic assignment (Chow et al., 2014), cancer therapy (Chan et al., 2014), urban truck forecasting (You et al., 2016), and mixed logit estimation for transit route choice modeling (Hong et al., 2017).

In the following subsections, different types of IO models are introduced, followed by examples in corresponding transportation applications.

5.3.1 Inverse Linear Programming

Consider first the linear programming problem under matrix notation: $\min \{c^T x : Ax \leq b, x \geq 0\}$. The n -dimensional vector x is a set of decision variables, c is an n -dimensional vector of objective coefficients, A is an $m \times n$ constraint matrix, and b is an m -dimensional vector of side constraint values.

In the basic inverse optimization problem, a researcher observes x^* and has a prior distribution of the coefficients, c_0 . Based on this information, they seek to update a posterior value of c such that x^* is an optimal solution by perturbing it minimally from c_0 in L_1 norm. This is expressed mathematically in Eq. (5.1).

$$\min_c |c_0 - c| : x^* = \operatorname{argmin} \{ c^T x : Ax \leq b, x \geq 0 \} \quad (5.1)$$

The L_1 norm minimization is used because inverse problems are generally ill-posed problems (variants also exist for L_∞ -norm). For a given observation or set of observations, there are an infinite set of solutions that fit to them. Selection rules are used to regularize the problem to obtain a unique solution that is physically meaningful and stable. The L_1 norm minimization can provide such regularity for inverse problems (Tenorio, 2001). This ensures that a unique solution can be obtained given a prior c_0 . Ahuja and Orlin (2001) showed that Eq. (5.1) can be reformulated as an LP. This is done by introducing two nonnegative decision variable vectors e and f such that their difference is equal to $c_0 - c$: $c_0 - c = e - f$, where $c = c_0 - e + f$. The problem in Eq. (5.1) is reformulated using strong duality and dual feasibility conditions with dual variable γ of the original LP and weights w , as shown in Eq. (5.2).

$$\min_{\gamma, e, f} w^T e + w^T f \quad (5.2a)$$

Subject to

$$A^T \gamma \geq c_0 - e + f \quad (5.2b)$$

$$b^T \gamma = (c_0 - e + f)^T x^* \quad (5.2c)$$

$$\gamma, e, f \geq 0 \quad (5.2d)$$

Minimum positive (f) or negative (e) perturbations are achieved by ensuring that the solution's dual problem is feasible via Eq. (5.2b) and the dual objective value is equal to the primal objective value—strong duality—via Eq. (5.2c). The model is illustrated in Exercise 5.1.

Exercise 5.1

Consider the following network in Fig. 5.3, which has prior travel times labeled on each link. You run a Google Maps search from node 1 to node 6 and it returns a path along links [(1, 2), (2, 5), (5, 3), (3, 6)]. Formulate and solve the inverse shortest path (only allow positive differences f , $c = c_0 + f$, to represent congestion effects) based on L_1 norm with weights $w = \frac{1}{c_0}$ (which seeks relative perturbations) to infer the congestion effect taking place for Google to make such a recommendation.

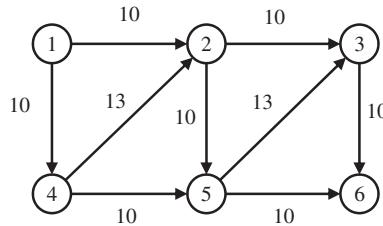


Fig. 5.3 Sample network for Exercise 5.1.

The LP is a shortest path problem which has equality constraints for the flow conservation. This is expressed as follows.

$$\min_x Z = \sum_{(i,j)} c_{ij} x_{ij}$$

Subject to

$$\begin{aligned} x_{12} + x_{14} &\geq 1 \\ -x_{12} - x_{42} + x_{25} + x_{23} &\geq 0 \\ -x_{23} - x_{53} + x_{36} &\geq 0 \\ -x_{14} + x_{42} + x_{45} &\geq 0 \\ -x_{25} - x_{45} + x_{53} + x_{56} &\geq 0 \\ -x_{56} - x_{36} &\geq -1 \end{aligned}$$

An optimum solution based on the prior c_0 using simplex algorithm is the path $[(1, 2), (2, 3), (3, 6)]$, which is clearly different from the observed path. The observed path implies $x^* = [1, 0, 0, 1, 0, 0, 0, 1, 1]$. The inverse shortest path as specified is as follows.

$$\min_f W = \frac{f_{12}}{10} + \frac{f_{14}}{10} + \frac{f_{23}}{10} + \frac{f_{25}}{10} + \frac{f_{36}}{10} + \frac{f_{42}}{13} + \frac{f_{45}}{10} + \frac{f_{53}}{13} + \frac{f_{56}}{10}$$

Subject to

$$\begin{aligned} y_1 - y_2 &\leq 10 + f_{12} \\ y_1 - y_4 &\leq 10 + f_{14} \\ y_2 - y_3 &\leq 10 + f_{23} \\ y_2 - y_5 &\leq 10 + f_{25} \\ y_3 - y_6 &\leq 10 + f_{36} \\ -y_2 + y_4 &\leq 13 + f_{42} \\ y_4 - y_5 &\leq 10 + f_{45} \\ -y_3 + y_5 &\leq 13 + f_{53} \\ y_5 - y_6 &\leq 10 + f_{56} \\ y_1 - y_6 &= 10 + f_{12} + 10 + f_{25} + 10 + f_{36} + 13 + f_{53} \\ f &\geq 0 \end{aligned}$$

The dual variable γ is not constrained to be nonnegative because the shortest path constraints are meant to be equality constraints. The solution to the inverse shortest path problem is $c_{23}^* = 10 + 13 = 23$ and $c_{56}^* = 10 + 13 = 23$ with an objective value of $W^* = 2.6$. This suggests the least perturbing (and hence most likely) scenario is one where congestion occurs at links (2, 3) and (5, 6) to lead Google to suggest its route.

What happens if the true perturbation was much different? For example, if the true $c_{23}^* = 100$, then repeated observations from different shortest path queries for different OD pairs and parameter updates should continually push the value of c_{23}^* up either until it reaches 100 or a value in between for which the network does not recognize any difference in effect. For example, if all shortest paths for every OD pair get perturbed the same way for any value $60 \leq c_{23}^* \leq 100$, the IO would also not distinguish between those values. In that sense, IO is not trying to estimate a true parameter value but is instead searching for a parameter that has the best fitting network effect. Regarding the repeated learning, this is essentially a deterministic Bayesian learning framework. Stochastic Bayesian learning is also possible with inverse optimization for a population of heterogeneous agents with their own respective models, which we see in [Section 5.4](#).

[Exercise 5.1](#) illustrates one use of IO: it can infer parameters in a third-party system (such as Google) as a way of reverse engineering their known information, beliefs, or operating policy, which has strong implications on why private operators have concerns for sharing data with the public. This concept is explored further in [Chapter 6](#). There are many other applications as well. IO can be used to monitor one's own system for unexpected deviations or inconsistencies in policy, which can be useful for identifying incidents in a complex system or for catching agents within the system that are operating out of the norm. It can be used to measure parameters as revealed preferences, such as in inferring revealed objective weights of a multiobjective optimization as explored in [Chan et al. \(2014\)](#). It can be used to measure latent parameters such as capacity effects, which is illustrated further in this chapter. And of course, IO can also be used in an online Bayesian learning context to update one's system based on observations of outputs impacted by uncontrolled parameters.

5.3.2 Inverse Integer Programming

Like inverse linear programming, inverse integer programming (IP) has many applications in transport systems. Many transport logistics systems are based on integer programming: routing, scheduling, and facility location, for example.

The challenge with inverse IP is that the strong duality condition in Eq. [\(5.2c\)](#) no longer holds because of the presence of a duality gap ([Gomory, 1958; Wolsey, 1981](#)). [Gomory's \(1958\)](#) cutting plane algorithm (i.e., the “Gomory cut”) has been used to address duality gap in solving IP

problems, and Wang (2009) showed that it can also be used to solve inverse IP problems. Algorithm 5.1 is shown here.

Algorithm 5.1: Wang's (2009) Cutting Plan Algorithm for Solving the Inverse Integer Programming Problem

Inputs: observed decision variables of original IP x^* , parameters (A, b, I) of IP $\max_x \{c^T x : Ax \leq b, x \geq 0, x_i \in \mathbb{Z}, \forall i \in I\}$, prior objective coefficients c_0

0. Initiate an empty set $\mathcal{S} = \{\}$ of constraints.

1. Solve Eq. (5.3) and let (y^*, e^*, f^*) be an optimal solution (for minimization IPs Eq. (5.3c) would be less than or equal to instead):

$$\min_{y, e, f} w^T e + w^T f \quad (5.3a)$$

Subject to

$$A^T y \geq c_0 - e + f \quad (5.3b)$$

$$(c_0 - e + f)^T x^* \geq (c_0 - e + f)^T \tilde{x}^s, \forall \tilde{x}^s \in \mathcal{S} \quad (5.3c)$$

$$y, e, f \geq 0 \quad (5.3d)$$

Note: other constraints may be added to this to ensure that the constraints in the original IP are met. For example, if $c \geq 0$ is needed to work, then the inverse problem should include constraints $c_0 - e + f \geq 0$.

2. $\tilde{x} = \operatorname{argmax}_x \{(c_0 - e^* + f^*)^T x : Ax \leq b, x \geq 0, x_i \in \mathbb{Z}, \forall i \in I\}$.

If $(c_0 - e^* + f^*)^T x^* \geq (c_0 - e^* + f^*)^T \tilde{x}$, then stop, and $c^* = c_0 - e^* + f^*$.

Otherwise, $\mathcal{S} := \mathcal{S} \cup \{\tilde{x}\}$ and go to 1.

Outputs: Estimated posterior objective coefficients c^* .

Note that although the inverse IP solution does include dual prices as well, they may not have any economic interpretation (as discussed by Gomory) if they are based on binding constraints created from the cuts. Wang (2009) proved that the algorithm converges in finite iterations.

To demonstrate this algorithm, we apply this to a variant Dial-a-Ride problem (see Chapter 7). The model we consider is shown in Eq. (5.4), modified from Cordeau and Laporte (2007). We incorporate weighted objectives for ride time and wait time (as pickup time).

$$\min Z = \gamma \sum_{k \in V} \sum_{i \in N} \sum_{j \in N} c_{ij} X_{ijk} + \alpha \sum_{i \in P} R_i + \beta \sum_{i \in P} T_i \quad (5.4a)$$

Subject to

$$\sum_{k \in V} \sum_{j \in N} X_{ijk} = 1, \forall i \in P \quad (5.4b)$$

$$\sum_{j \in N} X_{0jk} = \sum_{j \in N} X_{j,2n+1,k}, \forall k \in V \quad (5.4c)$$

$$\sum_{j \in N} X_{0jk} \leq 1, \forall k \in V \quad (5.4d)$$

$$\sum_{j \in N} X_{ijk} = \sum_{j \in N} X_{n+i,jk}, \forall i \in P, k \in V \quad (5.4e)$$

$$\sum_{j \in N} X_{jik} = \sum_{j \in N} X_{ijk}, \forall i \in P \cup D, k \in V \quad (5.4f)$$

$$T_i - T_j \leq -d_i - t_{ij} + (1 - X_{ijk})M, \forall i, j \in N, k \in V \quad (5.4g)$$

$$W_i - W_j \leq -q_i + (1 - X_{ijk})M, \forall i, j \in N, k \in V \quad (5.4h)$$

$$T_{n+i} - T_i - R_i \leq d_i, \forall i \in P \quad (5.4i)$$

$$0 \leq W_i \leq u, \forall i \in N \quad (5.4j)$$

$$X_{ijk} \in \{0, 1\} \quad (5.4k)$$

$$t_{i,n+1} \leq R_i \leq R_{\max} \quad (5.4l)$$

$$0 \leq T_i \leq T_{\max} \quad (5.4m)$$

where $P = \{1, \dots, n\}$ is the set of pickup locations, $D = \{n+1, \dots, 2n\}$ is the set of drop-off locations, vehicle “depots” are $\{0_1, \dots, 0_{|V|}, 2n+1\}$, N is the set of all nodes, q_i , $i \in P$, is the group size (assumed to be 1 for this example), d_i is the service duration (loading/unloading/waiting), u is vehicle capacity, c_{ij} is the travel cost, t_{ij} is the travel time, R_{\max} is the maximum ride time, α is a weight that is assigned to the minimizing ride time objective, X_{ijk} is the route decision of vehicle k , T_i , $i \in \{P, D\}$, is the start of service at node i , W_i is the load upon leaving node i , and R_i , $i \in P$, is the ride time of pickup i . Eq. (5.4a) is a weighted objective function including operator travel time, passenger ride time, and wait time. Eqs. (5.4b)–(5.4f) are flow conservation constraints that ensure at least one vehicle is assigned to serve the passengers. Eq. (5.4g) sets the arrival times and acts as subtour elimination constraints. Eq. (5.4h) captures the vehicle load at each stop. Eq. (5.4i) determines the ride time for each passenger. Eq. (5.4j) is a vehicle capacity constraint. The remainder are boundary constraints and binary constraints.

Exercise 5.2 illustrates how IO can be used to reverse engineer mobility companies’ routing algorithms. The IP and LP solutions to this example are obtained using commercial solvers from MATLAB. Other popular solvers include Gurobi (free for academics), CPLEX, and AMPL.

Exercise 5.2

Consider a single microtransit shuttle shown in Fig. 5.4 serving three customers making simultaneous requests, each with a pickup and a drop-off location on a Euclidean space. The vehicle is assumed to have capacity of $u=2$ passengers. An open tour is assumed ($c_{i,2n+1} = t_{i,2n+1} = 0 \forall i \in N$), travel costs are equivalent to travel times ($c_{ijk} = t_{ijk}$), and dwell times at each stop is $d_i = 2$, and $R_{max} = T_{max} = 1440$.

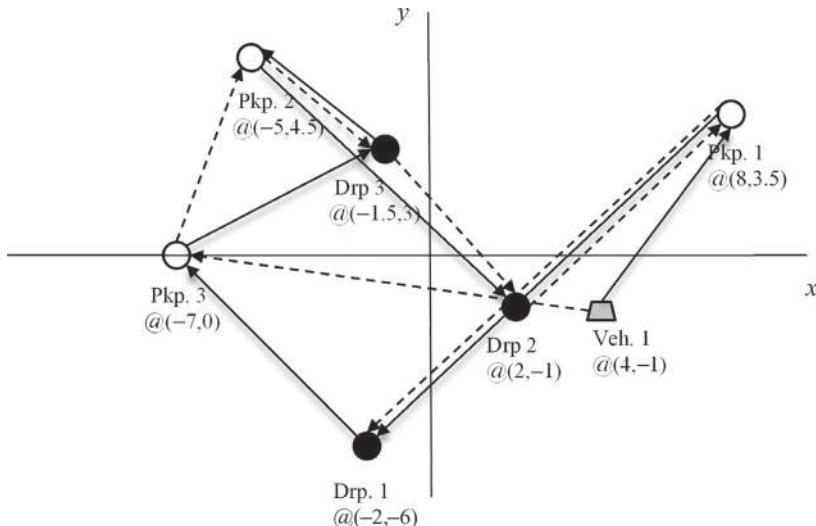


Fig. 5.4 Example dial-a-ride problem for Exercise 5.2.

Assuming a prior value of $\alpha_0 = \beta_0 = 1$ and $\gamma \equiv 1$, Eq. (5.4) produces the following route $(0, 3P, 2P, 3D, 2D, 1P, 1D)$ (shown as a series of dashed arrows). However, the shuttle is observed to take route $(0, 1P, 1D, 3P, 3D, 2P, 2D)$ (shown as a series of solid arrows) instead. Use Algorithm 5.1 to solve the inverse IP under L_1 norm to estimate (α, β) to fit this observation.

Let us denote $x^* = [X^*, T^*, W^*, R^*]$ to reflect the observed $(0, 1P, 1D, 3P, 3D, 2P, 2D)$ and associated decision variables. We seek

$$\begin{aligned} \min_{\alpha, \beta} & |\alpha_0 - \alpha| + |\beta_0 - \beta| : x^* = \operatorname{argmin}_{x^*} \left\{ \sum_{k \in V} \sum_{i \in N} \sum_{j \in N} c_{ijk} X_{ijk} + \alpha \sum_{i \in P} R_i \right. \\ & \left. + \beta \sum_{i \in P} T_i : Ax \leq b, x \geq 0, X \in \mathbb{Z}^+ \right\} \end{aligned}$$

where A is the original constraint matrix for Eqs. (5.4b)–(5.4m) and b is a vector of the original right side constraint values. We define $\alpha_0 - \alpha = e_\alpha - f_\alpha$ and $\beta_0 - \beta = e_\beta - f_\beta$, and set

$$d[\alpha, \beta] = \begin{bmatrix} c_1 \\ \dots \\ \alpha_0 - e_\alpha + f_\alpha \\ \dots \\ \beta_0 - e_\beta + f_\beta \\ \dots \\ c_n \end{bmatrix}, \text{ where } c_1 \dots c_n \text{ are the coefficients of the original}$$

model that are not being perturbed. To initiate [Algorithm 5.1](#), we set $\mathcal{S} = \{\}$ and solve the following LP:

$$\min_{y, e, f} e_\alpha + f_\alpha + e_\beta + f_\beta$$

Subject to

$$A^T y \geq d$$

$$\alpha_0 - e_\alpha + f_\alpha \geq 0$$

$$\beta_0 - e_\beta + f_\beta \geq 0$$

$$e, f \geq 0$$

The additional constraints $\alpha_0 - e_\alpha + f_\alpha \geq 0$ and $\beta_0 - e_\beta + f_\beta \geq 0$ ensure that the coefficients cannot go negative as the IP would not work, and y is left unbounded since the original IP has equality constraints.

The solution obtained is $e_\beta = 1$, which implies a $\tilde{\alpha}^1 = 1$ and $\tilde{\beta}^1 = 1 - 1 = 0$ in the first iteration. The values are plugged back in to Eqs. (5.10)–(5.22) to obtain \tilde{x}^1 . The condition is checked: $d[\tilde{\alpha}^1, \tilde{\beta}^1]^T x^* \leq d[\tilde{\alpha}^1, \tilde{\beta}^1]^T \tilde{x}^1$. In this case, the inequality is switched to “ \leq ” because the original IP objective is a minimization problem. We get $d[\tilde{\alpha}^1, \tilde{\beta}^1]^T x^* = 75.5592$ and $d[\tilde{\alpha}^1, \tilde{\beta}^1]^T \tilde{x}^1 = 72.0039$. Since the inequality is not satisfied, we add \tilde{x}^1 to the set of cutting planes: $\mathcal{S} = \{\tilde{x}^1\}$. The new LP that is solved is as follows:

$$\min_{y, e, f} e_\alpha + f_\alpha + e_\beta + f_\beta$$

Subject to

$$A^T y \geq d$$

$$\alpha_0 - e_\alpha + f_\alpha \geq 0$$

$$\beta_0 - e_\beta + f_\beta \geq 0$$

$$-2.9284e_\beta + 2.9284f_\beta + 6.5774e_\alpha - 6.5774f_\alpha \leq -6.4838$$

$$e, f \geq 0$$

The new constraint is from the cutting plane due to \tilde{x}^1 . Based on this LP, we obtain $\tilde{\alpha}^2 = 1 + 0.5405 = 1.5405$ and $\tilde{\beta}^2 = 1 - 1 = 0$. The stopping condition is still not met: $d[\tilde{\alpha}^2, \tilde{\beta}^2]^T x^* = 91.2120$ and $d[\tilde{\alpha}^2, \tilde{\beta}^2]^T \tilde{x}^2 = 89.8970$. It takes two more iterations to reach $\alpha^* = \tilde{\alpha}^4 = 2.1255$ and $\beta^* = \tilde{\beta}^4 = 0.0245$ with IP objective value of $d[\tilde{\alpha}^4, \tilde{\beta}^4]^T x^* = d[\tilde{\alpha}^4, \tilde{\beta}^4]^T \tilde{x}^4 = 110.195$. The total perturbation is $e_\alpha^* + f_\alpha^* + e_\beta^* + f_\beta^* = 2.101$.

Although we obtain parameters that can achieve the same objective value, it does not mean that the optimal decision variables under the optimal learned parameters are the same as the observed decision variables. However, the solution is highly informative. This example shows that the observed pattern suggests this operator values ride time objective much more than wait time objective: in this case, nearly 100 times more preferred given the prior.

Another useful metric is the objective value. We see that the observed solution has an objective value of 110.195, whereas the model with the prior parameters has an objective value of 151.642. This difference can be used to help a third party infer an operator's policies. For example, the reduced objective values suggest that the operator's policy is *effectively* less constrained than the prior model. While we do not know the specific operational policy or algorithm, we can use this information to help infer the operator's latent policy. By testing two different policies on the prior model, we might see that one candidate policy leads to an objective of 130 while another to 115. The latter policy, in this case, is quantifiably more likely to be the one in use.

The inverse IP can be used to estimate parameters of different IP models in an online learning framework. One example is the calibration of the HAPP/mHAPP models discussed in [Chapter 4](#). [Chow and Recker \(2012\)](#) investigated the effectiveness of this method from observations of travel diary data for different households. Travel diary data is extracted from the 2001 California Household Travel Survey for residents of Orange County, CA. The activity locations and road network from the OCTA Model are shown in [Fig. 5.5](#). Based on the travel survey data, median arrival times to different activity types from the population are shown in [Table 5.4](#).

The median arrival times are used as prior goal arrival times from which individual goal arrival times are jointly estimated along with weights of the multiobjective utility function for an individual household. Since goal

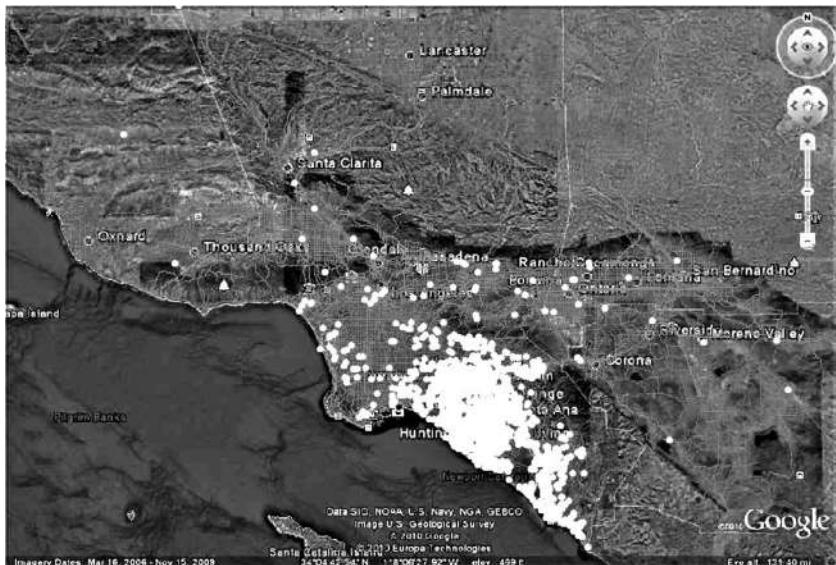


Fig. 5.5 OCTAM Network and activity locations of households in Orange County, CA, from the 2001 California Household Travel Survey. (Source: [Chow and Recker, 2012](#).)

arrival times are side constraint values, an exogenous early and late arrival penalty is assumed for everyone in the population, as shown in Eq. (5.5) added to the HAPP model.

$$\min \hat{Z} + \sum_{u \in P} e_u Z_{e,u} + \sum_{u \in P} l_u Z_{l,u} \quad (5.5a)$$

Subject to

$$T_u + Z_{e,u} - Z_{l,u} = g_u, \quad \forall u \in P \quad (5.5b)$$

$$Z_{e,u}, Z_{l,u} \geq 0, \quad \forall u \in P \quad (5.5c)$$

\hat{Z} is the original HAPP model objective without arrival delay costs. The set P includes activities that need to be conducted. e_u and l_u are the exogenous penalty rates for being early or late from the goal arrival time g_u , respectively. $Z_{e,u}$ and $Z_{l,u}$ are deviations from the goal arrival time. Based on the structure of the constraints, the $Z_{e,u}$ and $Z_{l,u}$ are treated as the “parameters” for estimation in the inverse HAPP with the priors set to zero and g_u is a latent goal arrival time per activity for the household. For example, a prior value of early arrival deviation is defined as $Z_{e,u}^0 = \max(0, g_u^0 - T_u^*)$, where g_u^0 is the prior from [Table 5.4](#) and T_u^* is the observed arrival time at the activity. Similar logic applies to deriving $Z_{l,u}^0 = \max(0, T_u^* - g_u^0)$. The posterior side constraint

Table 5.4 Median arrival times by activity type from 2001 California Household Travel Survey

Activity	Median arrival time	Activity	Median arrival time
Work at home	11:45 a.m.	Childcare, daycare, after school	9:36 a.m.
Eat/prepare meal at home	3:00 p.m.	Eat out	1:34 p.m.
Watch TV/videos at home	4:30 p.m.	Medical	11:55 a.m.
Shop by phone/TV/internet at home	2:25 p.m.	Fitness activities	2:45 p.m.
Exercise at home	8:30 a.m.	Recreational (vacation, camp)	12:20 p.m.
Other at home	3:00 a.m.	Entertainment (movies, club, bar)	5:32 p.m.
Wait for/get on vehicle	1:40 p.m.	Visit friends/relatives	3:00 p.m.
Leave/park a vehicle	11:06 a.m.	Community meetings, civic event	3:25 p.m.
Boarding airplane, rail, intercity bus	11:16 a.m.	Occasional volunteer work	11:41 a.m.
Alighting airplane, rail, intercity bus	3:30 p.m.	Church, temple, religious meeting	5:30 p.m.
Pick up someone or get picked up	3:00 p.m.	Buy gas	1:35 p.m.
Drop off someone or get dropped off	8:30 a.m.	Incidental shopping (groceries)	2:09 p.m.
Work	8:10 a.m.	Major shopping	1:50 p.m.
Work-related	11:30 a.m.	ATM, bank, post office, utilities	1:00 p.m.
School (preK to 12)	8:00 a.m.	Other personal/household	1:05 p.m.
School (college, vocational)	9:50 a.m.	Accompany another for activity	2:02 p.m.

g_u is then estimated by leaving it as a decision variable in the inverse optimization problem and adding the constraint from Eq. (5.6) (modified from Eq. 5.5b) into Eq. (5.3).

$$T_u^* + (Z_{e,u}^0 - e_{Zeu} + f_{Zeu}) - (Z_{l,u}^0 - e_{Zlu} + f_{Zlu}) = g_u, \forall u \in P \quad (5.6)$$

The variables $e_{Zeu}, f_{Zeu}, e_{Zlu}, f_{Zlu}$ are the perturbation variables from the prior deviation values. This allows the side constraint g_u to be estimated

simultaneously as the objective coefficients. This method can be applied to any side constraint variable that is related to objective variables through an equality constraint and constitutes one of the key contributions of [Chow and Recker \(2012\)](#).

To illustrate this, consider an activity in which the median or average goal arrival time from the population is shown to be 9 a.m. A sampled individual is observed to arrive at 9:25 a.m. We do not know what their goal arrival time is. It could be the same as the population average, or it could be 9:30 a.m., or any other time. Based on these values, the prior deviations are set to $Z_{l,u}^0 = \max(0, 9:30 - 9) = 30$ min and $Z_{e,u}^0 = \max(0, 9 - 9:30) = 0$. Suppose based on the observed pattern we find that $e_{Zeu}^* = f_{Zeu}^* = e_{Zlu}^* = 0$ and $f_{Zeu}^* = 10$ min. This implies that for this individual, $g_u^* = 9:15$ a.m., $Z_{e,u}^* = 0$, $Z_{l,u}^* = 10$ min. The result is interpreted as the individual preferring to arrive at 9:15 a.m. but due to the mix of scheduling preferences and travel disutilities they end up arriving 10 min late to this activity. This structure gives much more flexibility to infer heterogeneous goal arrivals and schedule deviations.

Household record no. 1048899 from the survey data is shown to have a single-vehicle household member leave home at 7:35 a.m., arrive at work at 8:10 a.m., go eat at 1:00 p.m., go shopping at 3:35 p.m., and return home for dinner at 4:50 p.m. Solving the inverse HAPP with goal arrival time estimation as formulated in [Chow and Recker \(2012\)](#) with [Algorithm 5.1](#) leads to a converged solution after 10 iterations, taking 4.015 s on a 64-bit Intel Core i7 CPU with 2.67 GHz, 4 GB RAM, Windows 7 operating system with a CPLEX IP solver. We call this solution the InvHAPPb (the “b” is to indicate the estimation of side constraints).

By comparison, a set of uninformed parameters is one where all the objective coefficients are set to 1. The solution to this problem is noted as “Indifferent” to indicate a lack of difference in objective weights. Lastly, the inverse HAPP can also be estimated without allowing for estimation of the side constraints. We call this estimated solution the “InvHAPPSTW” which indicates a conventional inverse HAPP with soft time windows. A comparison of schedules generated by HAPP using the three estimated sets of parameters alongside the observed schedule is shown in [Fig. 5.6](#). By not calibrating the parameters, the resulting output schedule is most different from the observed schedule. Estimating the objective coefficients without allowing side constraint estimates leads to a similar schedule but the timing ends up differing.

Another application of inverse IP is on urban truck routing, which is discussed in [Section 4.5](#). When observing truck route patterns, a public agency is interested in learning the parameters used by the truck operators. This

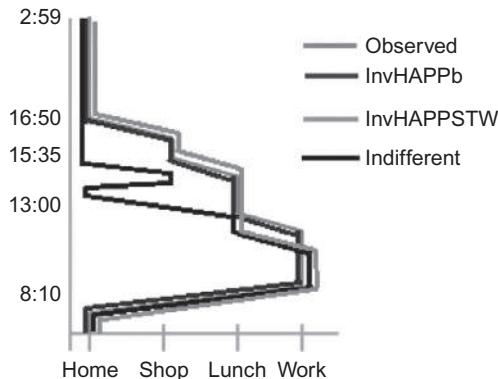


Fig. 5.6 Comparison of HAPP outputs based on different parameters. (Source: [Chow and Recker, 2012](#).)

leads to a learning problem that can be modeled as an inverse VRP like the inverse DARP shown earlier in [Exercise 5.2](#).

[You et al. \(2016\)](#) conducted a case study of the drayage truck activity near the San Pedro Bay Ports in Southern California. Sample truck GPS data from 2010 was collected with sample trajectory data shown in [Fig. 5.7](#).

In one example based on the truck GPS data, [You et al. \(2016\)](#) examined the increased difficulty of estimating a fleet of multiple vehicles simultaneously. This is conducted by taking a fleet of two trucks (denoted A and B) making various trips over a day. When isolated as individual agents, the inverse VRP can perfectly reproduce the observed patterns. However, when considering the management of the two trucks as one agent operator, the inverse VRP is less able to exactly capture the observed pattern. This is summarized in [Fig. 5.8](#).

5.3.3 Inverse Nonlinear Programming

A third category of IO is with nonlinear programming (NLP) problems. In these problems, the dual conditions used in the IO methodology are replaced with more generalized Karush-Kuhn-Tucker (KKT) optimality conditions. The nonlinear traffic assignment problem in Eq. (3.3) in [Chapter 3](#) is used as a representative example of nonlinear convex optimization models.

In the inverse traffic assignment problem, it is assumed that we observe path flows v_{krs}^* and corresponding link flows x_a^* for a network $G[N, A]$ with OD demand W . The inverse problem is to determine the set of parameters $\theta_a \geq 0$ (e.g., capacity effects) of the link performance function $c_a[x_a; \theta_a]$ for

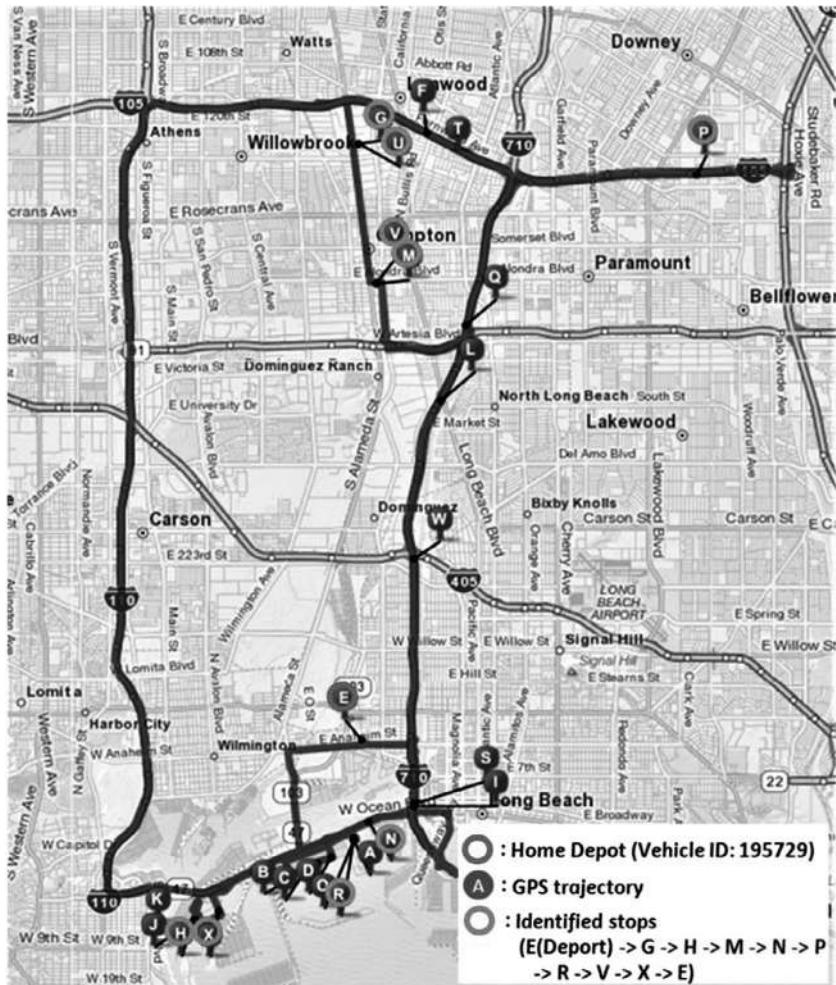


Fig. 5.7 Sample GPS data from drayage trucks out of the San Pedro Bay Ports. (Source: You et al., 2016.)

link $a \in A$ such that the observed flows are optimal (i.e., in a user equilibrium). In Eq. (3.3), the constraint (3.3c), $x_a = \sum_{k \in K} \sum_{(r, s) \in W} \delta_{aks} v_{krs}$, is purely definitional and can be left out of the constrained primal problem. In that case, only one set of Lagrange multipliers are needed, μ_{rs} , for each of the OD pairs (r, s) pertaining to each demand equality constraint in Eq. (3.3b). Let us define l_k as the set of links that form path k . The inverse traffic assignment problem is as shown in Eq. (5.7) based on the derivation in Chow et al. (2014).

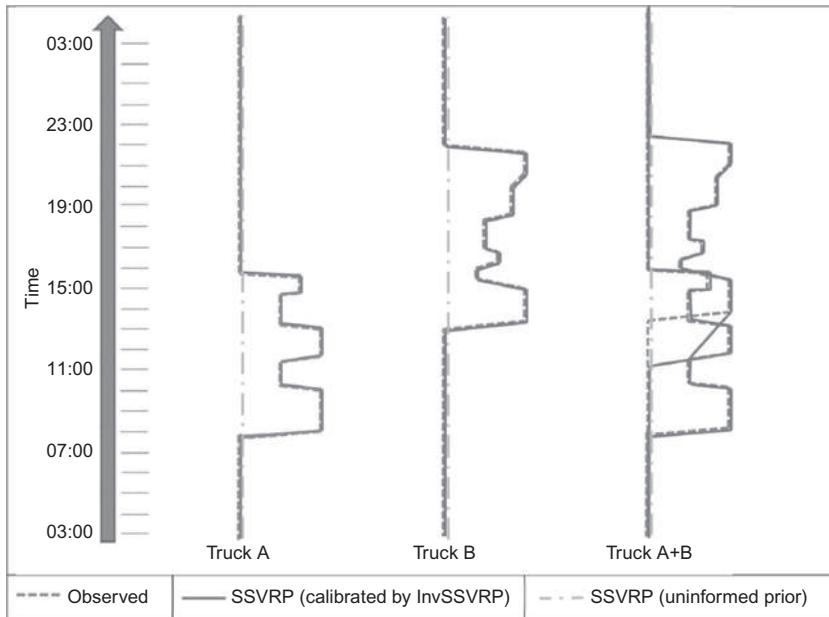


Fig. 5.8 Comparison of inverse VRP results for individual and fleet of two vehicles.
(Source: You et al., 2016.)

$$\min_{\mu, e, f} \sum_{a \in A} w_a(e_a + f_a) \quad (5.7a)$$

Subject to

$$\sum_{a \in l_k} c_a [x_a^* [v_{krs}^*]; \theta_a^0 - e_a + f_a] - \mu_{rs} \geq 0, \quad \forall k, (r, s) \in W \quad (5.7b)$$

$$\sum_{a \in l_k} c_a [x_a^* [v_{krs}^*]; \theta_a^0 - e_a + f_a] - \mu_{rs} = 0, \quad \forall k, (r, s) \in W \quad (5.7c)$$

$$\theta_{0a} - e_a + f_a \geq 0, \quad \forall a \in A \quad (5.7d)$$

$$e_a, f_a \geq 0, \quad \forall a \in A \quad (5.7e)$$

In this model Eq. (5.7b) represents the set of KKT conditions related to the demand constraints like the duality conditions used as constraints in the inverse LP. The optimal Lagrange multiplier μ_{rs}^* found from solving this model represents the equilibrium travel time for OD (r, s) . If there is no flow on an observed path it implies the cost $\sum_{a \in l_k} c_a [x_a^* [v_{krs}^*]; \theta_a^0 - e_a + f_a] > \mu_{rs}$, whereas flow suggests the cost $\sum_{a \in l_k} c_a [x_a^* [v_{krs}^*]; \theta_a^0 - e_a + f_a] = \mu_{rs}$ as shown in Eqs. (5.7b)–(5.7c). These two sets of constraints are equivalent to

Wardrop's user equilibrium principle as discussed in [Chapter 3](#). Eq. (5.7d) is used to regularize the model like in [Exercise 5.2](#) so that the posterior parameter remains nonnegative.

We illustrate this with the following [Exercise 5.3](#).

Exercise 5.3

Consider a simple traffic network shown in [Fig. 5.9](#) where the link performance function is the BPR function $c_a = c_a^0 \left(1 + 0.15 \left(\frac{x_a}{\theta_a} \right)^4 \right)$ for each link $a \in A$. The total demand for (O, D) is 10 units, $c_a^0 = (10, 20, 25)$, the observed path flows are $x_a = v_k = (3.583, 4.645, 1.772)$, and the prior capacity effect parameters are $\theta_a^0 = (3, 3, 3)$. Estimate θ_a to ensure the observed flows are under user equilibrium.

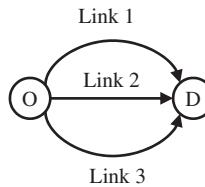


Fig. 5.9 Sample network for [Exercise 5.3](#).

Since there is flow on all three paths, the inverse problem is formulated as follows.

$$\min_{\mu, e, f} (e_1 + f_1 + e_2 + f_2 + e_3 + f_3)$$

Subject to

$$10 \left(1 + 0.15 \left(\frac{3.583}{3 - e_1 + f_1} \right)^4 \right) - \mu = 0$$

$$20 \left(1 + 0.15 \left(\frac{4.645}{3 - e_2 + f_2} \right)^4 \right) - \mu = 0$$

$$25 \left(1 + 0.15 \left(\frac{1.772}{3 - e_3 + f_3} \right)^4 \right) - \mu_{rs} = 0$$

$$3 - e_1 + f_1 \geq 0$$

$$3 - e_2 + f_2 \geq 0$$

$$3 - e_3 + f_3 \geq 0$$

$$e_1, e_2, e_3, f_1, f_2, f_3 \geq 0$$

Solving this problem in MATLAB with a commercial interior point algorithm results in $\theta_a^* = (2, 4, 3)$ which matches the actual parameters that generated these observations.

The inverse traffic assignment problem can infer parameters for infrastructure facilities where they are not readily observable. In general, link performance functions for static traffic assignment problems do not reflect realistic traffic flow characteristics with physical queueing, capacity, and spillback effects. Instead, the parameters are ambiguously defined to capture relative effects of congestion on a road. This is a well-known issue (Boyce et al., 1981) with no straightforward translation of number of links and roadway types to the practical capacity parameter. Boyce and Zhang (1997) consider three types of calibration for assignment models: using priors from other studies, solving for the constraint-based parameters, or applying a statistical estimation procedure. The calibration problem is even more significant with freight assignment because freight facilities are operated by port authorities or freight operators and therefore are less observable to public road agencies.

A case study of air freight in California was conducted by Chow et al. (2014). There were 13 major airports in California as shown in Fig. 5.10. Data from the Freight Analysis Framework, Bureau of Transportation Statistics, and Federal Aviation Administration were used to obtain observed air flows and prior capacity parameters for 2007. Using the inverse optimization, the following parameters were calibrated for the airports in 2007 in Table 5.5.

Based on the calibration, the accuracy of the resulting assignment was measured using a root mean squared error. A validation using 2010 data was also performed. A summary of the results is shown in Table 5.6. The transfer times

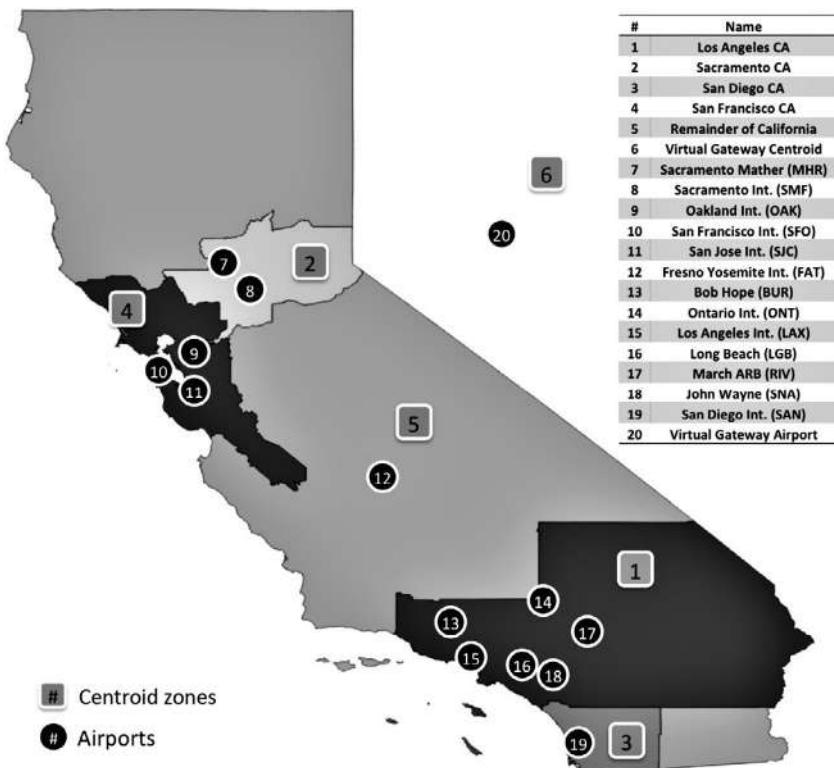


Fig. 5.10 Thirteen major airports in California. (Source: [Chow et al., 2014](#).)

fit the 4–5 h average time frame suggested by [Ohashi et al. \(2005\)](#). Due to congestion the load and unload times for the trucks delivering to the airports increased to over 2 h. The payloads and percent empty flows are reasonable values as well. The flows suggest that changes in the airline markets led to a reduction of empty flows from 22% down to 13%.

5.4 MULTIAGENT INVERSE TRANSPORTATION PROBLEMS

Up until now the IO methods involve a single model and observations of decision variable outputs of that model. In many instances, observations are not made at the aggregate level, but come from individual behavioral agent decisions. This dichotomy between a system model and learning through agent observations is an important distinction. First, IO methods require system observations, but obtaining that information from individual agents result in heterogeneous inputs. The result is that additional estimation

Table 5.5 Prior and calibrated capacity parameters for freight flows in the airports in California in 2007

Airports	Prior capacity parameter (ktons)	Calibrated capacity parameter (% +/−)
MHR	188.552	83.623 (−56%)
SMF	216.952	105.678 (−51%)
OAK	1206.393	569.303 (−53%)
SFO	162.667	256.138 (+57%)
SJC	119.722	124.030 (+4%)
FAT	20.074	39.323 (+96%)
BUR	104.518	98.166 (−6%)
ONT	1153.375	688.197 (−40%)
LAX	588.748	1196.158 (+103%)
LGB	140.010	104.493 (−25%)
RIV	17.329	56.230 (+224%)
SNA	25.390	109.096 (+330%)
SAN	102.246	208.264 (+104%)

Table 5.6 Summary of model calibration and validation

Measures	2007	2010
RMSE of inbound/outbound flows	4.631 (11.7%)	8.746 (15.0%)
Average transfer time (h)	4.116	4.067
Average load/unload time (h)	2.256	1.616
Truck vehicle volume (ktons)	1538	1256
Truck payload (commodity tons/ vehicle tons)	66.8%	67.8%
Aircraft volume (ktons)	60	44
Aircraft payload (commodity tons/ vehicle tons)	78.5%	86.8%
Truck empty %	33.2%	32.2%
Aircraft empty %	21.5%	13.2%

of system aggregation from agent observations is needed, which leads to increased inefficiencies. Second, the resulting information from agent observations may not be consistent with the system model when inferring system parameters. This limitation is evident in the IO literature, where most efforts have focused on systems with learning from the same system level observations.

Recent studies have tried to address these points by using noisy observations. [Aswani et al. \(2015\)](#) set up a bilevel problem to estimate from noisy data, and [Esfahani et al. \(2015\)](#) modeled the noisy information problem as a robust optimization model. Both approaches explain the heterogeneity with

stochastic variables to allow suboptimal observations. However, this leads to discrepancies with mechanistic assumptions in the prevailing system model (i.e., it may lead to an observation that is not optimal with respect to their own preferences). The limitation of those IO methods to using system observations is also in computational efficiency. For example, [Güler and Hamacher \(2010\)](#) proposed an IO model to infer link capacities for a minimum cost flow problem from observed flows. They conclude that the model is NP-hard.

[Chow and Recker \(2012\)](#) proposed a multiagent framework for IO where a sample of individuals' trip scheduling data is obtained and used to infer parameters of the HAPP models for every agent. Parameters of multiple agents are estimated such that the mean of their parameters is a fixed point. This leads to a learning process for heterogeneous parameters of a system model where individually calibrated optimization models correspond to observations as optimal solutions.

5.4.1 Model and Solution Method

Consider a network $G(N, A)$ that receives observations from a population P of agents behaviorally seeking to travel from an origin $r_i \in N$ to a destination $s_i \in N, \forall i \in P$ according to a shortest path in terms of additive link costs. Each agent $i \in P$ has a perception of parameters in a subnetwork $g_i \subseteq G$; these varying perceptions are reflected in heterogeneous parameters at the system level.

In the basic multiagent inverse transportation problem framework, let us assume there are no congestion or capacity effects, and only heterogeneous link costs are present. In other words, each link cost $c_a, a \in A$, is described by a distribution over P such that an agent's perceived values of c_a, i justify their revealed route choice x_i^* . Parameter learning is achieved with a set of inverse shortest path problems $\phi^{-1}(g_i, c_0, x_i^*)$, one for each agent, constrained to have an invariant common prior, as illustrated in [Fig. 5.11](#) and in Eq. (5.8), where $\phi[g_i, c_i]$ is a shortest path problem with subgraph g_i and perceived link costs c_i .

$$\min_{c_0, c_i} \left\{ |c_0 - c_i| : x_i^* = \operatorname{argmin}_x \phi[g_i, c_i] \right\}, \quad \forall i \in P \quad (5.8a)$$

Subject to

$$c_0 = \frac{1}{|P|} \sum_{i \in P} c_i \quad (5.8b)$$

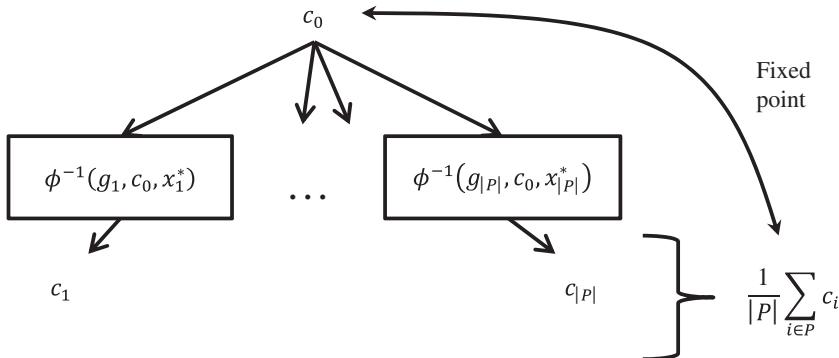


Fig. 5.11 Multiagent inverse optimization as a fixed-point problem.

The constraint in Eq. (5.8b) interprets the prior as a *common prior* among the population. The common prior serves to regularize the estimation of the parameters across the population to minimize variance. The term common prior was first defined by Harsanyi (1967) for Bayesian games with incomplete information.

Definition 5.1 A **common prior** is an assumption that multiple players in a Bayesian game with incomplete information exhibit prior beliefs that are drawn from a commonly observed source of information, that is, there exists “common knowledge” that everyone knows, and everyone know that everyone knows it.

Feinberg (2000) showed if players do not agree to disagree then a common prior always exists and is unique. This assumption is held for these multiagent inverse transportation problems and allows for formulating the fixed point to relate the common prior to the posteriors.

The common prior is a latent vector for some parameters and observable for others. For example, the common prior for goal arrival times is observable as presented by Table 5.4. For latent common priors, Eq. (5.8b) is used to estimate the vector as a fixed point. The fixed point in Eq. (5.8) can be reached by any convergent iterative algorithm. The Method of Successive Averages from Chapter 3 is considered here.

Algorithm 5.2 is convergent as shown in Theorem 5.1.

Theorem 5.1 (Chow and Recker, 2012). *The Method of Successive Averages applied to Eq. (5.8), where $\phi[g_i, c_i]$ is a mathematical programming problem, converges to a unique invariant common prior.*

Algorithm 5.2: (Xu et al., 2017). MSA-Based Algorithm to Solve Eq. (5.8) for a Population With Latent Common Prior Vector

Inputs: an initial common prior c_0 , subgraphs for every agent in the population g_i , stopping criterion

0. Initialize $c_0^1 = c_0$ and set $n = 1$.
1. For each agent $i \in P$, solve an inverse shortest path problem $c_i^n = \phi^{-1}(g_i, c_0^n, x_i^*)$.
2. Set average to $\mu^n = \frac{1}{|P|} \sum_{i \in P} c_i^n$.
3. Update common prior: $c_0^{n+1} = \frac{n}{n+1} c_0^n + \frac{1}{n+1} \mu^n$. Set $n = n + 1$ and go to step 1 if stopping criterion not reached.

Outputs: Invariant common prior c_0^* , optimal posterior c_i^* for every agent i for which Eq. (5.8b) is satisfied

Proof Since the prior represents a set of objective weights and the weights are finite and relative, there always exists a bounded range of parameters for which Eq. (5.8b) is satisfied. MSA exhibits the property of Law of Large Numbers and will always converge to an expected point given a bounded range. Since the common prior always exists and is unique, and is defined as the set of beliefs that converge upon the expected values of the population, this is a sufficient condition for convergence to the invariant common prior. ▀

Let us graphically illustrate this convergent property of [Algorithm 5.2](#) with [Fig. 5.12](#). In the figure, we show a one-dimensional parameter (relative to a second parameter). Suppose there are six agents whose parameters fall into the fixed ranges represented by each bar. As this illustration shows, the

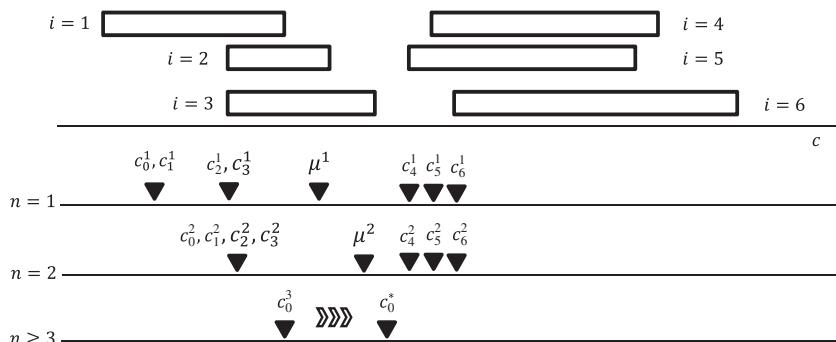


Fig. 5.12 Illustration of [Algorithm 5.2](#).

estimated values of c_4, c_5, c_6 will remain the same every iteration since the prior is coming from the left. As a result, the MSA will eventually converge to a fixed point somewhere between these values and the right edge of $i=1$.

Now that we have an algorithm to estimate parameters and invariant common prior, let us reexamine the example from [Section 4.5](#) involving the urban truck deliveries shown in [Fig. 4.17](#) taken from [You et al. \(2016\)](#). There are two classes of truck agents and for a given prior the optimal parameters for the inverse optimization can be obtained by inspection. The results of the estimated values and MSA updates for the first six iterations are presented in [Table 5.7](#).

In each iteration, each class finds an inverse optimal solution based on perturbation from the posterior of the previous iteration as the current prior to replicate the observed patterns. For example, in iteration 1 with prior $(1, 1, 1, 1)$, class 1 inverse optimal parameters are $(1, 1, 1, 0.011)$, while for class 2 they are $(15, 0, 15, 0)$. The weighted average ($3/5$ for class 1, $2/5$ for class 2) is then $\mu^1 = (6.6, 0.6, 6.6, 0.007)$. The first iteration MSA involves taking $1/2$ of the weighted average and the prior, resulting in $\hat{c}_0^2 = (3.8, 0.8, 3.8, 0.503)$.

The fixed points for both classes coincide at $(15, 0, 15, 0)$, which makes sense because at those values the objective values for both classes are equal, and hence variance is minimized at zero. The sampled sequences can then be aggregated into a time-of-day seed OD matrix from which iterative proportional fitting can be applied to expand the matrix to the population level.

Elasticities and trade-offs for the population can then be made. For example, increasing 15 min of early schedule arrival corresponds to an increase of 1 visit to zone 2 or zone 4. The start times are bound to the schedule constraints of each firm. Furthermore, trips between zones are directly dependent on one another for each class, as tours can be distinguished.

Compared to models in the literature, the proposed method allows policymakers to assess the impacts of policies like PierPass or off-peak deliveries (simply as additional schedule constraints on the calibrated operators), effects of additional congestion from commuters as modifiers to the travel time/cost parameters, and additional facility investments as either changes in utilities of existing zones or additional candidate zones.

5.4.2 Comparing Methodology Against Mixed Logit Model

As discussed under [Section 4.4](#), a sample $S \subset P$ of the population may be used to infer the fixed point and distribution of the heterogeneous parameters. In such a case, the invariant common prior is a sample prior for the whole

Table 5.7 Illustration of parameter estimation using Algorithm 5.2 for example from Section 4.5

Table 5.8 Scenarios evaluated in example

No.	Scenario	Model
1	Baseline	Multinomial logit
2		Mixed multinomial logit
3		Shortest path problems calibrated with Algorithm 5.2
4	Link 3 removed	Multinomial logit
5		Shortest path problems calibrated with Algorithm 5.2

population and the posteriors obtained for every agent is a sample of a distribution representing the whole population. The effectiveness of this approach is demonstrated in a series of examples from [Xu et al. \(2017\)](#) in which agent route choice is modeled using this multiagent approach and compared to a route choice model based on discrete choice.

There are two objectives. The first is to illustrate the capability of the proposed method to capture heterogeneity of users' preferences at one network level (*link* costs) even when observations are made at another level (*route* choice). This objective is achieved by using a simple network with enumerated paths and simulated link costs that vary across the population. These link cost variations reflect different traffic and environmental conditions (e.g., weather and road surface conditions) present during each user's trip, while observable route choices may be obtained from GPS, phone, or transit smart card data (to varying degrees). Link cost heterogeneity is reflected in distributions of the link costs across the population.

The second objective is to demonstrate how the method can better handle structural changes in the underlying network. This is accomplished by applying the estimated models on a scenario where one of the links is removed.

To give the results more context, we estimate two discrete choice models: an aggregate multinomial logit model for route choice, and a mixed

Exercise 5.4

Consider a network shown in [Fig. 5.13](#) labeled with five links and four nodes, where the perceived link costs of 500 agents traveling from node 1 to node 4 are simulated as shown in Test Set 1 at the following link: <https://github.com/BUILTNYU/Network-learning-via-multi-agent-inverse-transportation-problems>. The perceived link costs are assumed to be independent of each other. Based on the simulated perceived costs, the observed routes are shared with a researcher. An initial common prior of $c_0 = 0.5$ is assumed for all links. Estimate the parameters using the multiagent

inverse shortest path problem and compare the procedure with the multinomial and mixed logit models of route choice in accurately estimating the route flows.

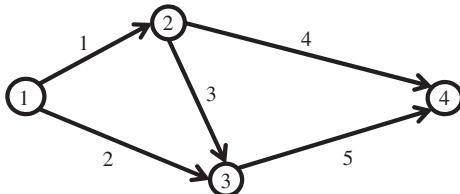


Fig. 5.13 Test network with node and link IDs.

There are three paths in this example as indicated by the following link sequences: (1, 4), (2, 5), (1, 3, 5). Based on the simulated link costs of the 500 samples we observe 48% choose path (1, 4), 48% choose path (2, 5), and the remaining 4% choose (1, 3, 5). For [Algorithm 5.2](#), a tolerance of 0.001 of the maximum change in the common prior is used as the stopping condition. Since there are three routes, there are only two degrees of freedom for link costs to vary, so we do not expect estimated distributions to reflect more than two discrete alternatives.

For the aggregate multinomial and mixed logit models, the utility functions are based on route costs to be consistent with the route choices. For the logit models, the average path costs are assumed to be known as the explanatory path cost variable X_j for each alternative j . $U_j = \beta_j X_j + e_j$ is an aggregate utility function that is dependent only on the same average path cost variables for everyone. X_2 is set to be the utility of path 2 (2, 5) relative to path 1 (1, 4): $X_2 = c_2 + c_5 - c_1 - c_4$, while X_3 is the utility of path 3 (1, 3, 5) relative to path 1: $X_3 = c_3 + c_5 - c_4$. In this mixed logit model, the β_j is normally distributed.

We first estimated the parameters from the data using the multiagent IO. [Algorithm 5.2](#) was employed with the convergence shown in [Fig. 5.14](#). Based on a tolerance of 0.001, the algorithm terminated after 22 iterations of the MSA.

[Fig. 5.15](#) illustrates how the multiagent IO outputs a distribution of link costs across the population based on observation of their route choices and the reliance on the normative route choice behavior in the inverse transportation problem. The values are $\{0.489, 0.498, 0.009, (0.490, 0.493), (0.481, 0.484)\}$ corresponding to links 1–5. The link costs ended up being homogeneous for the first three links and split over two different values for the remaining two links. This reflects how even a network with only two degrees of freedom in information can lead to an estimation of heterogeneous link costs.

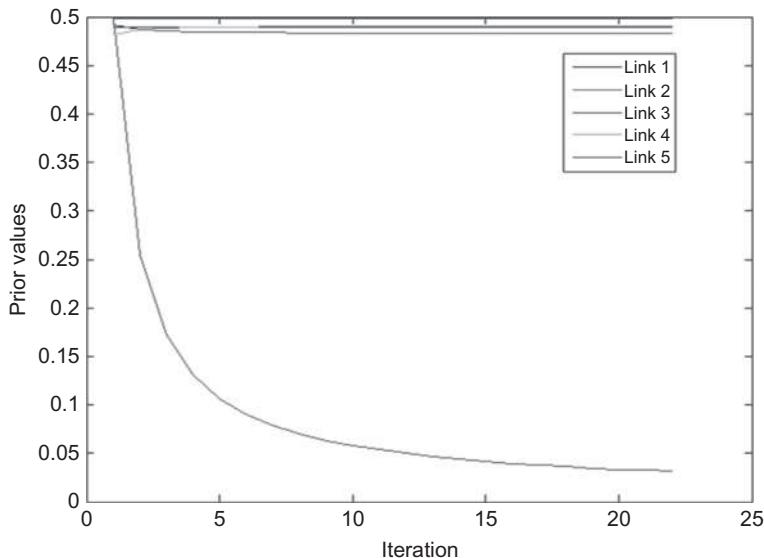


Fig. 5.14 Convergence of Algorithm 5.2 on test network. (Source: Xu et al., 2017.)

For context, the route choices are modeled using multinomial and mixed logit models in *R* using the average route costs as the explanatory variables. The estimated multinomial logit model has a log-likelihood value of -417.24 . The McFadden R^2 (ρ^2) value is 0.240 . For the mixed logit model, a sampling of 100 simulated Halton draws was used to obtain the results. Using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method, the algorithm converged to an estimate after four iterations. For the mixed logit model, the log-likelihood value is $LL = -417.23$ and $\rho^2 = 0.240$. The estimated coefficients are presented in Table 5.9.

Since the two networks do not have the same link cost distributions, a direct comparison of the results is not expected. However, the results clarify the value of the multiagent inverse transportation problems when interpreted alongside one another.

- While the multiagent IO method endogenously obtained the average link costs, the discrete choice models required prior information about the average path costs in order to be estimated.
- The discrete choice models clearly do not provide estimates of link-level parameters, much less link-level heterogeneity.
- The estimated results suggest that the standard deviations of the mixed logit model (and hence the distribution assumption for taste variation in path costs) are statistically insignificant (*t*-stat of 0.0081). Despite there

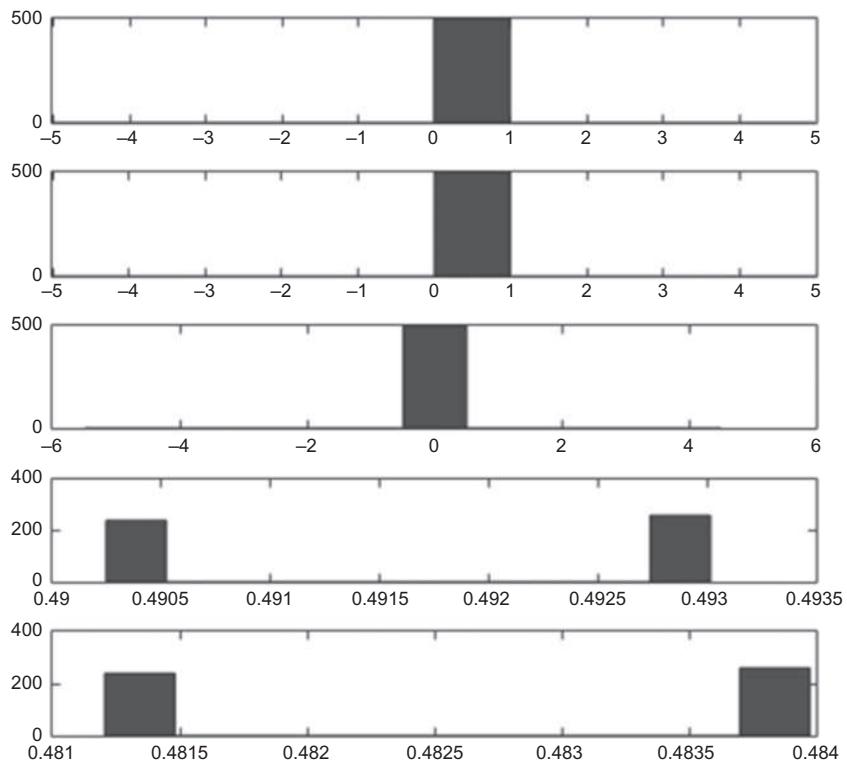


Fig. 5.15 Output distribution of posterior link costs across the population of 500 simulated agents for link 1 (top) to link 5 (bottom). (Source: Xu et al., 2017.)

Table 5.9 Estimated parameters and significance tests for multinomial and mixed multinomial logit model

Variable	Estimate	Standard error	t-Statistic
Multinomial logit			
X	-4.93040	0.44997	-10.957***
Mixed logit			
X	-4.93848	2.08812	-2.3650*
sd. X	0.18917	23.22305	0.0081

*P-value < 0.05.

***P-value < 0.001.

being path-level variation in perceived costs, it is difficult to capture this heterogeneity using the mixed logit model for this example.

- The multiagent IO correctly fits agent route choice to obtain 100% fit to the data of 48% for path 1, 48% for path 2, and 4% for path 3. On the other hand, the estimated shares from MNL are 50.2% for path 1, 45.7% for path 2, and 4.1% for path 3.
-
-

Exercise 5.5

For the same network in Exercise 5.4, compare the prediction of route choices using the multiagent IO and the multinomial logit model when link 3 is closed.

Under the new scenario, the estimated models are applied to validate their accuracy in terms of total route shares compared to the simulated ground truth. When link 3 is closed, the alternative path 3 no longer exists, and there are only two routes left to choose from. Under these scenarios, the simulated observed routes show that 50% of the travelers take path 1.

The shortest path assignment using the link costs estimated with the multiagent inverse optimization indicates with 100% fit the optimality of the observed choices. For context, the statistical models show some error as reported in Table 5.10.

Table 5.10 Estimated shares (MNL) vs actual shares of route choices when link 3 is closed

Alternatives	Estimated shares	Actual shares	Error
Path 1	0.524	0.5	0.024
Path 2	0.476	0.5	0.024

multinomial logit model that allows distributions in the path cost taste parameter. In total, the following scenarios presented in Table 5.8 are evaluated in Exercises 5.4 and 5.5. Parameter estimation is run for the first three scenarios.

5.4.3 Case Study: California Household Travel Survey

How well does the multiagent IO method fit to real data? This question is addressed with a continuation of the case study of southern California household travel survey data from Chow and Recker (2012) shown in Fig. 5.5. The earlier example only considered the IO fitting of one household and did not address the ability to fit objective coefficients and goal arrival times of a population of households jointly with their latent objective common priors. We extract 78 household samples from the data that

conducted four activities or less. Out of these 78 households, 65 are single-vehicle households while 13 are two-vehicle households. Carpooling is ignored in this example. These samples are used to compare the goodness of fit of the multiagent IO for the HAPP model.

To measure how well each inverse model fits the observed activity schedule data, two goodness-of-fit measures are defined. Both measures represent the ratio of the squared error between inverse estimated values against observed patterns, to the squared error of the HAPP model with uninformed priors (e.g., $c_0^0 \equiv 1$) against observed patterns. For example, a measure of 0.5 suggests that the multiagent IO model parameters are twice as accurate as a model with uninformed priors. The first measure shown in Eq. (5.9a) is based on matching aggregated OD trip patterns (ρ_{OD}^2) while the second shown in Eq. (5.9b) is based on matching activity arrival time patterns (ρ_T^2). Note that P^+ is the set of activities (excluding drop-off at home) whereas P is the population of agents.

$$\rho_{OD}^2 = 1 - \frac{\sum_{(u, w)} \left(\sum_{i \in P} X_{uw, i}^{obs} - \sum_{i \in P} X_{uw, i}^{HAPP[c_i^*]} \right)^2}{\sum_{(u, w)} \left(\sum_{i \in P} X_{uw, i}^{obs} - \sum_{i \in P} X_{uw, i}^{HAPP[c_0^0]} \right)^2} \quad (5.9a)$$

$$\rho_T^2 = 1 - \frac{\sum_{i \in P} \left(\sum_{u \in P^+} \left(T_{i, u}^{obs} - T_{i, u}^{HAPP[c_i^*]} \right) \right)^2}{\sum_{i \in P} \left(\sum_{u \in P^+} \left(T_{i, u}^{obs} - T_{i, u}^{HAPP[c_0^0]} \right) \right)^2} \quad (5.9b)$$

Table 5.11 Comparison of uninformed prior vs optimal invariant common prior

	Travel time	Return home delay	Length of day
Uninformed prior (c_0^0)	1	1	1
Mean of posterior (\bar{c}^1)	1.0752	0.6995	0.1972
Std. dev. posterior (s. d. (c^1))	0.6064	0.4562	0.3449
Invariant common prior (c_0^*)	1.2287	0.2715	0.0598
Mean of posterior (\bar{c}^*)	1.2831	0.2059	0.0606
Std. dev. posterior (s. d. (c^*))	0.4421 (-27%)	0.1151 (-75%)	0.1631 (-53%)

Table 5.12 Comparison of performance measures

	SSE: HAPP[c ⁰ ₀]	SSE: HAPP[c ¹]	SSE: HAPP[c*]	$\rho^2[c_0^0]$	$\rho^2[c^1]$	$\rho^2[c^*]$
OD	128	8	10	0	0.938	0.922
T	6.5813×10^7	2.9052×10^7	4.3910×10^5	0	0.559	0.993

Three objective coefficients are estimated: travel time, return home delay, and length of day outside home. The performance of the estimated HAPP models for the 78 households is summarized in [Tables 5.11](#) and [5.12](#). [Table 5.11](#) shows the parameters of the objective coefficients based on perturbing them from an uninformed prior compared to those that are perturbed from the invariant common prior found using [Algorithm 5.2](#). The variance of the estimated parameters across the population drops significantly due to finding the invariant common prior to perturb from.

[Table 5.12](#) presents the computed performance measures from Eq. (5.9). The HAPP model with posterior parameters c^* is compared to a base HAPP model with completely uninformed priors as the parameters (c_0^0) and a HAPP model with posterior parameters perturbed from that uninformed prior (c^1). These results indicate the model with both calibrated common prior and posteriors performs accurately on both space and time dimensions.

5.5 NETWORK LEARNING

In this last section, the multiagent IO learning framework is applied to infer latent network attributes from which only a sample of agent route observations is available. One major advantage of this approach is that leveraging the information from agent behavior reduces the reliance on having to estimate the population data like conventional methods.

5.5.1 Methodology

Consider a capacitated multicommodity flow problem in Eqs. [\(5.10a\)](#)–[\(5.10d\)](#), where M is the set of commodities and $u = \{u_{a \in A}\}$ is a vector of capacity constraints for a subset of links in the network.

$$\min_x \sum_m c^T x_m \quad (5.10a)$$

Subject to

$$Ax_m = b_m, \forall m \in M \quad (5.10b)$$

$$\sum_{m \in M} x_m \leq u \quad (5.10c)$$

$$x_m \geq 0, \forall m \in M \quad (5.10d)$$

Eq. (5.10b) is flow conservation constraints for each commodity $m \in M$. Eq. (5.10c) is the set of capacity constraints for the vector of link flows bundled from the individual commodities. The inverse problem to infer the values of u from observed x_m and other network parameters is NP-hard (Güler and Hamacher, 2010). Instead of tackling this inverse problem directly, we seek dual prices w corresponding to the constraints in Eq. (5.10c). A value of $w_a = 0$ means that a link $a \in A$ is not operating at capacity u_a , while $w_a > 0$ reflects the impact of a binding capacity on agents' route choices. In other words, we do not concern ourselves with finding capacity, but instead with finding the effects of the capacity and its interaction with the agents.

In this problem, we assume link costs are not heterogeneous and are known in advance. Each agent has a perceived value of the dual price of the capacitated links. The capacitated problem can be decomposed into a master problem for determining optimal dual prices and unconstrained subproblems for each commodity. The dual price is reflected within each agent's shortest path problem through the Partial Dualization Theorem (Ahuja et al., 1993): the w corresponding to Eq. (5.10c) in the multicommodity flow problem is equivalent to a w for the uncapacitated shortest path problem of each agent $i \in P$ as shown in Eq. (5.11).

$$\min_{x_i} (c + w)^T x_i \quad (5.11)$$

By relying on this relationship, we introduce a multiagent inverse transportation problem to infer the network dual prices. Each agent solves an IO where there is a common prior dual price vector w_0 . We define two non-negative decision variables e_i and f_i for each agent such that $w_0 - w_i = e_i - f_i$, and solve Eqs. (5.12a)–(5.12e) for each agent subject to Eq. (5.12f) for all agents. In other words, route dependencies are captured by bundle constraints such as capacity (Eq. 5.10c). With decomposition, the original problem is decomposed into individual shortest path problems where the costs in the objective are updated to reflect the dual price obtained from the restricted master problem (Eq. 5.11). In the inverse problem, the requirement for a common prior (Eq. 5.12f) ensures the solution fits the bundling constraints.

$$\min_{y_i, e_i, f_i} e_i + f_i, \forall i \in P \quad (5.12a)$$

Subject to

$$A^T \gamma_i \geq c + w_0 - e_i + f_i, \quad \forall i \in P \quad (5.12b)$$

$$b^T \gamma_i = (c + w_0 - e_i + f_i)^T x_i^*, \quad \forall i \in P \quad (5.12c)$$

$$w_0 - e_i + f_i \geq 0, \quad \forall i \in P \quad (5.12d)$$

$$e_i, f_i \geq 0, \quad \forall i \in P \quad (5.12e)$$

$$w_0 = \frac{1}{|P|} \sum_{i \in P} w_i \quad (5.12f)$$

Eqs. (5.12a)–(5.12c) are the standard inverse LP equations. Eq. (5.12d) is the nonnegativity constraint for the estimated posterior dual price. In the case of taking the inverse shortest path, there are equality constraints so the dual variables for that problem are unbounded. The following theorems can be made.

Theorem 5.2 (Xu et al., 2017). *The problem shown in Eqs. (5.12a)–(5.12f) has a unique solution in a common prior dual price vector for all capacitated links, and this vector is the same for all agents, i.e., $w_0 = w_i \forall i \in P$.*

Proof A multicommodity flow problem solution has a unique set of dual prices (Ahuja et al., 1993). This homogeneity occurs because the dual price is a lower bound threshold for everyone, and the highest value price is the one kept. This can be illustrated with two agents A and B sharing a link a . Suppose agent A would leave link a if the dual price was w_A . This means any value of $w \geq w_A$ would incentivize agent A to leave link a . Now suppose agent B has a dual price of $w_B > w_A$. Any common prior price $w_A \leq w_0 < w_B$ would not be fixed, because agent B would perturb up toward w_B while agent A would be indifferent, until the common prior and final prices become fixed at $w_0 = w_B$, and both agent A and B share the same w_B . ■

Theorem 5.3 (Xu et al., 2017). *The unique inverse optimal parameters to Eqs. (5.12a)–(5.12f) can be reached by starting with an initial guess at $w_0^1 = 0$ and then following a basic iterative update of $w_0^{n+1} := \frac{1}{|P|} \sum_{i \in P} w_i^n$ without having to resort to convergent algorithms like MSA.*

Proof Since $w_0^1 = 0$ represents the lower boundary, in each iteration n the updated average of w_i^n would always be increasing due to the lower threshold condition explained in the Proposition 1 proof. This means a basic iterative update of letting $w_0^{n+1} := \frac{1}{|P|} \sum_{i \in P} w_i^n$ is monotonically increasing. Therefore it is guaranteed to reach the unique solution. ■

Let us call the variant Algorithm 5.2 with the $w_0^{n+1} := \frac{1}{|P|} \sum_{i \in P} w_i^n$ update as Algorithm 5.2A.

Theorem 5.4 (Xu et al., 2017). *The unique inverse optimal parameters to Eqs. (5.12a)–(5.12f) can be reached in polynomial time using the basic iterative update from Algorithm 5.2A.*

Proof Each run of the agent IO problem is an LP which is polynomial time solvable. The number of iterations of the iterative update is finite. This can be shown in a worst-case scenario; suppose out of $|P|$ agents, $|P|-1$ of them all exhibit dual price of 0 for a particular link while one agent i has a dual price of $w_i > 0$. In this case, in each iteration all the $|P|-1$ agents would keep setting the w 's to 0 and agent i 's to w_i . This means in the worst case the average will always be increasing by $\frac{w_i}{|P|}$ as a finite step size until the optimum is reached. ■

Exercise 5.6

Consider three link flows observed in the network in Fig. 5.16, $x = \{100, 200, 100\}$. We can assume there are three groups of homogeneous agents, agent group 1 choosing link 1, agent group 2 choosing link 2, and agent group 3 choosing link 3. Each agent group seeks a dual price to explain their link choice, resulting in nine values of w (for each agent and each link), and three values of w_0 . Illustrate two iterations of Algorithm 5.2A.

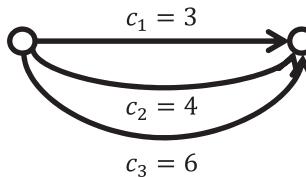


Fig. 5.16 Toy network used for illustrating methodology.

The algorithm is initiated by setting $w_0^1 = \{0, 0, 0\}$. An inverse shortest path problem is run for each agent. For agent group 1, $w_1^1 = \{0, 0, 0\}$ because they are already traveling on the shortest path with dual prices at zero. For agent group 2 to choose link 2, a value of $w_2^1 = \{1, 0, 0\}$ is needed. Lastly for agent group 3 to choose link 3, a value of $w_3^1 = \{3, 2, 0\}$ is needed. At the end of this iteration, the weighted average of the three agents is taken as the new prior: $w_0^2 = \left\{ \frac{200+300}{400}, \frac{200}{400}, 0 \right\} = \left\{ \frac{5}{4}, \frac{1}{2}, 0 \right\}$. If this is advanced a second iteration, we get $w_1^2 = \left\{ \frac{5}{4}, \frac{1}{2}, 0 \right\}$, $w_2^2 = \left\{ \frac{3}{2}, \frac{1}{2}, 0 \right\}$, and $w_3^2 = \{3, 2, 0\}$. These lead to a new prior $w_0^3 = \left\{ \frac{(125+300+300)}{400}, \frac{50+100+200}{400}, 0 \right\} = \left\{ \frac{29}{16}, \frac{7}{8}, 0 \right\}$. By inspection we can see that the dual prices approach $w_0^* = w_1^* = w_2^* = w_3^* = \{3, 2, 0\}$.

These properties of the methodology signify the effectiveness of using agent observations to learn network parameters. [Exercise 5.6](#) illustrates the properties of the method.

5.5.2 Example of Nguyen-Dupuis Network

In this example, we conduct a parameter recovery test using the [Nguyen-Dupuis \(1984\)](#) network shown in [Fig. 5.17](#) to illustrate the effectiveness of the method. The standard demand and link cost parameters from the Nguyen-Dupuis network are assumed: 400 travelers for OD (1, 2), 600 travelers for OD (4, 2), 800 travelers for OD (1, 3), and 200 travelers for OD (4, 3). By design, the paths in the Nguyen-Dupuis network can be easily enumerated. Initial capacities of 400 at link 1 and 800 at link 7 are assumed.

We generate a set of simulated route choice observations. Paths are randomly drawn with probability equal to the percent flow on that path from the solution to the multicommodity flow problem. The data set is fully accessible as *Test Set 3* on <https://github.com/BUILTNYU/Network-learning-via-multi-agent-inverse-transportation-problems>. Although the multicommodity flow problem may require an integer solution, an LP-relaxed solution is obtained in this case, revealing dual prices of $w_1^* = 7$ and $w_7^* = 5$. We show how the methodology can reproduce these parameters.

Based on simulated route choices, there are six distinct sets of agents. The agents are assumed to behave according to a shortest path problem with latent dual prices that we need to infer.

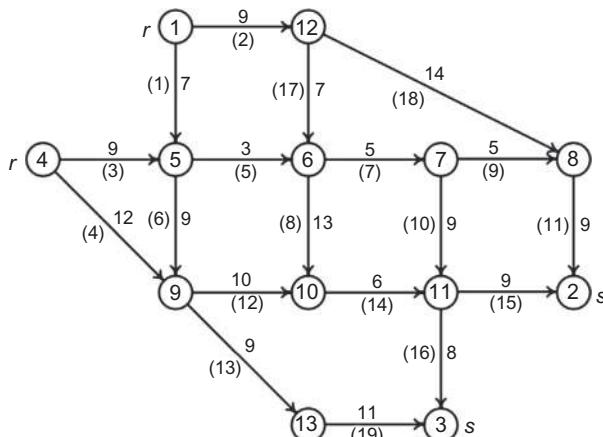


Fig. 5.17 Nguyen-Dupuis network.

The inverse shortest path problem $\phi^{-1}(g_1, w_0^1, x_1^*)$ is illustrated as follows for an agent going from node 1 to node 2 from a prior of $w_0^1 = [0, 0]$. Because the shortest path constraints are equality constraints, the dual prices here are unbounded.

$$\min \phi^{-1} = e_{1,1}^n + f_{1,1}^n + e_{1,7}^n + f_{1,7}^n$$

Subject to

$$\begin{aligned}
-\gamma_1 + \gamma_5 &\leq 7 + w_{0,1}^n - e_{1,1}^n + f_{1,1}^n \\
-\gamma_1 + \gamma_{12} &\leq 9 \\
-\gamma_4 + \gamma_5 &\leq 9 \\
-\gamma_4 + \gamma_9 &\leq 12 \\
-\gamma_5 + \gamma_6 &\leq 3 \\
-\gamma_5 + \gamma_9 &\leq 9 \\
-\gamma_6 + \gamma_7 &\leq 5 + w_{0,7}^n - e_{1,7}^n + f_{1,7}^n \\
-\gamma_6 + \gamma_{10} &\leq 13 \\
-\gamma_7 + \gamma_8 &\leq 5 \\
-\gamma_7 + \gamma_{11} &\leq 9 \\
+\gamma_2 - \gamma_8 &\leq 9 \\
-\gamma_9 + \gamma_{10} &\leq 10 \\
-\gamma_9 + \gamma_{13} &\leq 9 \\
-\gamma_{10} + \gamma_{11} &\leq 6 \\
+\gamma_2 - \gamma_{11} &\leq 9 \\
\gamma_3 - \gamma_{11} &\leq 8 \\
\gamma_6 - \gamma_{12} &\leq 7 \\
\gamma_8 - \gamma_{12} &\leq 14 \\
\gamma_3 - \gamma_{13} &\leq 11 \\
e_{1,1}^n - f_{1,1}^n &\leq w_{0,1}^n \\
e_{1,7}^n - f_{1,7}^n &\leq w_{0,7}^n \\
-400\gamma_1 + 400\gamma_2 &= 400(9 + 9 + 14) \\
e, f &\geq 0
\end{aligned}$$

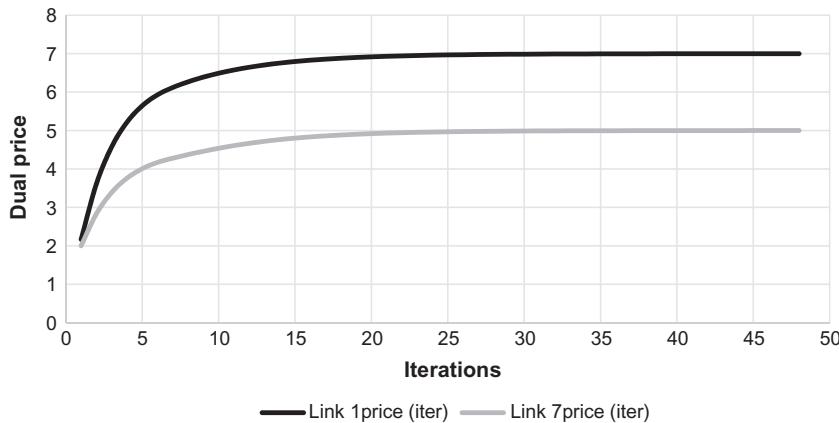


Fig. 5.18 Convergence of dual prices using Algorithm 5.2A. (Source: Xu et al., 2017.)

Based on running Algorithm 5.2A we get the following convergence of the dual price parameters for those two links shown in Fig. 5.18.

The methodology can be applied to network monitoring in an online learning setting. Data is received in real time from one agent at a time (with equal time intervals; conversely, we can view this as a system monitor drawing a sample with a steady frequency). After each agent observation, an update is conducted of the network dual price parameters to learn of any changes in the network.

To illustrate this, we assume the first 100 observations of the network operate under the initial state with unobservable link 7 capacity of 800. After 100 observations, the link 7 capacity drops to 500 for the next 100 observations. In this state, the hidden dual prices are found to be $w_1^{**}=7$ and $w_7^{**}=6$. Finally, for the last 100 observations the capacity returns to 800. The data is summarized in Fig. 5.19 and accessible on the same GitHub link as *Test Set 4*. The boundaries where the link capacity changes state are shown as vertical gray lines while the observations of path choice are shown as the points.

In this case, each update uses the posterior of the previous update as its prior for the dual prices. We get the following trajectory of the posterior dual prices shown in Fig. 5.20 as an example of how the monitored dual prices change over the 300 sequential observations.

The result shows that the proposed method is indeed sensitive to regime changes in this example, even as there is a learning period after each state change. The learning rate depends on the likelihood of the right observation that comes along to reveal the need for a change. For example, the change to

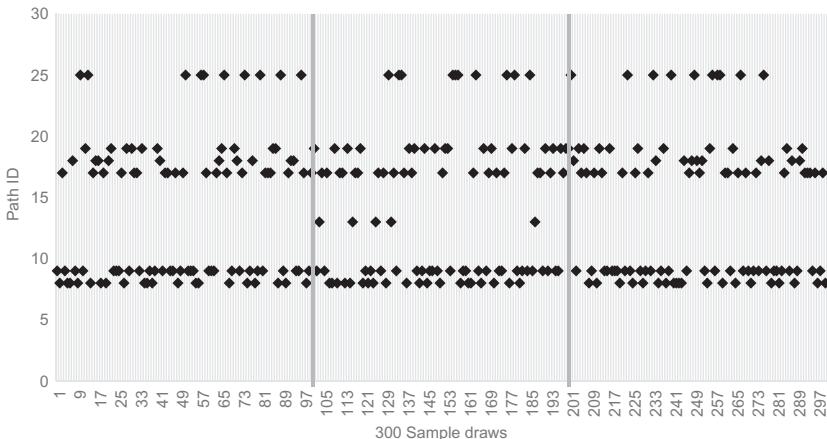


Fig. 5.19 Trajectory of simulated route observations in an online learning setting. (Source: Xu et al., 2017.)

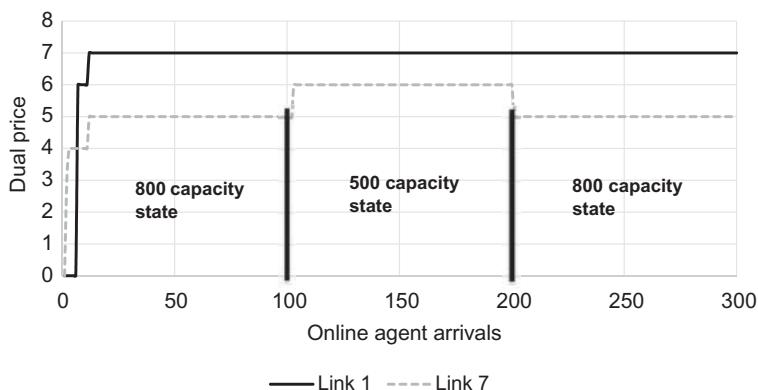


Fig. 5.20 Dual price trajectories based on 300 simulated agent arrivals. (Source: Xu et al., 2017.)

the 500 capacity state does not impact the monitoring of the dual price immediately. It is not until a new route observation of path 13, indicating a detour in route because of the decreased capacity, does the dual price shift. As a result, the sampling rate is important. The routes are also important. In this case, the monitoring system is able to detect a shift back and forth because the 500 capacity state leads to a different set of routes than the 800 capacity state. If the routes remain the same, no change may be detected.

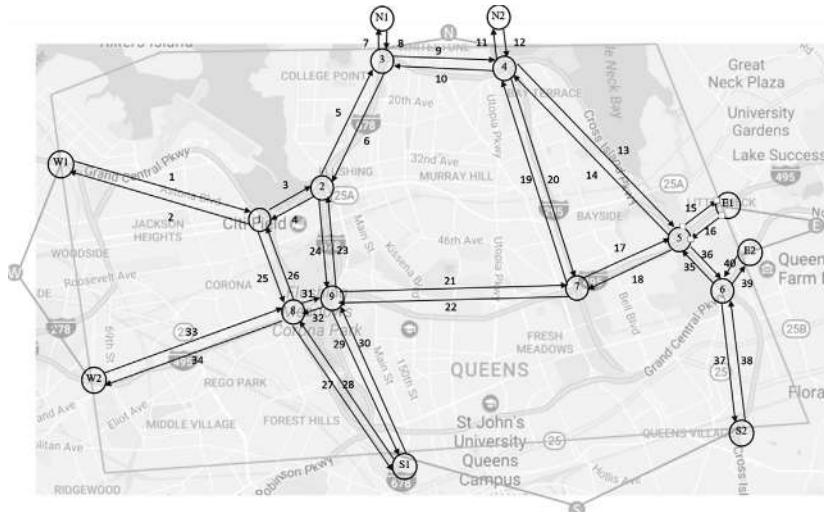


Fig. 5.21 Queens freeway network. (Source: Xu et al., 2017.)

5.5.3 Case Study: Google Maps Queries in Queens, New York

We illustrate network learning using a real data example from Xu et al. (2017). A highway network from Queens in New York City is shown in Fig. 5.21 overlaid upon a Google Maps image. The network is designed to have two entries/exits for each of the four cardinal directions. On June 5, 2017, a series of shortest paths were queried from Google Maps API based on Google's real-time travel times.

As congestion occurred in the network, the effects of the capacity on shifting routes were recognized by the network learning algorithm. The dual prices reflect links that became more congested with binding capacity effects that resulted in route diversions as shown in Fig. 5.22. The magnitudes of the dual prices give a relative measure of the insufficient capacity in the link with respect to other links. For a qualitative comparison, snapshots of the shortest paths found in Google Maps were made every half hour in Fig. 5.23 and compared to a snapshot of the multiagent IO output dual prices in Fig. 5.24.

Conclusions are drawn from this illustrative example.

- Network system attributes like link dual prices can be updated using only samples of individual route observations, *without need to estimate total link or path flows*. This demonstrates the significance of this methodology in being able to cheaply monitor a transportation network's system performance over time.

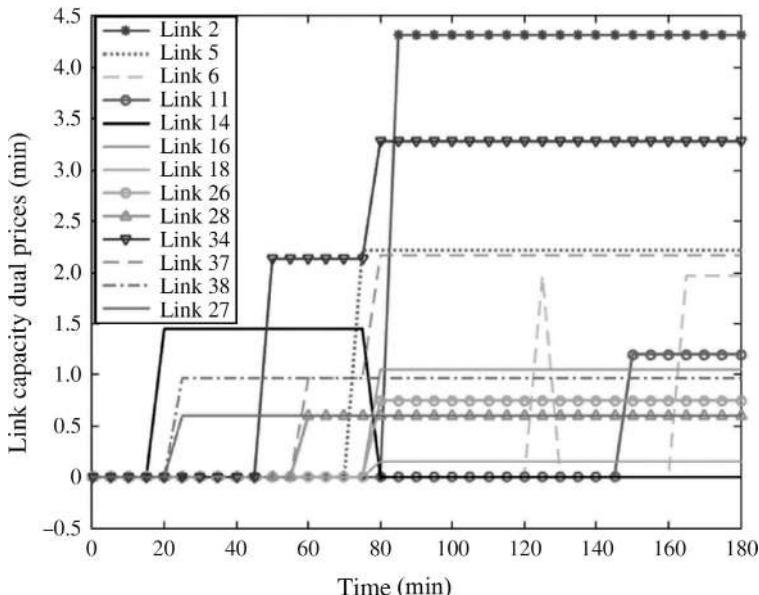


Fig. 5.22 Link dual price trajectories obtained from online network learning. (Source: Xu et al., 2017.)

- The changes show that the inference model is indeed sensitive to changes in the system. As traffic increases from 6:30 a.m. to 9:30 a.m. in the study period resulting in more spillbacks and incidents impacting link capacities, the set of dual prices steadily increases on average as shown in Fig. 5.22. A visual comparison between Figs. 5.23 and 5.24 indicates similarities in positive dual prices where congestion occurs. For example, the 7:00 a.m. screenshot shows that the segment between nodes 4 and 5 is highly congested and that is interpreted correctly in Fig. 5.24. The 7:30 a.m. screenshot reveals the alternative path traversing the link between nodes 8 and S1 is congested, which is captured correctly in the inference model. The 8:30 a.m. screenshot indicates congestion between nodes 5 and 7, which is also captured by the model. This delay lingers through 9:30 a.m. and is properly captured as well by the inference model.

RESEARCH AND DESIGN CHALLENGES

The following are open-ended challenges that sometimes refer to real data sets. Readers are encouraged to try to tackle the problems and post their

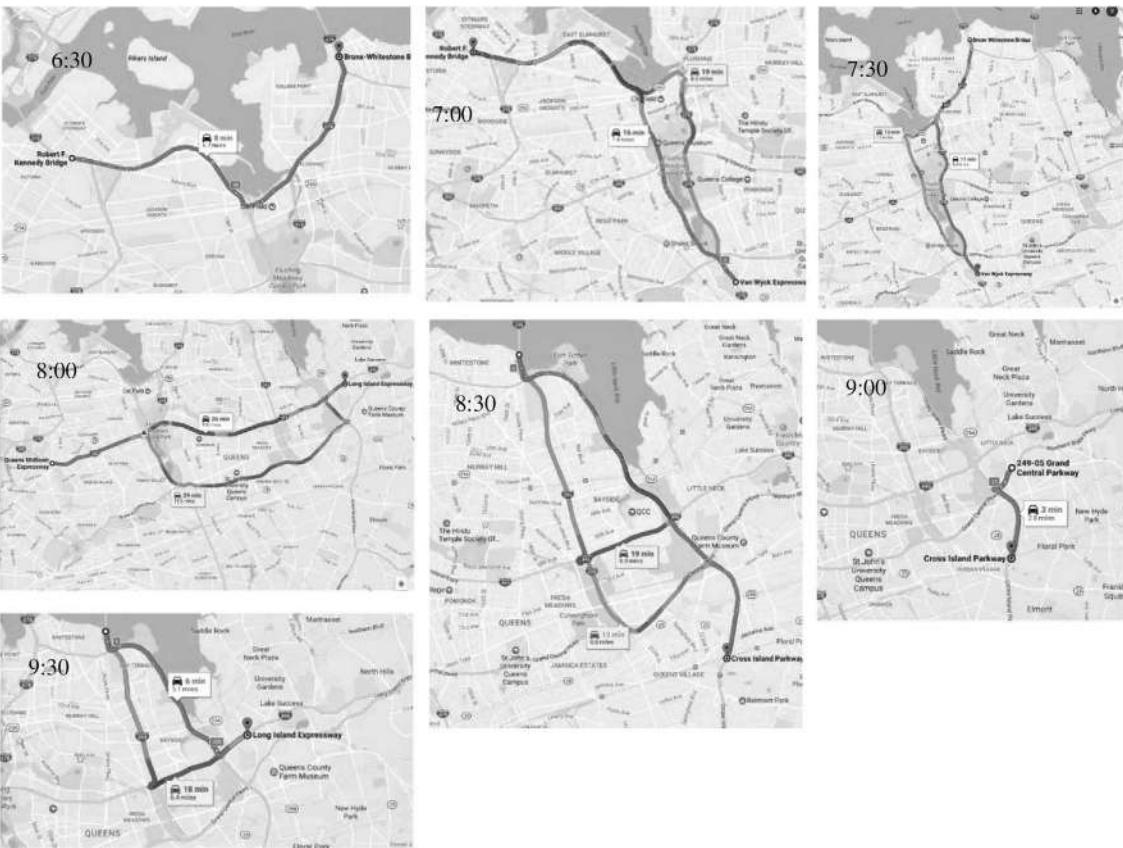


Fig. 5.23 Half-hour interval screenshots of Google Maps real-time shortest path queries (Xu et al., 2017).

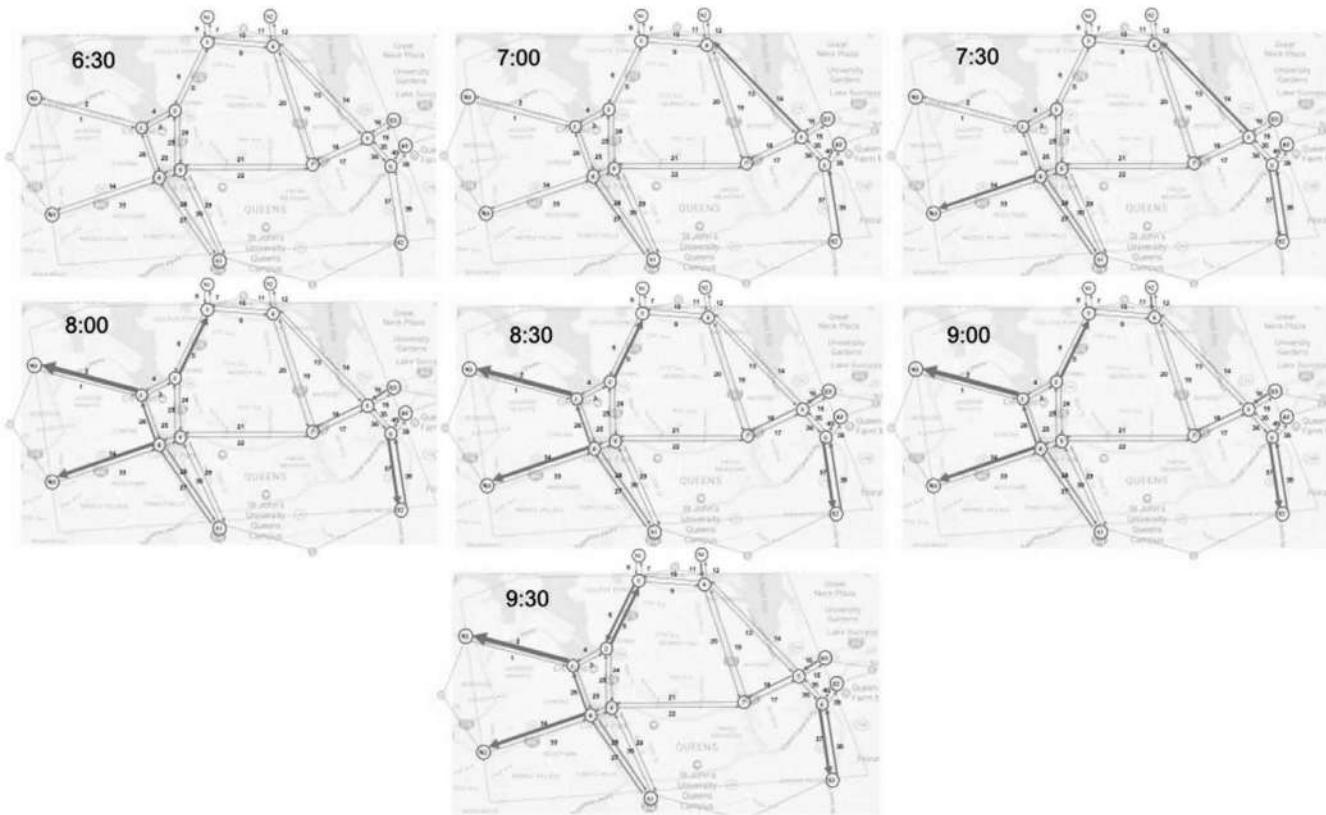


Fig. 5.24 Snapshots of multiagent IO output dual prices at every half hour with nonzero prices represented by *heavier arrows*. (Source: Xu et al., 2017.)

designs or solutions on the following GitHub page, <https://github.com/BUILTNYU/Elsevierbook/tree/master/Chapter%205>.

- (5.1) Pick two machine learning methods from [Table 5.1](#) and apply them to two different steps in the Four Step Model for a household travel survey/employment data for a region of your choice. Compare strengths and limitations of the trained models.
- (5.2) Write a program to solve an inverse LP with L_1 norm and test it on the network in [Exercise 5.1](#).
- (5.3) In the many-to-many assignment game, buyers are matched to sellers to maximize the payoff awarded as shown as follows (see [Roth & Sotomayor, 1990](#)):

$$\max \sum_{i \in P} \sum_{j \in Q} a_{ij} x_{ij}$$

Subject to

$$\sum_{i \in P} x_{ij} \leq q_j, \quad \forall j \in Q$$

$$\sum_{j \in Q} x_{ij} \leq w_i, \quad \forall i \in P$$

$$x_{ij} \in \{0, 1\}, \quad \forall j \in Q, \forall i \in P$$

where Q is the set of sellers, P is the set of buyers, a_{ij} is the payoff for a match, q_j is a maximum number of matches for seller $j \in Q$, w_i is the maximum number of matches for buyer $i \in P$, and x_{ij} is a variable indicating a match between buyer i and seller j . If $a_{ij} = \max(0, U_{ij} - c_j)$ where c_j is the cost of production, design an inverse optimization program to learn the value of U_{ij} .

- (5.4) Download a bike GPS trace from Openstreetmap (or something similar): <https://www.openstreetmap.org/user/absinthologue/traces/2651853>.

Hypothesize a set of objectives that this user might have in their route choice criteria (e.g., travel time, scenic, safety, elevation change, etc.) where the data can be obtained. Solve an inverse shortest path problem with L_1 norm from a diffuse prior to learn the objective weights revealed by the route.

- (5.5) Take the NYC yellow taxi data (http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml) and identify the idle vehicle movements. Use this data to prepare a data set of destination choice for idle vehicles. Specify a utility function with random coefficients across the population. Use Eq. (5.8) with an MSA-type algorithm like

Algorithm 5.2, starting with a diffuse common prior, to find a set of posterior values of coefficients for each taxi under L_1 norm with an invariant common prior. Compare the distribution of the coefficients with the parameters of a mixed logit model estimated using Algorithm D.2.

- (5.6) Create a simplified road network of a region of your choice in the United States such that nodes have corresponding geocodes in the region. For a weekday from 6 a.m. to 12 p.m., sample Google Maps queries of real time travel times and routes from one random node to another node in that network every 10 min. Using this information, update the dual price of the links in the simplified road network (see [Exercise 5.1](#)). Note down when Google Maps updates with incidents on the road network. Can effects of the incidents be measured in terms of the link dual prices?
- (5.7) Implement [Algorithm 5.1](#) in a program of your choice and test it using the inverse DARP in [Exercise 5.2](#).
- (5.8) Use [Algorithm 5.1](#) to solve inverse facility location problem to infer the demand h_i in the p-median problem (see Eq. 7.8) for observed Starbucks locations in a region of your choice. Use census tracts as the demand zones. Specify a linear regression model where the inverse optimal h_i are the dependent variables and infer coefficients of the variables (e.g., income level, population, population of certain age groups, number of employees of certain industries, etc.) to infer Starbucks store location strategy.
- (5.9) Download a GPS trajectory of trucks from the Shenzhen data set (<http://www-users.cs.umn.edu/~tianhe/BIGDATA/>). Fit a multiobjective single-vehicle routing problem to this data that includes arrival time windows for destinations belonging to each zone (assume all destinations in a zone are homogeneous).
- (5.10) For the inverse DARP, create random instances where the inverse DARP is solved with increasingly larger fleet size from 1 to 5 vehicles. Evaluate the performance of the inverse optimization as a function of the fleet size.
- (5.11) For the household travel survey in NYC (<https://www.nymtc.org/DATA-AND-MODELING/Travel-Surveys/2010-11-Travel-Survey>), sample $n=50, 100, 200, 500$ households.
 - a. For each set, solve inverse HAP.
 - b. Evaluate the consistency and sample distribution of the population parameters as the sample size increases.
 - c. Evaluate the consistency of time of day OD matrices based on aggregation of the sampled trips to population level.

- d. Evaluate the consistency of the predicted time of day OD matrices across different sample sizes when travel times in the region increase by 20%.
- (5.12) Eq. (5.7) is an inverse traffic assignment problem under user equilibrium. Derive an inverse system optimal traffic assignment problem. For the Sioux Falls network, solve the UE and SO solutions. Randomize a weight α such that the simulated observation of link flows is $X_{obs} = \alpha X_{UE} + (1 - \alpha) X_{SO}$. Can the inverse traffic assignment problems be used to infer α ? The term can be used to infer the degree of centrality in the traffic flow and can be applied to determine, for example, how much centrality there is in truck flows due to alliances between carriers.
- (5.13) For the mHAPP model in Challenge 4.5, use inverse mHAPP to calibrate the parameters and compare the results to Challenge 4.5.
- (5.14) Create a coarse multimodal network to represent borough level activity in NYC. In this system, construct an mHAPP model to model users' mode choices throughout the day. Sample from the household travel survey in NYC (<https://www.nymtc.org/DATA-AND-MODELING/Travel-Surveys/2010-11-Travel-Survey>) and estimate mHAPP parameters.
 - a. Use these parameters to create a synthetic population for each borough in NYC. Compare the results to a population synthesis tool like Doppelganger (<https://github.com/sidewalklabs/doppelganger>).
 - b. Calibrate a MATSim model for the same data. Compare the MATSim performance for a new system scenario (such as adding a new transit station and line) (i) without mHAPP support and (ii) with the use of similarly calibrated mHAPP to synthesize an initial population of activity schedules to feed the day-to-day adjustment. Is there an improvement in convergence rate?
 - c. For the scenario in Challenge 5.14b, use mHAPP to estimate the population distribution of the elasticities to the new system changes.

CHAPTER 6

Privacy in Learning

6.1 INTRODUCTION

[Chapter 5](#) and this chapter deal with “Learning from Public Information.” While [Chapter 5](#) covers learning, this chapter deals with the latter: public information. The implication of using information from the public or information made available to the public is that the data is susceptible to privacy concerns. Due to privacy awareness, users, and operators may be less willing to share the data with the public. For users, sharing data with the public makes it easier for their identities to be stolen or for personal information to be made available to the wrong people. Operators, on the other hand, worry about competitors getting a hold of information that can be used to reverse engineer their operational policies, as illustrated in [Section 5.3.2](#).

Despite these concerns, there are also many benefits to sharing more data to the public in a smart cities setting. As noted in [Section 2.2](#), smart cities involve several dimensions that involve data gathering from the public to fuel knowledge creation. The advantages of smart cities are inherently connected to the type of data that users and operators are wary of sharing. Privacy preservation is arguably one of the biggest current obstacles in making smart cities truly viable.

There are many variations of the privacy preservation problem. In the case of users, anonymity in the presence of many other users’ data is desired. As an example, one privacy preservation tactic is the requirement to aggregate user survey data to a zone level so that the identity of a user cannot be revealed. In the case of operators, data cannot reveal too much information about the operator’s operating designs. An example is when ride-hail operators provide only passenger pickup information at a zone level so that there is insufficient information to learn the services’ pricing, dispatch, and routing algorithms.

This chapter deals with both types of data sharing privacy preservation: user privacy and operator privacy. Because user privacy focuses on hiding the identity of a record within a database of records, a technique called

“differential privacy” is introduced and applied to two types of user data: survey data and revealed location data. Operator privacy deals with hiding certain pieces of information. However, this can be too restrictive if the shared data is incomplete and the public agencies are unable to apply the data to practical uses. A new method is introduced involving diffusion of the data objects to mimic the anonymity objective in differential privacy. Instead of providing incomplete data, an operator has the option to provide messy or biased data that lessens the reliability of reverse engineered information. Several applications of operator privacy are explored: vehicle assignment decisions (to protect idle vehicle repositioning algorithms), scheduling decisions involving continuous time variables, route decisions, and tour decisions.

Lastly, the chapter examines the implications of having a privacy preservation model in place. Learning models become less accurate and other modeling and system design decisions that depend on the learning weaken. However, more analytics would also be possible because of the use of privacy preservation models. We look at network learning implications.

6.2 USER PRIVACY

One of the prevailing methods in user privacy preservation is differential privacy ([Dwork et al., 2006](#); [Dwork, 2008](#)). Differential privacy involves applying a function \mathcal{K} to a database that randomizes the data in such a way that the aggregate data output has at most ϵ difference when one element of the database is removed. By ensuring this, users can participate in the database without fear of being identified because the difference between the filtered databases with and without their data would be nominal (less than ϵ).

The motivation for this type of privacy definition is explained in [Dwork et al. \(2006\)](#). She showed that it is impossible to provide a privacy measure on a database such that an adversary with access to it cannot learn a user’s information. This is because the availability of auxiliary information (e.g., “Person X is two inches shorter than the average population height”) in combination with the database access would reveal the individual’s information, whereas access only to the auxiliary information would not. An alternative objective that can provide a measure of privacy is to instead minimize “the increase in risk of identification” due to participation in a database. The differential privacy theory was the culmination of several other studies conducted by the authors: [Dwork and Nissim \(2004\)](#), [Blum et al. \(2005\)](#), and [Dwork et al. \(2006\)](#).

Differential privacy has been applied to several different fields including transport. [Chen et al. \(2012\)](#) studied the use of differential privacy to protect transit smart card data in Montreal. [Le Ny and Pappas \(2014\)](#) designed filters for user data in ITS and smart grid systems to control for privacy. [Dong et al. \(2015\)](#) considered the challenges of incorporating privacy in travelers' origins and destinations in analyzing the routing game in a population.

6.2.1 Differential Privacy

A formal definition of differential privacy is given by [Dwork et al. \(2006\)](#).

Definition 6.1 ([Dwork et al., 2006](#)). *A randomized function \mathcal{K} gives ϵ -differential privacy if for all data sets D_1 and D_2 differing on at most one element, and all $S \subseteq \text{Range}[\mathcal{K}]$,*

$$\Pr[\mathcal{K}[D_1] \in S] \leq e^\epsilon \Pr[\mathcal{K}[D_2] \in S] \quad (6.1)$$

The condition in Eq. (6.1) can be defined as a combination of a deterministic function f and a noise. We further define a “ L_1 sensitivity” of the function f as follows.

Definition 6.2 ([Dwork et al., 2006](#)). *The L_1 sensitivity of a function $f: D^n \rightarrow \mathbb{R}^d$ is the smallest number $S[f]$ such that for all $x, x' \in D^n: \|f[x] - f[x']\|_1 \leq S[f]$.*

As shown in [Dwork et al. \(2006\)](#), a random filter \mathcal{K} defined in Eq. (6.2) with Laplace distributed noise having standard deviation $\frac{S[f]}{\epsilon}$ is ϵ -differentially private.

$$\mathcal{K}[D] = f[D] + Y, \quad Y \sim \text{Lap}\left[\frac{S[f]}{\epsilon}\right] \quad (6.2)$$

$\text{Lap}[b]$ is a Laplace distribution with the probability density function shown in Eq. (6.3) with mean $\mu = 0$ and standard deviation b .

$$f_L[x | \mu, b] = \frac{1}{2b} \exp\left[-\frac{|x - \mu|}{b}\right] \quad (6.3)$$

The concept is illustrated by [Zumel \(2015\)](#) and modified for [Exercise 6.1](#).

Exercise 6.1

Consider two data sets S and S' :

$$\begin{aligned} S &= \{0, 0, 0, \dots, 0\} \text{ (100 zeros)} \\ S' &= \{1, 0, 0, \dots, 0\} \text{ (1 one, 99 zeros)} \end{aligned}$$

Using a function $f[x \in S] = \mu_S$, determine a differentially private filter with $\epsilon = 1$.

Case I : deterministic filter $\mathcal{K}[D \in S] = \mu_S$

In this case, $\mathcal{K}[D \in S] = 0$ and $\mathcal{K}[D \in S'] = 0.01$. This filter is not ϵ -differentially private because an adversary can simply define a threshold $T = 0.005$ such that all queries can immediately determine which set the datum belongs to.

Case II : $\mathcal{K}[D \in S] = \mu_S + Y$, $Y \sim \text{Lap}[D | \mu = 0, b = 0.01]$

In this case, $S[f] = 0.01$ since it is the gap between the two data sets, and based on Eq. (6.2) the necessary standard deviation should be $b = S[f]/\epsilon = 0.01$. The noise would yield the following shown in Fig. 6.1.

We then verify if the noise is sufficient to meet the condition in Eq. (6.1), which we set as $\text{Diff} = e^\epsilon \Pr[x \in S_2] - \Pr[x \in S_1] \geq 0$. This is presented in Table 6.1 where $\Pr[x \in S_1]$ and $\Pr[x \in S_2]$ are obtained from Eq. (6.3) where $\mu = 0$ for S_1 and $\mu = 0.01$ for S_2 .

For a range of values of x , we see that the differences of Eq. (6.1) are either equal to or greater than zero. Hence this is differentially indistinguishable for $\epsilon = 1$. This means that someone trying to discern whether a value x belongs to either S_1 or S_2 would see that they are probabilistically nearly identical (off by a factor of at most e^ϵ).

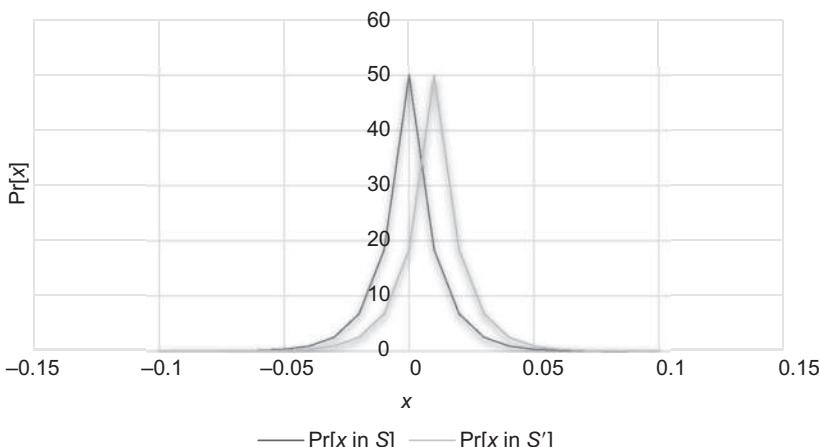


Fig. 6.1 Laplace distributed noise added to functions of values from S and S' .

Table 6.1 Illustration of the differential privacy criterion evaluation

x	$\Pr[x \in S_1]$	$\Pr[x \in S_2]$	$Diff$	x	$\Pr[x \in S_1]$	$\Pr[x \in S_2]$	$Diff$
-0.10	0.00227	0.000835	0	0.01	18.39397	50	117.5201
-0.09	0.00617	0.00227	0	0.02	6.766764	18.39397	43.23324
-0.08	0.016773	0.00617	0	0.03	2.489353	6.766764	15.90462
-0.07	0.045594	0.016773	0	0.04	0.915782	2.489353	5.850982
-0.06	0.123938	0.045594	0	0.05	0.336897	0.915782	2.152456
-0.05	0.336897	0.123938	0	0.06	0.123938	0.336897	0.791844
-0.04	0.915782	0.336897	0	0.07	0.045594	0.123938	0.291303
-0.03	2.489353	0.915782	0	0.08	0.016773	0.045594	0.107164
-0.02	6.766764	2.489353	0	0.09	0.00617	0.016773	0.039424
-0.01	18.39397	6.766764	0	0.10	0.00227	0.00617	0.014503
0	50	18.39397	0				

The Laplace distributed noise can be simulated using an inverse cumulative distribution function (inverse CDF). The CDF is $F_L[x|\mu,b] = \frac{1}{2} \exp\left[\frac{x-\mu}{b}\right]$ if $x < \mu$ and $F_L[x|\mu,b] = 1 - \frac{1}{2} \exp\left[-\frac{x-\mu}{b}\right]$ if $x \geq \mu$. The inverse CDF is shown in Eq. (6.4), where \tilde{x} is the simulated value of $x \in S$ and $r \in [0, 1]$ is a random draw.

$$\tilde{x} = \begin{cases} b \ln[2r] + \mu, & r < \frac{1}{2} \\ -b \ln[2(1-r)] + \mu, & r \geq \frac{1}{2} \end{cases} \quad (6.4)$$

Transport applications for differential privacy are divided into two primary categories. The first consists of travel-related survey stated preference data that users take, such as household travel surveys, OD surveys, or reviews of mobility services or activities. The second consists of revealed preference data obtained from monitoring or tracking individuals and primarily relate to their locations and time. For example, this could include cellphone location data, license plate tracking, transit smart card data, and so on. One example of each is illustrated in the following two sections.

6.2.2 User Survey Data

For survey data, one of the primary applications is behavioral modeling using discrete choice models. Almost no major studies have been conducted in differential privacy applications in user participation in data sets for choice modeling.

Individuals are typically surveyed on certain travel choices (e.g., mode, route, destination, departure time) along with attributes of the alternatives pertaining to the individual and individual-specific characteristics such as income, employment, vehicle ownership, and so on. Income is particularly of interest because when discrete choice models control for income, the estimated parameter associated with that variable can be used to estimate the expected consumer surplus of the set of options for that individual. For a discrete choice model with multinomial logit specification, the consumer surplus becomes $CS_n = \frac{1}{\alpha} \ln \sum_i \exp V_{in}$, where CS_n (\$) is the consumer surplus for individual n , V_{in} is the systematic utility of the discrete choice model pertaining to alternative i for individual n , and α is the marginal utility of income derived from the estimated model. The CS_n is such because the utility functions have Gumbel-distributed disturbances which feature such a property for expected maximum of multiple i.i.d. Gumbel distributions (see [Appendix D](#)).

For transportation planning purposes, it is often very important to be able to estimate the consumer surplus for each individual and use that to forecast the benefits to the population. However, people are hesitant to share their

actual income due to privacy concerns and may only answer a survey with income brackets, which complicates the model with categorical variables that do not allow for direct interpretation of the α without loss of information. For example, the US National Household Travel Survey uses income levels separated every \$5K up to \$80K, which has a \$20K bracket followed by one bracket for \$100K+.

What happens if we ask for users to report actual incomes and require that data requests for this data set pull ϵ -differentially private income values for an individual instead of the actual income? How biased would the estimated consumer surplus be under different values of ϵ ? This is illustrated in the [Exercise 6.2](#).

Exercise 6.2

Consider a sample of 20 individuals choosing between Auto and Transit modes, where a binary logit model has the following systematic utility for Auto: $V_{Auto,n} = -0.4868 + 0.01719I_n$ and $V_{Transit,n} = 0$, where I_n is income (\$1000s). The values are presented in [Table 6.2](#).

Table 6.2 Sample of 20 individual incomes to illustrate effect of differential privacy on consumer surplus

Individual	Original income	Choice	Original V_{in}	Original CS_n
1	\$7936	Auto	-0.3503	\$31,013
2	\$27,101	Auto	-0.0208	\$39,710
3	\$402,476	Auto	6.4335	\$374,258
4	\$27,101	Auto	-0.0208	\$39,710
5	\$37,061	Transit	0.1504	\$44,852
6	\$67,081	Auto	0.6666	\$62,870
7	\$47,207	Auto	0.3249	\$50,524
8	\$97,161	Transit	1.1838	\$84,381
9	\$27,101	Transit	-0.0208	\$39,710
10	\$147,130	Auto	2.0430	\$125,908
11	\$41,979	Auto	0.2350	\$47,547
12	\$17,338	Transit	-0.1887	\$35,084
13	\$92,033	Auto	1.0956	\$80,496
14	\$12,317	Transit	-0.2750	\$32,863
15	\$32,058	Auto	0.0644	\$42,216
16	\$220,267	Auto	3.3005	\$194,061
17	\$41,979	Transit	0.2350	\$47,547
18	\$32,058	Transit	0.0644	\$42,216
19	\$22,162	Auto	-0.1057	\$37,319
20	\$17,338	Transit	-0.1887	\$35,084
Total				\$1,487,369

Simulate the effect that a differential privacy filter with $\epsilon = \{1000, 100\}$ would have on the standard error for the estimated consumer surplus for this sample.

For $K[D] = f[D] + Y$ in Eq. (6.2), set f to be equal to the original income. In that case, the L_1 sensitivity is $S[f] = 394,540$ and the standard deviation is $b = 394,540/\epsilon$. For each ϵ , 10 simulations are sampled for each of the 20 individuals using Eq. (6.4). One sample set of filtered data for each ϵ is shown in Fig. 6.2.

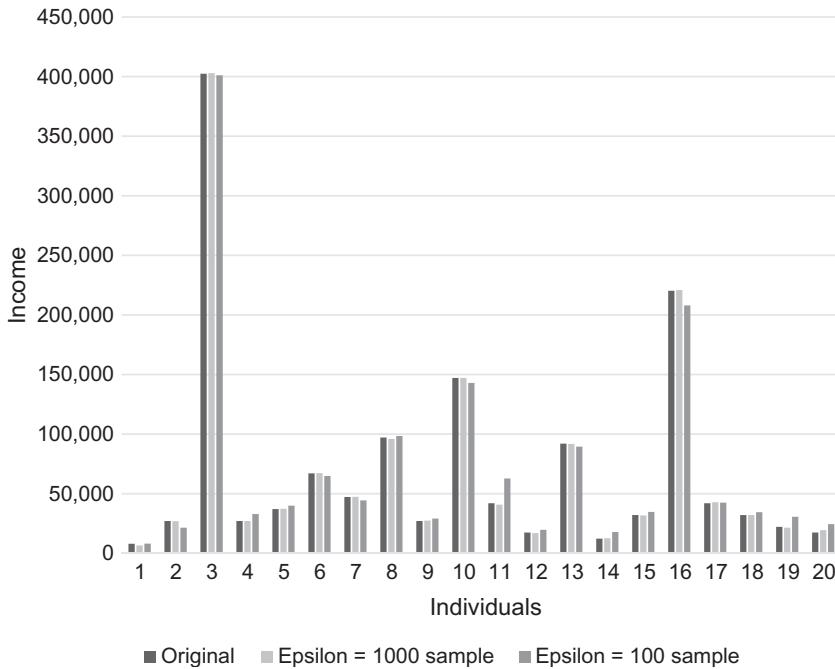


Fig. 6.2 Illustration of filtered income data according to $\epsilon = \{1000, 100\}$.

The results of the estimated parameters and consumer surplus for each simulated set of income values are presented in Table 6.3. It also includes the averages of the standard errors for the parameters and total consumer surplus. The true $\beta_{ASC} = -0.4868$, $\beta_I = 0.01719$, $\sum CS = \$1,487,369$ for this population of 20 individuals. As ϵ decreases from a requirement of 1000 to a requirement of 100 for the income data, a greater amount of randomization is needed. Because of the increased randomization, the standard errors of both estimated parameters and consumer surplus all increase by about five times even as the sample averages of the β_{ASC} , β_I , $\sum CS$ remain close to the true values. This example illustrates how an agency can use the differential privacy to control for the privacy of the survey participants' data subject to a choice model's reliability requirements.

Table 6.3 Summary of 10 simulated queries of income data with differential privacy for $\epsilon = \{1000, 100\}$

Simulation	$\epsilon = 1000$			$\epsilon = 100$		
	$\tilde{\beta}_{ASC}$	$\tilde{\beta}_I$	$\sum_n \tilde{CS}_n$	$\tilde{\beta}_{ASC}$	$\tilde{\beta}_I$	$\sum_n \tilde{CS}_n$
1	-0.4899	0.01735	\$1,481,215	-0.4622	0.01632	\$1,539,995
2	-0.4758	0.01689	\$1,503,453	-0.4139	0.01597	\$1,559,014
3	-0.4879	0.01723	\$1,486,767	-0.4487	0.01716	\$1,482,739
4	-0.4822	0.01705	\$1,494,186	-0.4402	0.01688	\$1,508,441
5	-0.4937	0.01727	\$1,483,964	-0.4971	0.01738	\$1,486,615
6	-0.4875	0.01718	\$1,488,157	-0.4495	0.01643	\$1,531,023
7	-0.4839	0.01709	\$1,494,836	-0.5037	0.01849	\$1,447,404
8	-0.4866	0.01713	\$1,487,744	-0.4087	0.01538	\$1,594,220
9	-0.5028	0.01750	\$1,471,318	-0.4231	0.01619	\$1,547,216
10	-0.4805	0.01701	\$1,496,411	-0.4818	0.01743	\$1,472,058
Average	-0.4871	0.01717	\$1,488,805	-0.4529	0.01676	\$1,516,873
Standard error	0.002367	5.59×10^{-5}	\$2849	0.01055	2.82×10^{-4}	\$14,298

6.2.3 User Location Data

Revealed user location data is a much more studied topic with differential privacy applications in transportation. Examples include [Dong et al. \(2015\)](#), who studied user OD locations obtained from Google route queries; [Chen et al. \(2012\)](#), who applied the method to transit smart card transactions; and [Aïvodji et al. \(2016\)](#), who managed user privacy while sharing their location data for ridesharing purposes. The revealed information may be due to some system-wide monitoring for users of the system. People are especially sensitive about the privacy of their locations in time.

To demonstrate the application of differential privacy in this setting, another example is introduced here where individuals' OD zones are filtered such that a different neighboring zone may be selected randomly. Like in [Exercise 6.2](#), a simulation of the noise is used to numerically evaluate the relationship between the design parameter ϵ and the resulting standard error of a travel demand model data object: the average trip length of travelers often used in calibrating trip distribution forecast models (see [Ortuzar and Willumsen, 2002](#)). In the case of zone locations, a discrete probability mass function is necessary. One such is derived proportionately to the PDF as a function of distance from the original zone, as illustrated with [Exercise 6.3](#).

Exercise 6.3

Consider a region of five zones with centroids shown in [Fig. 6.3](#) with the travel time (minutes) matrix and 20 samples of OD locations. Determine the effect that differential privacy with $\epsilon=5$ would have on the trip distribution and average trip length.

From the original data, the average trip length is $L=43.3$ min. For the noise, a discretized form of the Laplace distribution is used, where each zone has a probability that is proportional to the Laplace distribution and the sum of zone probabilities add to one. A value of $S[f]=78$ is used. Based on the discretized Laplace distribution, the following discrete probability mass functions for the relocation noise from one zone to another zone are shown in [Fig. 6.4](#).

Ten simulations are generated for each origin and destination of each of the 20 samples, as presented in [Table 6.4](#). The trip distribution matrix becomes more diffused as shown in [Fig. 6.5](#).

The sample average of the average trip length is $\bar{L}=42.0$ min and standard error is $\epsilon_L=0.939$ min, compared to the true value of $L=43.3$.

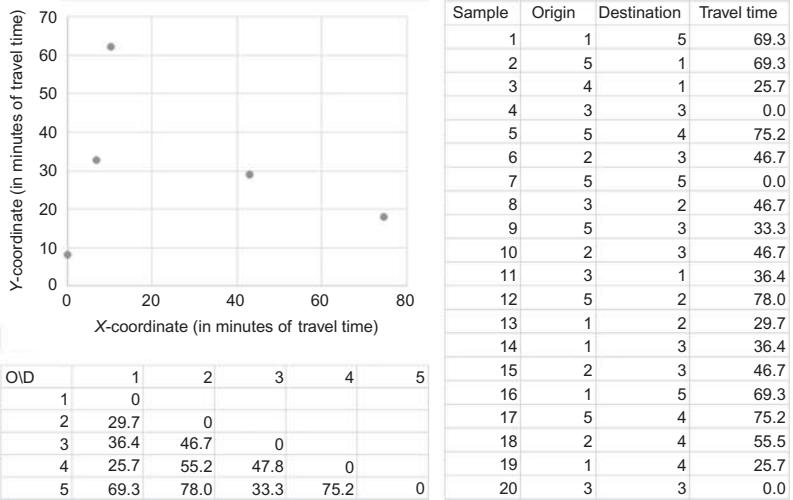


Fig. 6.3 Spatial distribution and travel times between zone centroids, and 20 sample OD locations.

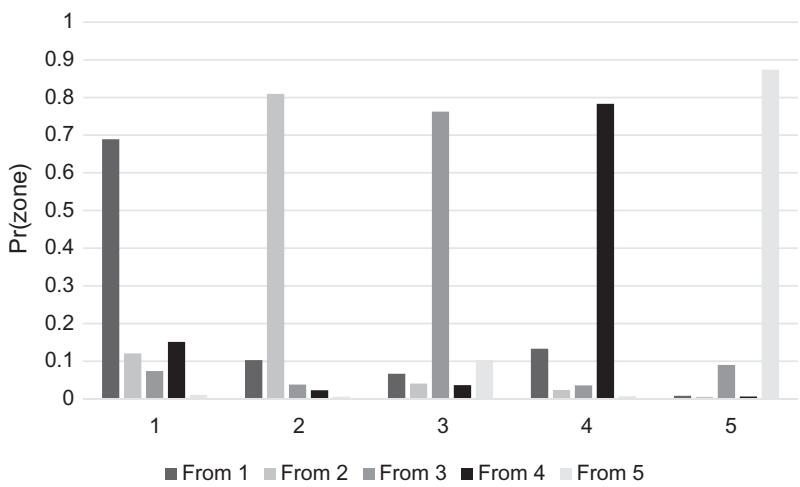


Fig. 6.4 Distributions of location noise by origin zone for $\epsilon=5$.

Table 6.4 Simulated origins and destinations

O1	D1	O2	D2	O3	D3	O4	D4	O5	D5	O6	D6	O7	D7	O8	D8	O9	D9	O10	D10
1	5	2	5	1	5	4	5	2	5	1	5	1	5	1	5	2	5	1	5
5	1	5	1	5	1	5	1	5	3	5	1	5	1	5	2	5	3	5	4
4	4	4	1	4	1	1	4	4	3	4	4	4	1	4	2	4	4	2	1
3	3	1	3	3	2	3	3	3	3	3	3	2	3	3	3	3	3	5	3
5	1	5	4	5	1	5	1	5	4	5	4	5	4	5	4	5	4	5	4
5	3	3	3	3	4	5	4	1	3	1	3	3	3	3	3	3	5	5	3
5	5	5	1	3	5	3	2	5	5	5	3	5	5	5	5	5	5	5	5
3	2	3	2	3	2	2	3	3	2	5	2	1	2	3	2	3	2	3	2
5	3	3	3	5	3	5	3	5	3	5	3	5	3	5	3	5	3	5	3
2	3	2	3	1	3	2	3	2	3	2	4	2	3	2	3	2	3	2	3
3	1	3	1	3	4	3	1	3	1	3	1	3	1	3	2	3	3	4	1
5	2	5	2	5	2	5	2	5	2	5	2	3	3	3	2	5	2	5	1
1	2	2	2	1	2	1	2	3	2	1	2	1	2	1	2	4	2	3	2
1	3	1	3	3	3	4	3	4	3	1	3	1	3	1	3	2	3	1	1
2	2	2	3	1	3	2	3	2	3	2	3	1	3	2	1	2	3	2	3
1	5	4	5	1	5	1	5	1	5	2	5	1	5	3	5	2	5	1	5
3	1	5	4	3	3	5	4	5	4	5	4	5	4	5	4	5	2	5	4
2	4	2	4	2	4	2	4	2	4	1	4	2	4	2	4	1	4	2	4
1	4	1	1	1	4	1	4	1	3	1	4	1	4	1	4	1	1	1	4
3	3	3	3	3	3	3	3	3	3	3	3	2	3	3	3	3	2	3	3

	1	2	3	4	5
Original OD distribution	1	0%	5%	5%	5%
OD distribution of 10 simulated samples	1	2%	4%	6%	5%
2	0%	0%	15%	5%	0%
3	5%	5%	10%	0%	0%
4	5%	0%	0%	0%	0%
5	5%	5%	5%	10%	5%

	1	2	3	4	5
Original OD distribution	1	0%	5%	5%	5%
OD distribution of 10 simulated samples	1	2%	4%	6%	5%
2	1%	1%	9%	5%	3%
3	4%	7%	11%	1%	2%
4	2%	1%	2%	2%	1%
5	6%	6%	8%	8%	3%

Fig. 6.5 Comparison of OD distributions for original and ϵ -differentially private locations.

6.3 OPERATOR COMPETITIVE PRIVACY CONTROL

Whereas the central question in user privacy is whether a user would participate in a database, operator privacy is driven by the goal of willingly sharing some data with a collaborator to achieve some benefit such that the data also becomes accessible to one or more adversaries. With a public collaborator, it is assumed that the data used for any public benefits would also require opening that data to the broad public. There is a need to stay competitive by restricting the amount of shared information to minimize the risk of adversarial strategies successfully reverse engineering the operator's policies. The trade-off leads to an information control problem (Sankar et al., 2011; Dong et al., 2015). An example of this type of privacy control is in Tsai et al. (2015), who proposed to share biased network link costs such that up to K shortest paths would all be equivalent in length. In this way they can achieve k -anonymity (Sweeney, 2002) in shortest path information.

A k -anonymous privacy control strategy involves filtering a data object (e.g., a tour, a route, a location, a dynamic fare price charged) such that the true object cannot be distinguished from $K - 1$ other objects. It acts as a kind of data diffusion. The objective of such data diffusion is to make the original data object sufficiently anonymous while maintaining accuracy in aspects that are useful to the receiver. In this section several classes of k -anonymous operator privacy control strategies are explored. The material is drawn primarily from He et al. (2017) and He and Chow (2018) but also expanded upon.

6.3.1 *k*-Anonymous Diffusion Models

The general framework of *k*-anonymous diffusion models has two components. The first deals with generating the K data objects. The second deals with constructing a filter to maximize the anonymity of the original among the objects. As an information-theoretic control problem, the objective of entropy maximization can be used to generate the objects and design the filter (Sun et al., 2013). When unconstrained, entropy maximization allocates the likelihood of different data sets equally. Consider the entropy maximization objective in Eq. (6.5) (see Wilson, 1967).

$$\max E = - \sum_i x_i \ln x_i \quad (6.5)$$

In an allocation problem (such as designing the probabilistic flows of a privacy filter) over K objects, the optimum without any additional information is to allocate $x_i = \frac{1}{k}$, $\forall i = 1, \dots, k$. For example, if $K=3$, the allocation of $x_1 = x_2 = x_3 = \frac{1}{3}$ yields $E^* = 1.0986$. This solution can serve as an upper bound for constrained problems with $K=3$.

A two-step procedure is used to design the filter \mathcal{K} for an original data object y_1 subject to some accuracy requirement Δ for the information Ω provided by the filtered data objects $\mathcal{K}[y_1]$. The first step (let us call it *SP1*) generates a set of K data objects $\{y_k\}_{k \leq K}$ that would maximize entropy. The second step (let us call it *SP2*) assigns the diffusion $\{x_k\}_{k \leq K}$ of the original data object to the set $\{y_k\}_{k \leq K}$ to ensure that constrained entropy is maximized subject to a noisy desired information $\tilde{\Omega}[\mathcal{K}[y_1]]$ being within the accuracy requirement shown in Eq. (6.6).

$$\tilde{\Omega}[\mathcal{K}[y_1]] \leq \Delta \quad (6.6)$$

The framework is shown in Fig. 6.6 where an iterative process removes unused data objects and adds them to a set U in each iteration to seek new objects in *SP1*. The loop continues until the final output only has $x_k > 0$. Note that $\lim_{x \rightarrow 0} x \ln x = 0$.

The solution quality can be measured by how much harder it is for an adversary to learn the operator's policies based on K anonymous data points instead. In Chapter 5, it is shown that parameters of an optimization model can be learned as a function of the inverse model: $\hat{\theta} = M^{-1}[y_1; \theta_0]$ where y_1 is some observed decision variables, M is the model operated by the operator, θ_0 is the prior information on the operator's model parameters, and $\hat{\theta}$ is a deterministic estimate. Under the *k*-anonymous diffusion model, the $\{x_k\}$ -diffused data objects $\{y_k\}_{k \leq K}$ become a sample of "observed" variables,

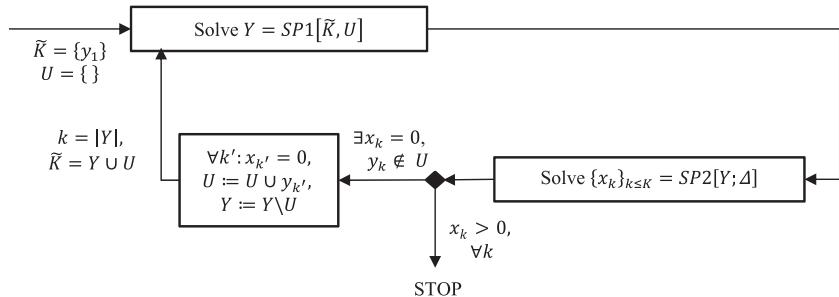


Fig. 6.6 Framework for k -anonymous diffusion models.

and the estimate of $\hat{\theta}$ becomes a sampled value with a standard error ϵ_θ . This standard error is a measure of performance of the diffusion model: the objective is to maximize this error ϵ_θ given a desired Δ .

The data objects y_k have so far been defined vaguely. In practice, there are many different potential data objects to diffuse. Some of these are explored in the following sections: vehicle assignment decisions, routes, and tours.

6.3.2 k -Anonymous Vehicle Assignment Diffusion Model

In many mobility service systems, there is a need to reposition or rebalance idle vehicles (see [Sayarshad and Chow, 2017](#)). In many cases the type of rebalancing algorithm employed can make or break a service. Several examples of such systems that failed are shown in [Chapter 1](#). In some cases, a factor for the failure may be due to the effectiveness of the algorithm. As such, the algorithm is a trade secret in which a mobility operator seeks to protect from adversaries.

Assignment of idle vehicles in its most basic form can be structured as a Koopmans-Hitchcock Transportation Problem ([Samuelson, 1952](#)) in which all nodes are either origins or destinations. Revealing the flows, like with the NYC taxi data, allows adversaries to infer several different parameters:

- Demand and utility of a destination zone relative to other destination zones
- Perceived travel times between OD pairs
- Inference of the weight between current costs and future expected costs for algorithms that consider look ahead opportunity costs (see [Sayarshad and Chow, 2015](#))

- If and how the repositioning considers spatial constraints or preferences (e.g., cruising vs waiting)
- If and how the repositioning considers en route matching with passengers

Given all these potential targets for adversarial learning, it makes sense that private operators may not wish to share the details. Indeed, companies like Uber have been hesitant to share “drop-off to pickup” portions of their operational data. Public agencies, on the other hand, benefit from such disaggregate data:

- Spatial and temporal distribution of empty trips which help infer multi-modal trip demand
- Distribution of empty trip miles which impact congestion without directly providing transport service

If the operators are required to, they might consider using a k -anonymous diffusion model framework. One model under this framework is to control for privacy while ensuring that the filtered data’s average empty trip miles are similar to the original empty trip miles.

The data object in this case is a set of flows, given known supply and demand quantities at each node, and travel costs between each OD pair. [He and Chow \(2018\)](#) introduced a method to generate network optimization data objects by using a link-based cost minimization objective. The link costs are artificially constructed to avoid repeating their usage in other data objects in a set \tilde{K} . Consider a network in Fig. 6.7A with six nodes organized with three supply nodes on the left of one unit supply each and three demand nodes on the right with one unit demand each. The arrows indicate the original flows that the operator wishes to anonymize.

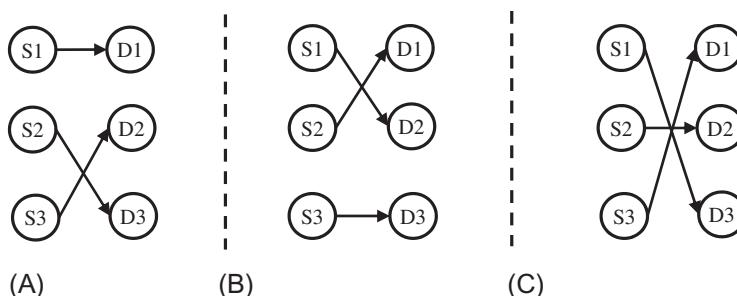


Fig. 6.7 (A) Original flows, (B) outcome assignment to minimize repeating the original flows, and (C) the outcome assignment to minimize repeating the combined flows on the left.

The three configurations in Fig. 6.7 represent different data objects. Each time a new data object is generated, the flows get added to a cost matrix. For example, the iterative updates to the cost matrix in Fig. 6.7 are shown as follows.

Based on original assignment in Fig. 6.7A:

c^1	D1	D2	D3
S1	1	0	0
S2	0	0	1
S3	0	1	0

After Fig. 6.7B:

c^2	D1	D2	D3
S1	1	1	0
S2	1	0	1
S3	0	1	1

After Fig. 6.7C:

c^3	D1	D2	D3
S1	1	1	1
S2	1	1	1
S3	1	1	1

After the data objects are generated (designated as $SP1$ in Fig. 6.6), they are used to determine the probability of being selected. Algorithm 6.1 is shown here and illustrated with Exercise 6.4.

Algorithm 6.1: k -Anonymous Vehicle Assignment Diffusion

Inputs: Original assignment vector y_1 , link cost matrix t , tolerance for average trip cost Δ , nodes designated as origins O or destinations D

1. $\tilde{K} := \{y_1\}$, $Y = \{y_1\}$, $k := 1$, $U := \{\}$
2. *while* $k < \tilde{K}$
 - a. $c^k := f[\tilde{K}]$
 - b. Solve for y_{k+1} from Eq. (6.7) as an LP

$$\min \sum_{(i,j)} c_{ij}^k y_{k+1,ij} \quad (6.7a)$$

Subject to

$$\sum_j \gamma_{k+1,ij} = \sum_j \gamma_{1,ij}, \quad i \in O \quad (6.7b)$$

$$\sum_j \gamma_{k+1,ji} = \sum_j \gamma_{1,ji}, \quad i \in D \quad (6.7c)$$

$$\gamma_{k+1,ij} \geq 0, \quad \forall (i,j) \quad (6.7d)$$

c. $\tilde{Y} := Y \cup \gamma_{k+1}$

d. $\tilde{K} := Y \cup U$

e. $k := k + 1$

3. Solve for $\{x_k\}_{k \leq K}$ from Eq. (6.8) as a convex nonlinear program with linear constraints (Frank-Wolfe algorithm, shown in [Chapter 3](#), can be used to solve)

$$\max - \sum_k x_k \ln x_k \quad (6.8a)$$

Subject to

$$\sum_k x_k = 1 \quad (6.8b)$$

$$\sum_{(i,j)} t_{ij} \gamma_{1,ij} - \sum_k \sum_{(i,j)} t_{ij} \gamma_{k,ij} x_k \leq \Delta \quad (6.8c)$$

$$\sum_k \sum_{(i,j)} t_{ij} \gamma_{k,ij} x_k - \sum_{(i,j)} t_{ij} \gamma_{1,ij} \leq \Delta \quad (6.8d)$$

$$x_k \geq 0 \quad (6.8e)$$

4. if $x_k > 0 \forall k \leq K$, go to Step 5, else
a. $U := U \cup \gamma_{k'}$ for all k' where $x_{k'} = 0$
b. $Y := Y \cup U$
c. $k := |Y|$
d. $\tilde{K} := Y \cup U$
e. go to Step 2
5. Randomize the order of the K data objects and their corresponding probabilities

Outputs: K sets of OD flow patterns γ_k , corresponding probabilities x_k .

Exercise 6.4

Consider an example with three origins and three destinations with travel times presented in [Table 6.5](#), which may represent an idle taxi relocation over an interval of 1 h. Suppose the original flow configurations are $\gamma_{1,11}=50, \gamma_{1,13}=50, \gamma_{1,23}=150, \gamma_{1,32}=200$. Design a 4-anonymous diffusion of the flow for different values of Δ .

Table 6.5 Travel times and demand

t_{ij} (min)	D1	D2	D3	Supply
O1	4	17	11	100
O2	18	7	6	150
O3	13	13	21	200
Demand	50	200	200	450

Three iterations of solving Eq. (6.7) with updated c^k results in the following matrices.

y_1	D1	D2	D3	c_{ij}^1	D1	D2	D3
O1	50	0	50	O1	50	0	50
O2	0	0	150	O2	0	0	150
O3	0	200	0	O3	0	200	0

y_2	D1	D2	D3	c_{ij}^2	D1	D2	D3
O1	0	100	0	O1	50	100	50
O2	50	100	0	O2	50	100	150
O3	0	0	200	O3	0	200	200

y_3	D1	D2	D3	c_{ij}^3	D1	D2	D3
O1	0	0	100	O1	50	100	150
O2	0	150	0	O2	50	250	150
O3	50	50	100	O3	50	250	300

y_2	D1	D2	D3				
O1	0	100	0				
O2	0	0	150				
O3	50	100	50				

Based on the set of y_k , the probabilities as they vary by Δ (in veh-min) are shown in Fig. 6.8. A value of $\Delta=900$ has an average trip length tolerance of 2 min/veh since there are 450 trips.

Fig. 6.8 illustrates how as the accuracy tolerance Δ increases up to 1475, the probabilities change until they converge upon the upper bound when all the probabilities are equal. Based on the rate of convergence, it is also clear that assignment y_2 is the most different from the original assignment. This makes sense because of the method employed in generating that assignment by perturbing as far from the original solution as possible. For values of Δ below 600, the solution is to keep only the original assignment without any diffuse results. In that case, the algorithm would remove the other unused assignments and add them to set U before generating new assignments. If K

were to be increased beyond 4, we should also see more assignments generated earlier that would be closer in accuracy to the original such that lower values of Δ might be able to obtain diffuse results.

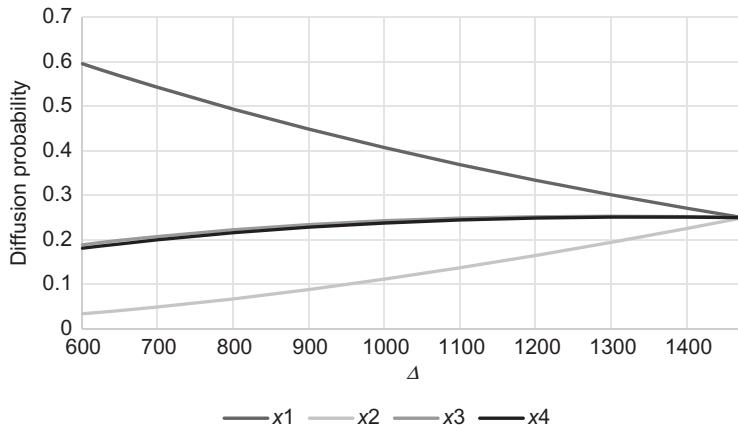


Fig. 6.8 Shift of probabilities from maximum probability for origin flows to unconstrained maximum entropy.

The exercise illustrates several categories of k -anonymous diffusions.

Definition 6.3 *A nonanonymous diffusion is a set of data objects in which the error tolerance Δ prevents the original data object from diffusing to other data objects.*

Definition 6.4 *A perfectly anonymous diffusion is a set of data objects in which the error tolerance Δ is sufficiently high that the upper bound entropy is reached. In this scenario, all data objects are equally likely to occur.*

The algorithm can be applied to different data sets where assignment of vehicles from origins to destinations is to be anonymized. It can also be tested on data sets where this information is available: NYC taxi data, bike share data, or commodity flow data, among others.

6.3.3 k -Anonymous Tour Diffusion Model

A second type of data diffusion is with tours for routing operations. Many private operations compete using custom vehicle routing algorithms: mobility-on-demand services, trucking companies, among others. As shown in Section 5.3, given routing data it is straightforward for an adversary

to reverse engineer either the specific routing policies or equivalent policies that result in the same or similar observed patterns. To be specific, the following types of parameters can be inferred using tour data:

- Identification and prioritization of different routing objectives, for example, travel times or route length, passenger wait times, passenger total journey times, or vehicle utilization
- Identification of dispatch criteria and constraints
- Existence and value of hard time windows or penalties for soft time windows
- Importance placed on minimizing future costs in a dynamic algorithm
- Presence and value of constraints to limit amount of passenger detours
- Value of destinations in profitable tour problems in which destinations are chosen among a set of candidates

Public agencies can benefit from such data. Anonymous tour data can be designed to be accurate regarding one or more different variables:

- Zonal spatial demand distribution or OD distribution
- Average passenger journey (wait time plus travel time) or in-vehicle travel time
- Vehicle miles traveled and passenger miles traveled

A similar k -anonymous diffusion model can be used to protect the privacy of tour data using the same framework shown in Fig. 6.6. He and Chow (2018) proposed such a model where the data object is the set of tours for a fleet of vehicles. The observable variables are assumed to be the set of pickup and drop-off locations and the travel times between them, the vehicle fleet size, and the vehicle capacities. Algorithm 6.2 is shown for this model, where the privacy control is set to ensure that the filtered data's average passenger travel time is similar to the original value.

Algorithm 6.2: k -Anonymous Tour Diffusion

Inputs: Original tour set γ_1 with true average passenger travel time τ , node pair travel time matrix t , tolerance for average passenger travel time error Δ , passenger requests as pickups $O = \{1, \dots, n\}$ and drop-offs $D = \{n+1, \dots, 2n\}$, a fleet V , initial vehicle locations $\{0_v\}_{v \in V}$ ($N = \{O, D, \{0_v\}\}$), a desired number of tours K

1. $K := \{\gamma_1\}, Y = \{\gamma_1\}, k := 1, U := \{\}$
2. *while* $k < K$

- a. $c^k := f[\tilde{K}]$, where the function f is based on counting the number of visits made to each node pair in \tilde{K}

- b.** Solve for γ_{k+1} from Eq. (6.9) as a dial-a-ride problem (adapted from Eq. 5.4)

$$\min \sum_{(i,j)} c_{ij}^k \gamma_{k+1,ij} \quad (6.9a)$$

Subject to

$$\sum_{v \in V} \sum_{j \in N} \gamma_{ijv} = 1, \forall i \in O \quad (6.9b)$$

$$\sum_{j \in N} \gamma_{0_v j v} = \sum_{j \in N} \gamma_{j, 2n+1, v}, \forall v \in V \quad (6.9c)$$

$$\sum_{j \in N} \gamma_{0_v j v} \leq 1, \forall v \in V \quad (6.9d)$$

$$\sum_{j \in N} \gamma_{ijv} = \sum_{j \in N} \gamma_{n+i, j v}, \forall i \in O, v \in V \quad (6.9e)$$

$$\sum_{j \in N} \gamma_{ji v} = \sum_{j \in N} \gamma_{jiv}, \forall i \in O \cup D, v \in V \quad (6.9f)$$

$$T_i - T_j \leq -d_i - t_{ij} + (1 - \gamma_{ijv})M, \forall i, j \in N, v \in V \quad (6.9g)$$

$$W_i - W_j \leq -q_j + (1 - \gamma_{ijv})M, \forall i, j \in N, v \in V \quad (6.9h)$$

$$T_{n+i} - T_i - R_i \leq d_i, \forall i \in O \quad (6.9i)$$

$$0 \leq W_i \leq u, \forall i \in N \quad (6.9j)$$

$$\gamma_{ijv} \in \{0, 1\} \quad (6.9k)$$

$$t_{i,n+1} \leq R_i \leq R_{\max} \quad (6.9l)$$

$$0 \leq T_i \leq T_{\max} \quad (6.9m)$$

c. $Y := Y \cup \gamma_{k+1}$

d. $\tilde{K} := Y \cup U$

e. $k := k + 1$

- 3.** Solve for $\{x_k\}_{k \leq K}$ from Eq. (6.10) as a convex nonlinear program with linear constraints (Frank-Wolfe algorithm, shown in [Chapter 3](#), can be used)

$$\max - \sum_k x_k \ln x_k \quad (6.10a)$$

Subject to

$$\sum_{k \in K} x_k \leq |V| \quad (6.10b)$$

$$\sum_{k \in K} \delta_{jk} x_k \geq 1, \forall j \in O \quad (6.10c)$$

$$\sum_{k \in K} (t_{j+n,k} - t_{jk}) x_k - \tau_j \sum_{k \in K} x_k \leq \Delta \sum_{k \in K} x_k, \forall j \in O \quad (6.10d)$$

$$-\sum_{k \in K} (t_{j+n,k} - t_{jk}) x_k + \tau_j \sum_{k \in K} x_k \leq \Delta \sum_{k \in K} x_k, \forall j \in O \quad (6.10e)$$

$$x_k \geq 0 \quad (6.10f)$$

where δ_{jk} is 1 when node j is visited by tour k , otherwise 0; and t_{jk} is the arrival time at node j via tour k .

4. if $x_k > 0 \forall k \leq K$, go to Step 5, else
 - a. $U := U \cup \gamma_{k'}$ for all k' where $x_{k'} = 0$
 - b. $Y := YU$
 - c. $k := |Y|$
 - d. $\tilde{K} := Y \cup U$
 - e. go to Step 2
5. Randomize the order of the K data objects and their corresponding probabilities

Outputs: K sets of tours γ_k , corresponding probabilities x_k .

To better understand the properties of this model, let us examine an illustrative example with a single vehicle fleet serving three passengers (six nodes) as shown in Fig. 6.9. The coordinates are to the nearest whole number shown while the travel time is simply defined as the Euclidean distance between two points in the graph in Fig. 6.9A. Exercises 6.5 (from He et al., 2017) through 6.6 (He and Chow, 2018) are based on this figure.

Based on the example, we define another type of solution that exists: a binding feasible data object diffusion. In a binding feasible data object diffusion, the maximum number of feasible objects with $x_k > 0$ is less than the desired K objects. According to this definition, the solution to Exercise 6.5 is binding for $K=90$ since not all the 90 tours generate $x_k > 0$.

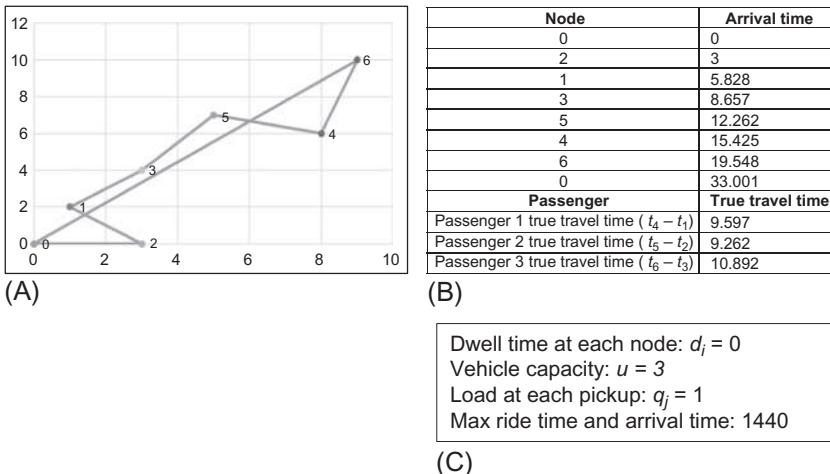


Fig. 6.9 (A) Example tour, with (B) true vehicle arrival times and passenger travel times, and (C) other known parameters. ((A) Source: [He et al., 2017](#).)

Exercise 6.5

For the data object in Fig. 6.9, enumerate all possible tours and find the optimal diffusion constrained with $\Delta=0.1$. Identify the unused tours, if any.

First, the travel times between each node pair are obtained and presented in Table 6.6.

Table 6.6 Travel times for Exercise 6.5

OD	0	1	2	3	4	5
0						
1	2.236					
2	3.000	2.828				
3	5.000	2.828	4.000			
4	10.000	8.062	7.810	5.385		
5	8.602	6.403	7.280	3.606	3.162	
6	10.817	8.944	8.485	6.325	1.000	4.123

There is a total of $6! = 720$ possible sequences of the six nodes. However, the number of feasible tours considering the precedence constraint reduces to 90 tours. If we enumerate them and sort them from shortest length to longest length, there is no need to solve the data object generation subproblem in Eq. (6.9). We just solve Eq. (6.10) for the 90 tours and obtain the solution in Fig. 6.10.

In this solution there are clearly several tours that are allocated negligible probability of occurrence. Tour no. 84 and 85 are two examples. This shows

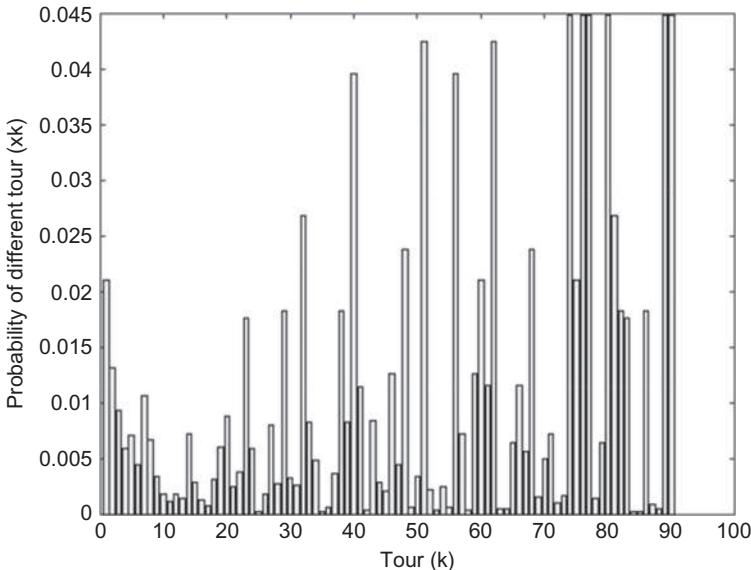


Fig. 6.10 Diffusion of $(0, 2, 1, 3, 5, 4, 6, 0)$ into 90 tours with $\Delta = 0.1$. (Source: He et al., 2017.)

that it is possible under constrained circumstances for an entropy maximizing k -anonymization model to remove a subset of objects. In Algorithm 6.2, these unselected objects are removed from the set and assigned to U and the remaining tours are used to initiate another iteration of Eq. (6.9) to generate up to K tours.

Definition 6.5 A *binding feasible data object diffusion* is a scenario where the optimal diffusion makes use of every single feasible data object in the set S of all feasible objects where $|S| < K$.

In the next exercise we explore the value of using the tour generation subproblem in Eq. (6.9). Instead of using the subproblem, one can naively use a k -best TSP (e.g., Van der Poort et al., 1999) or VRP algorithm to generate a set of shortest tours. However, Fig. 6.10 clearly shows that the shortest tours (on the left-hand side in the sorted list of tours) are not always the best tours to assign the diffusion probabilities. A tour generation subproblem needs to select tours that maximize the entropy. Knowing that entropy maximization seeks to spread out as differently as possible, the subproblem in Algorithm 6.2 does so by imposing a cost on legs used by prior select tours. This concept is illustrated in Exercise 6.6.

Exercise 6.6

Apply Algorithm 6.2 to the data object in Fig. 6.9 for $K=1, \dots, 10$ under $\Delta=0.1$ and compare the change in entropy of the tour generation with (1) a naïve tour generation based on shortest tour length and (2) the unconstrained diffusion.

We iteratively solve the k -anonymous tour diffusion model for $K=1, \dots, 10$ under the 10 shortest tours and under 10 tours found through Algorithm 6.2. For Algorithm 6.2, the solution (prior to randomizing the order in Step 5) when $K=10$ is shown in Fig. 6.11.

Fig. 6.11 shows how the diffusion leads to a wide assortment of results where the artificial average passenger time does fall within the constraint. When the number of tours is iteratively increased for each problem from 1 to 10, the entropy value for Algorithm 6.2 is plotted in Fig. 6.12 along with the naïve tour generation approach and the upper bound obtained for the unconstrained entropy.

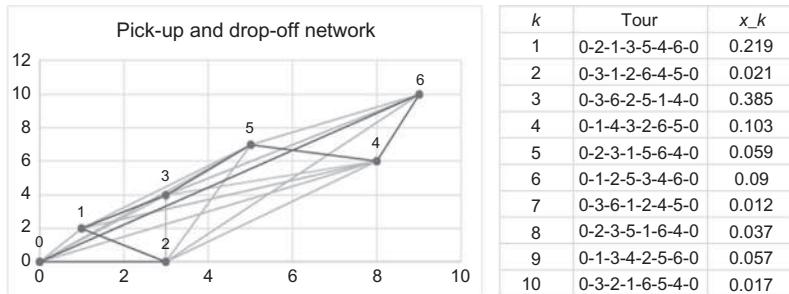


Fig. 6.11 Ten-anonymous diffusion of $(0, 2, 1, 3, 5, 4, 6, 0)$ using Algorithm 6.2 with $\Delta=0.10$.

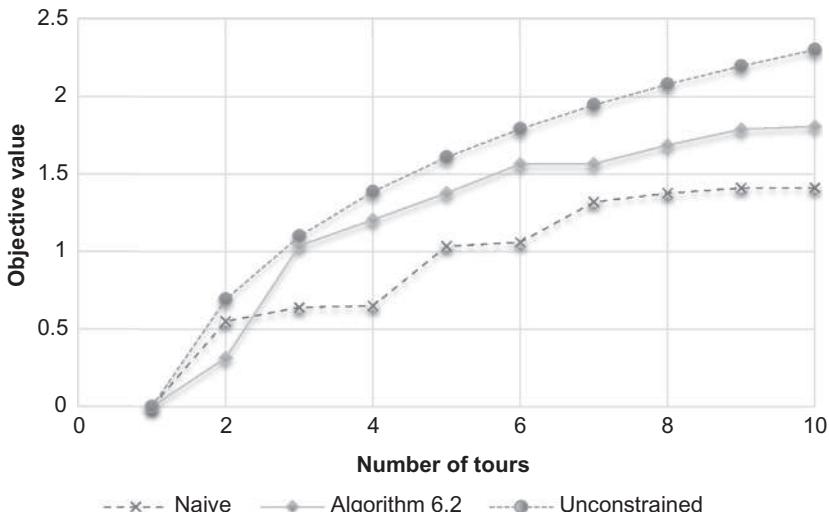


Fig. 6.12 Comparison of entropy value for number of tours under different tour generation methods.

There are two takeaways from this comparison. First, neither tour generation approach reaches the upper bound based on entropy maximization under an unconstrained setting, which is expected. Over the course of the 10 tours, the average performance of [Algorithm 6.2](#) leads to a higher performing entropy rate, cutting the gap with the upper bound by approximating half (58% of gap covered for cases when it is better).

Second, when the number of tours is few, the tour generation may end up having multiple nonunique solutions to choose from (since most link costs would be zero) and the algorithm may end up choosing a solution that is not a good fit with the Δ tolerance. As the number of tours increases, the risk of nonunique tours generated is overridden by the increased diversity in tours. Nonetheless, this shows that while the objective of the tour generation step does indeed seek to maximize entropy, it does not guarantee optimality.

Ultimately the impact of the privacy protection is to increase the noise around the parameters that one wishes to protect. For example, in the case of tour diffusion models one might be interested in protecting the identification of the objective parameters of the DARP model being used. The impact on the resulting model is discussed further in [Section 6.4](#).

6.4 NETWORK LEARNING WITH PRIVACY AWARENESS

The prior sections dealt with how private operators and users would protect their data from public agencies and public use. For this paradigm to work effectively, what the public agencies learn from the shared data needs to be sufficiently accurate. The Δ are not parameters generally decided by the private operators; they are design parameters set as requirements by the public agencies to ensure some accuracy is achievable.

Designing the appropriate Δ depends on what kind of learning the public agency seeks. As pointed out in [Dwork \(2008\)](#), learning from ϵ -differentially private data (what she calls *privately agnostic learning*) may not be efficient. It is likely the case also for k -anonymous diffused data. This is still a generally new area, so at this point only some guidelines are put forth with sketches of potential applications.

At the fundamental level, the public agency can use sensitivity analysis to identify parameter error thresholds for which the reliability of the learning is no longer useful. For example, the problem may be OD estimation based on privacy-protected probe vehicle data. If the public agency has a desired

reliability constraint, they can then solve a problem to maximize Δ such that the OD estimation with the reliability constraint is binding.

Armed with this information (presumably from both the private operators and public agencies), the next step is for the decision-makers to agree to a set Δ . Game theory can be a useful application here (e.g., Liu et al., 2013). Looking at the two decision-makers as the players, the convergence to a common set of agreed tolerance parameters can be modeled as a noncooperative game to identify Nash equilibria. This would help the decision-makers set achievable goals and identify opportunities to collaborate and share information to overcome undesirable equilibria.

A more complex scenario may involve dynamic learning and flexible adaptation of the tolerance. For example, perhaps some anonymized real-time data is shared with the public. The level of tolerance may be subject to change in response to the size of the market, the power of the private operator with respect to its adversaries, the threat level over time, and so on. Perhaps traveler location data is less susceptible to inference when the traveler is in an outlier setting like traveling in another city. In those cases, perhaps the tolerance level may be reduced. Dynamic privacy control may be able to exploit information much more efficiently by taking advantage of such opportunities. In those cases, Markov decision processes can be designed to optimally update the tolerance level over time.

From the user perspective, privacy remains a very heterogeneous issue that also depends on human behavior and risk aversion (Savage and Waldman, 2015; Dogruel et al., 2017). Public agencies need to have a better understanding of users' value of privacy and develop policies that are cognizant of that value. These include pricing or incentivization schemes or designing more integrated public involvement in the management of the transport services in the future.

We illustrate network learning as discussed in [Chapter 5](#) with the observation of a diffused route set instead of a single route. In this case, an observer receives a set of paths $\{\gamma_{ik}\}$ instead of a single path γ_i on the i th observation. The inverse problem cannot simply be converted from an inverse shortest path problem to an inverse multicommodity flow problem because some of the randomly generate paths are artificially generated and may not actually reflect the same link capacity state as the true state during the observation of the data object.

Instead, each of the K paths needs to be used individually in an inverse shortest path problem as shown in [Eqs. \(5.12a\)–\(5.12e\)](#), like the online learning example in [Section 5.5.2](#). This is illustrated with [Exercise 6.7](#).

Exercise 6.7

Suppose we have received $K=2$ paths, where path 1 is $\gamma_1=[4, 5, 6, 7, 8, 2]$ and path 2 is $\gamma_2=[4, 5, 9, 10, 11, 2]$ with diffusion of $x_1=0.70$ and $x_2=0.30$ as shown in Fig. 6.13. Plot the 95% confidence interval of the posterior dual price of link 7 as a function of the prior of link 7.

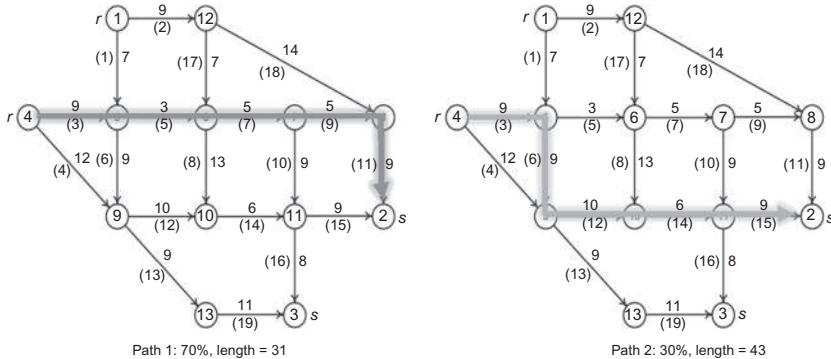


Fig. 6.13 Two-path diffused data object set.

The distribution of the inverse optimized dual price depends on the prior. The γ_1 is the uncapacitated shortest path for OD (4, 2), so it can essentially be any value. If the prior dual price of link 7 happens to be $w_7^0=7$, then the posterior $w_7^*|_{w_7^0=7, \gamma_1}=7$ as well. For γ_2 , however, the dual price must be 12 or higher. Therefore $w_7^*|_{w_7^0=7, \gamma_2}=12$. This yields an average of $\bar{w}_7^*|_{w_7^0=7}=7(0.7)+12(0.3)=8.5$. A plot of the inverse optimal dual prices and the weighted average as a function of the prior for link 7 is shown in Fig. 6.14

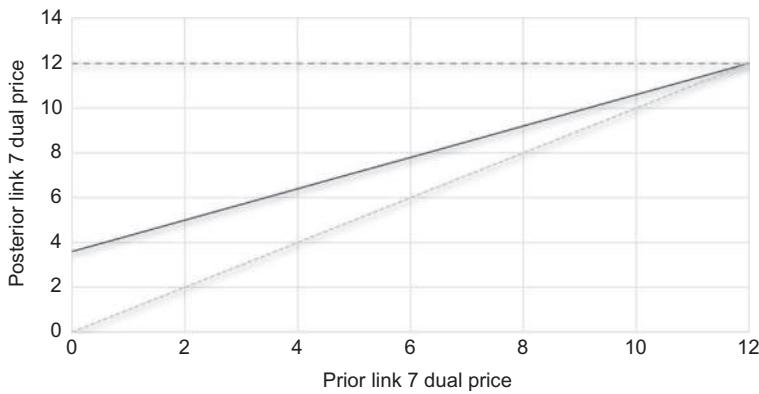


Fig. 6.14 Two-anonymous distribution of posterior link 7 dual price as a function of prior.

along with the discrete values. The plot shows that by simply introducing one extra path with arbitrary diffusion probabilities, the reliability of efforts to reverse engineer the state of the network can become considerably clouded.

The plot suggests that the network learning using k -anonymous diffused data can be done, although the reliability of the results depends not only on the generated tours but also on the prior information.

RESEARCH AND DESIGN CHALLENGES

The following are open-ended challenges that sometimes refer to real data sets. Readers are encouraged to try to tackle the problems, and post their designs or solutions on the following GitHub page, <https://github.com/BUILTNYU/Elsevierbook/tree/master/Chapter%206>.

- (6.1) Take the NYC taxi data (http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml) and:
 - a. Create an app that applies a differential privacy filter with specified ϵ to anonymize queried pickup and drop-off locations of users.
 - b. Generate a K -anonymous tour diffusion set for a randomly sampled set of taxi tours. Evaluate the goodness of fit measures of a destination choice model for idle taxis (which can currently be reverse engineered using the taxi data, see Challenge 5.5) when the data is diffused. How much less reliable is the taxi destination choice model outcome?
- (6.2) Design and evaluate a differential privacy filter for the timestamps of the Foursquare check-in data in NYC (<https://sites.google.com/site/yangdingqi/home/foursquare-dataset>).
- (6.3) Take Citi Bike data (<https://www.citibikenyc.com/system-data>) of stations at midnight to stations at 6 a.m. to represent nightly rebalancing patterns. Determine the error tolerance needed to allow for perfectly anonymized diffusion.
- (6.4) For a population of size P , a traditional household travel survey of size conducted with a sample of S has a sample error of ϵ_S . If the complete population was to share data but the database would be setup to be differentially private, what ϵ would be needed to have an equivalent sample error? How does this result depend on the variable of interest?

- (6.5) The Commodity Flow Survey (<https://www.bts.gov/product/commodity-flow-survey>) publishes commodity flow data aggregated up to 132 freight analysis zones across the country (2.6 zones per state). Using differential privacy considerations for firm location identity, design an average zone size that would be sufficiently private.
- (6.6) Download 1 week of your own Google timeline data and select a Δ in accuracy of travel times and generate a 5-anonymous tour diffusion.
- (6.7) Synthesize a 3-passenger pickup/drop-off problem instance with an open tour and show how the distribution of α , β , γ in Eq. (5.4) change when the number of tours and Δ change.
- (6.8) Take the data from Challenge 5.6. Compare the estimated link dual price distributions when the routes are diffused into 5-anonymous routes.
- (6.9) Conduct a survey to obtain willingness to share one's location data in exchange for a discount on toll pricing or transit fare. Considering the user's choice to share data, design a fare discount that would maximize consumer surplus.

CHAPTER 7

Network Design

7.1 INTRODUCTION

The last section of this book covers the design and operation of transport systems in a network. Network design has a long history with pervasive applications in transport, telecommunications, the Internet, logistics, economics, and more. As demonstrated in [Chapter 3](#) with examples like Braess' Paradox, effects of changes in a network setting can be counterintuitive and cannot be trivially assumed. As a result, any optimization in a network setting needs to consider these network effects, whether it is highway expansion, toll pricing, station setting, or setting system-wide policies such as idle vehicle relocation, dispatch, or routing.

In the era of smart cities, the problem of operating and managing a transport system in an urban network is more topical than ever. Autonomous vehicle fleets need algorithms to link their sensed information with routing policies to serve users efficiently. Dynamic on-street parking pricing systems inevitably impact the spatial preferences of cruising users which affect the congestion levels in arterial networks. Bus holding strategies need to consider the effects on passengers upstream and downstream as well as those making transfers to and from other lines. E-hail taxi services need to predict spatial patterns of both the users and other competing taxis. Bikeshare incentivization programs need to predict the spatial-temporal demand for bikes and offer sufficient incentives to users to encourage them to return bikes to desired locations. Public electric vehicle charging stations need to be located where they best serve electric vehicle fleets. Subway arrival time information panels impact the arrival times and routes taken by transit passengers. It is clear from these examples that the biggest challenges and most compelling content tackled in this book lie with these last two chapters.

Despite the broadness of the topic, it can be divided into two categories. The first, covered in this chapter, deals with the range of static network design models found in the literature: first covering capacitated networks without congestion effects, then on networks with congestion effects, and lastly on networks with multiple operators. [Chapter 8](#) covers more data-driven network design problems (NDPs) that involve real-time information, online

decision-making, and anticipatory actions. Many of the examples stated previously under smart cities fall under the latter chapter, although the fundamental network optimization structures are explored in this chapter.

Before moving forward, one final comment is made about the objective of network optimization models. In the prior chapters, optimization modeling is used to get to some solution state, whether that is a descriptive equilibrium state or to support a privacy-aware data sharing mechanism. In these last two chapters, the models are supporting decisions to make changes to the network itself. It is important to stress two points. First, model formulations only describe the conditions under which a state would be considered optimal; solutions need to be obtained by applying a solution algorithm. Second, even though it is tempting to take the results directly and apply them to a system, these are still ultimately just simplified models of reality. As models, the purpose is not to provide unfettered decision support, even for automated systems that rely on these outputs, but to provide practitioners with a framework to analyze trade-offs in designing solutions. Outputs of network design models should never be treated as final decisions, but instead should serve as starting points from which other rules and policies are fitted to serve realistic requirements. This is, in fact, one of the key roles of transportation systems engineers needed even in an age of automation and data-driven decision-making.

7.2 NETWORK DESIGN PROBLEMS

What is network design? It is a process of optimizing some objective or set of objectives for a system by making changes to its network structure. Due to the dependencies between components of the system, it is characterized as a type of network $G(N,A)$ of links A and nodes N serving demand $W=O \times D \times M$ for origins O , destinations D , and commodities M . The demand is characterized by the need for some users or commodities to be transported from one node to another or to be picked up or dropped off at a node. Supply is characterized by the network G . Even nonnetwork problems can be described as networks (e.g., scheduling).

Changes to a network come in many ways. They may involve construction of links, routes, or tours. They may involve changing link capacities, link costs, link congestion effects, node locations, node service rates, or scheduled departure times of lines on a route. Changes may also be state-dependent policies as opposed to deterministic decisions. Objectives of a network design problem may involve transport costs for the route or for the commodities. They may involve arrival time reliability or may ensure users are equitably transported and have adequate access to mobility.

The decision variables may be flow of passengers, goods, energy, information, or security risks, among others.

Magnanti and Wong (1984) formulate the network design problem in the most generic mold in Eq. (7.1).

$$\min_{x, y} \phi[x, y] \quad (7.1a)$$

Subject to

$$\sum_{j \in N} x_{ij}^m - \sum_{j \in N} x_{ji}^m = \begin{cases} w_m, & i = O[m] \\ -w_m, & i = D[m], \forall m \in M \\ 0, & \text{otherwise} \end{cases} \quad (7.1b)$$

$$x_{ij} \equiv \sum_{m \in M} x_{ij}^m \leq K_{ij} y_{ij}, \forall (i, j) \in A \quad (7.1c)$$

$$(x, y) \in S \quad (7.1d)$$

$$x_{ij}^m \geq 0, y_{ij} \in \{0, 1\}, \forall (i, j) \in A, m \in M \quad (7.1e)$$

In this problem, there is an objective function ϕ in terms of two sets of decision variables: x is the set of link flows on the network and y is the set of control variables. The objective may be minimizing cost, maximizing profit, maximizing social welfare, maximizing equity, minimizing disruption risk, and so on. The flow vector x is divided into different links $(i, j) \in A$ as well as different commodities $m \in M$. w_m is demand for commodity m , K_{ij} is a link capacity, and S is a set of constraints characterizing a specific NDP. One constraint may be related to budget for resources. Another may pertain to user equilibrium route assignment behavior leading to a mathematical program with equilibrium constraints (MPEC).

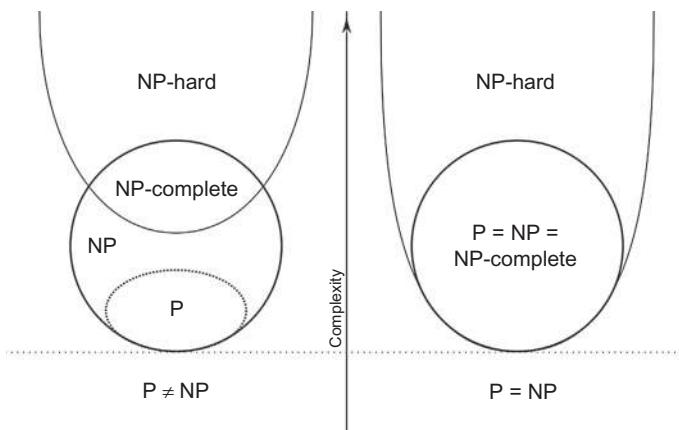
NDPs can be divided into groups of several well-known problems. Magnanti and Wong (1984) provided a detailed list to which several others are added to obtain Table 7.1.

Because network design models only present the framework of the optimization problem, it is also important to discuss solution algorithms that can be used to effectively obtain solutions. The models themselves exhibit certain structures for which certain solution methods outperform others. For example, constrained continuous optimization problems can be either linear or nonlinear, and for the nonlinear problems they can be convex or nonconvex. Discrete problems may be classified as NP-hard complexity, which means that the time to determine whether a solution is optimal is nonpolynomial time scalable. An Euler diagram of the types of problems by computational complexity is shown in Fig. 7.1.

Problems proven to be NP-hard cannot be efficiently solved exactly for any practical instance. In such cases, heuristics are needed. Since heuristics

Table 7.1 Network design problems covered in this chapter

Type	Problem	Example applications(s)
Single level	Minimum spanning tree Traveling salesman problem Vehicle routing problem Facility location problem	Railway route design Mobile traffic sensor deployment Dial-a-Ride, microtransit Bus stop location, electric vehicle charging, idle vehicle rebalancing
Bilevel	Line planning problem Discrete NDP	Transit line and frequency design Link investments, cordon location
Multiple operators	Continuous NDP Activity-based NDP Symbiotic NDP Network Design Game Privacy-Aware NDP	Toll pricing, capacity expansion Park and ride facilities, V2G Bike share, last mile service design Car-sharing companies DOT policy design using e-hail company shared data

**Fig. 7.1** Types of problems by complexity. (Source: CC Image courtesy of B. Esfahbod on Wikipedia.)

cannot ensure an optimal solution, the quality of a heuristic is determined by the combination of computational efficiency and some measure of suboptimality. The suboptimality may be an analytical worst-case bound or a computational worst-case bound. For algorithms that lack an analytical expression for this bound, tests are conducted over a range of instances to sample the algorithm effectiveness.

Table 7.2 Popular software packages for solving network design problems

Software	Link	Notes
AMPL	https://ampl.com/	Temporary versions available for students
CPLEX	https://www.ibm.com/products/ilog-cplex-optimization-studio	Free version available to academics through “Academic Initiative”
GAMS	https://www.gams.com/	This is an interface to existing solvers
Google OR-Tools	https://developers.google.com/optimization/	Free version available to academics through Academic License
Gurobi	http://www.gurobi.com/	Open source
Julia	https://julialang.org/	Trial version available
LINGO	https://www.lindo.com/index.php/products/lingo-and-optimization-modeling	
MATLAB	https://www.mathworks.com/products/matlab.html	
NumPy	http://www.numpy.org/	Python is open source
R	https://cran.r-project.org/web/views/Optimization.html	R is open source

Because NDPs can generally be classified under certain types of optimization problems, model solutions can be obtained using commercial software running generic algorithms. For example, a traveling salesman problem (TSP) can be formulated as an integer programming problem, in which case standard integer programming algorithms like Branch and Bound algorithm can be used to obtain a solution. More efficient exact algorithms customized for the TSP also exist, and a commercial solver may possess those algorithms in its library if the problem can be identified as a TSP. Some of the most popular software packages are listed in [Table 7.2](#).

7.2.1 Minimum Spanning Trees

In a complete graph (where every node is connected to every other node), a tree is a subgraph that connects a set of nodes n together such that the number of links is equal to $n - 1$ (which implies no cycles can be formed). A spanning tree is a tree that connects to all nodes in a graph. A minimum spanning tree (MST) is one that uses the minimum sum of link costs $\sum_{(i,j)} c_{ij}x_{ij}$, where c_{ij} is the cost of connecting a link (i,j) and x_{ij} is a binary variable that indicates whether an undirected link is formed in the subgraph. The MST is the simplest of the class of NDPs shown in [Magnanti and Wong \(1984\)](#).

The problem has great importance because it describes a problem of constructing a network that connects all nodes together at a minimum cost. It has been used to design telecommunications networks. In the context of transport, it can be used to design transit skeleton routes to ensure all nodes are covered. It can be used to identify critical nodes in a network based on lowest cost connectivity to other nodes in a network. It is also used in constructing tours, as we shall see later.

The problem can be formulated as an integer programming problem as shown in Eq. (7.2), where N is the set of nodes.

$$\min_x \phi = \sum_{(i,j) \in N \times N} c_{ij} x_{ij} \quad (7.2a)$$

Subject to

$$\sum_{(i,j) \in N \times N} x_{ij} = |N| - 1 \quad (7.2b)$$

$$\sum_{(i,j) \in S \times S} x_{ij} \leq |S| - 1, \forall S \subset N \quad (7.2c)$$

$$x_{ij} \in \{0, 1\}, \forall (i,j) \in N \times N \quad (7.2d)$$

Eq. (7.2b) ensures that the number of links is exactly equal to number of nodes minus one. Eq. (7.2c) ensures that every possible subtour is made infeasible. For example, a network with 4 nodes would have the following constraints (only the 3-node cases are shown since 2-node cases are already covered by Eq. (7.2d)):

$$x_{12} + x_{13} + x_{23} \leq 2$$

$$x_{12} + x_{14} + x_{24} \leq 2$$

$$x_{13} + x_{14} + x_{34} \leq 2$$

$$x_{23} + x_{24} + x_{34} \leq 2$$

This is an especially cumbersome formulation to setup to solve because it requires explicit enumeration of all possible subtours. In the case of 5 nodes, there needs to be $10 + 10 + 5 = 25$ subtours to eliminate. However, this problem is not NP-hard, and one of the simplest exact solution methods was proposed by [Prim \(1957\)](#) (for constructing a nationwide

telecommunications network) that is solvable in $O(n^2)$, as shown in [Algorithm 7.1](#).

Algorithm 7.1: Prim's (1957). Algorithm to Find the Minimum Spanning Tree

Inputs: A complete graph $G[N, A]$ with link costs c_{ij} , $(i, j) \in A$

1. Select any starting node n_1 and add to $g[n, a]$, where $n_1 = n$
2. Select the node n' closest to the spanning tree (at node n''), to join the tree: $n := n \cup n'$, $a := a \cup (n', n'')$
3. Repeat 2 until all nodes have joined tree

Outputs: Subgraph $g[n, a]$

Despite being a “greedy” algorithm that is typically a heuristic, the unique structure of this problem guarantees an optimal solution. This is illustrated in [Exercise 7.1](#).

Exercise 7.1

Given the network shown in [Fig. 7.2](#), use Prim's Algorithm to obtain a solution starting from node 4.

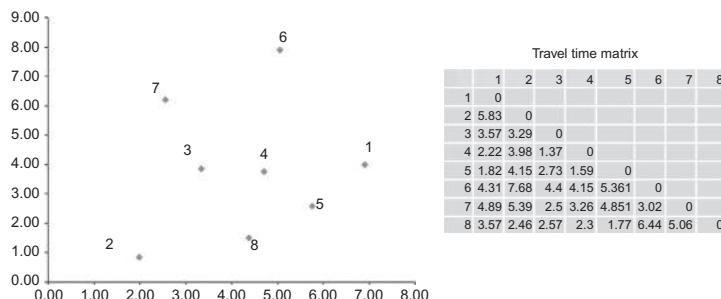


Fig. 7.2 Network used for [Exercise 7.1](#).

Starting from node 4, Prim's Algorithm has the following steps in [Fig. 7.3](#).

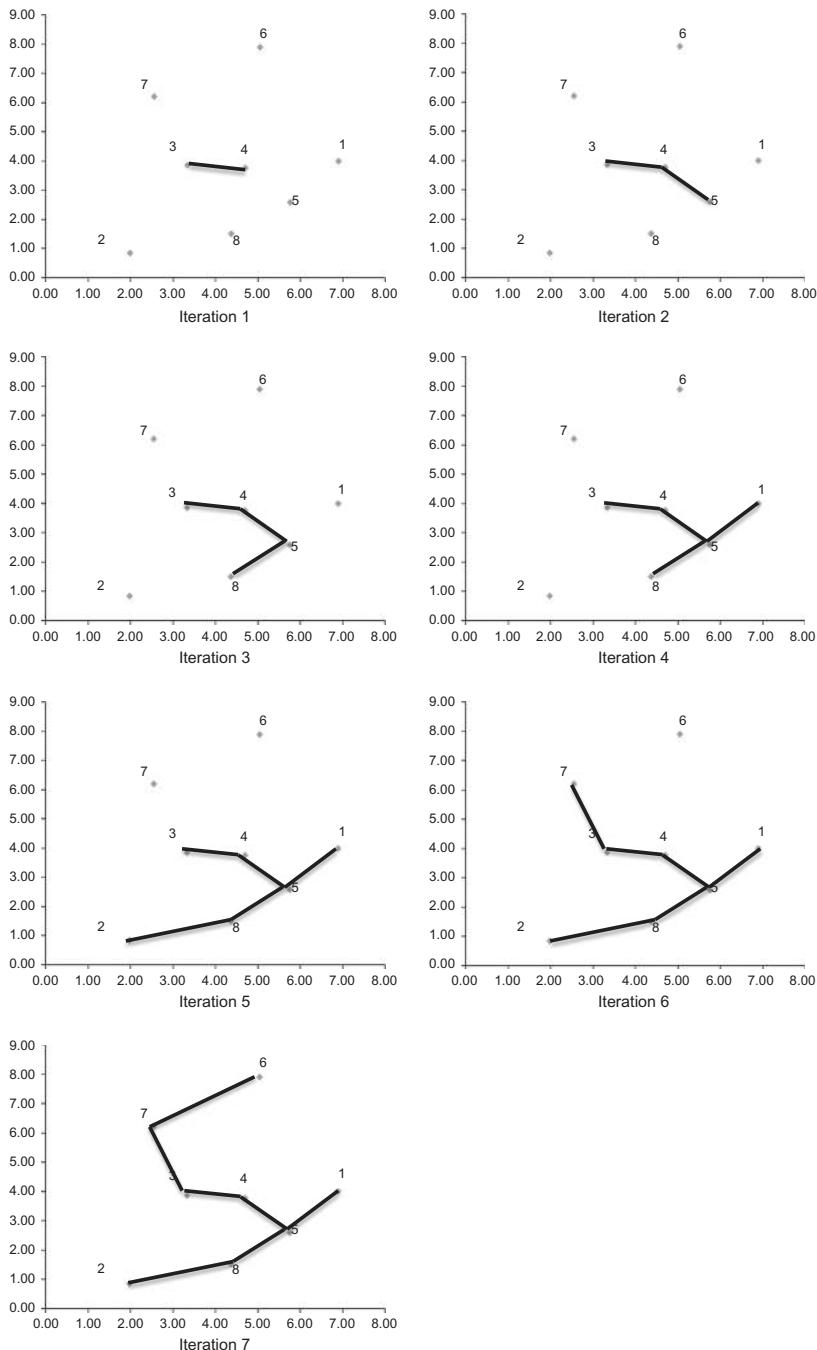


Fig. 7.3 Iterations of Prim's Algorithm for Exercise 7.1.

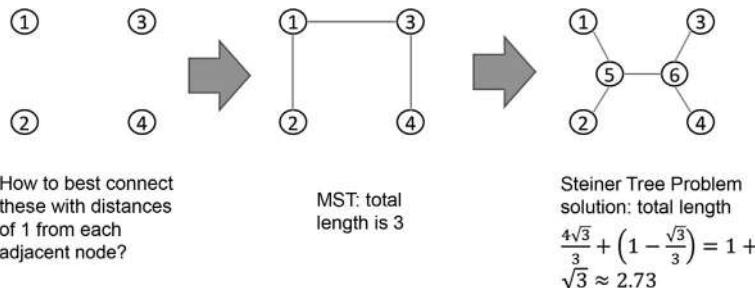


Fig. 7.4 Illustration of the Steiner Tree Problem.

Variants of the MST quickly become more complex. For example, trees with shorter lengths can be constructed if additional nodes are allowed. Consider the following example from Larson and Odoni (four nodes arranged in a square space with each node 1 unit distant from its closest two neighbors). A MST (due to symmetry there are multiple optima) is three links connecting the four nodes for a total cost of $\phi=3$. However, if additional nodes are allowed, it is possible to insert two nodes such that the total length is now $\phi=1+\sqrt{3}\approx 2.73$, as shown in Fig. 7.4. This variant is called a Steiner Tree Problem and can be used to identify new stops/stations for transit services. The problem is also known to be NP-complete (Garey and Johnson, 1977).

7.2.2 Traveling Salesman Problems

Perhaps one of the most famous transportation and computer science problems is the traveling salesman problem (TSP). The problem involves finding a minimum cost Hamiltonian cycle that begins at a given node in a network, visits all members of a specified set of nodes on the network once, and returns to the initial node. Its popularity is due to the surprising complexity of the seemingly simple problem, its numerous applications, and the basis of one of its most well-known heuristics to key concepts in graph theory. For example, the vehicle routing problem is a generalization of the TSP. Many scheduling and routing applications boil down to a TSP as a basic case. The problem is formulated as an integer programming problem by Miller et al. (1960) shown in Eq. (7.3).

$$\min_x \phi = \sum_{0 \leq i \leq n} \sum_{j \neq i} c_{ij} x_{ij} \quad (7.3a)$$

Subject to

$$\sum_{i=0, i \neq j}^n x_{ij} = 1, \forall 1 \leq j \leq n \quad (7.3b)$$

$$\sum_{i=0, i \neq j}^n x_{ji} = 1, \forall 1 \leq j \leq n \quad (7.3c)$$

$$u_i - u_j + px_{ij} \leq p - 1, \forall 1 \leq i \neq j \leq n \quad (7.3d)$$

$$x_{ij} \in \{0, 1\} \quad (7.3e)$$

$$u_i \geq 0 \quad (7.3f)$$

where n is the number of nodes to visit, p is an arbitrary number such that $p \geq n$, node 0 is the depot, and u_i is a dummy variable used for subtour elimination. Eq. (7.3b), (7.3c) ensure that one unit of flow visits every node. Eq. (7.3d) is a subtour elimination constraint applied to each potential link in the network. The constraint forces the gap between the u_i to follow a certain direction so that revisiting a prior node is no longer possible. The equation can also be modified slightly to serve as an arrival time constraint.

While exact solutions can be obtained using integer programming algorithms, the model has been shown to be NP-hard with a dynamic programming exact algorithm having a complexity of $O(n^2 2^n)$ (Held and Karp, 1962).

Due to the complexity, heuristics are needed for practical solutions. Christofides (1976) proposed one such heuristic and proved a worst-case bound for it. The heuristic is based on the construction of an “Euler tour” and knowledge of Euler’s Theorem. An Euler tour is a cycle that traverses every link on a connected graph exactly once. An Euler path is a path which traverses every link on a connected graph exactly once. The origin of Euler’s Theorem comes from 1736 when Euler tackled the “Königsberg Bridge Problem” shown in Fig. 7.5.

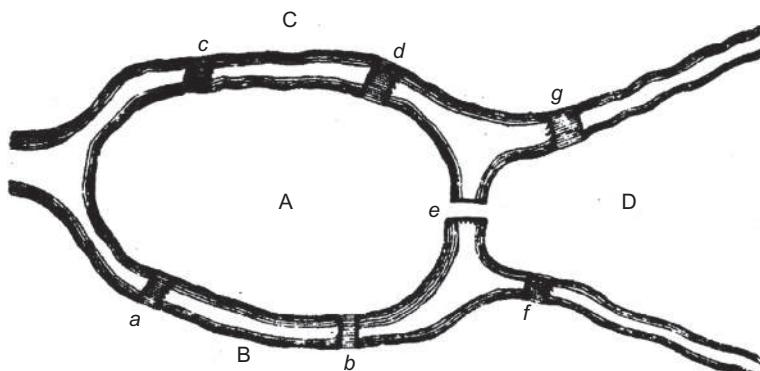


Fig. 7.5 Seven bridges of Königsberg. (Source: Wikipedia Commons: https://commons.wikimedia.org/wiki/File:The_Seven_Bridges_of_K%C3%B6nigsberg.)

The problem is to find a tour that would cross every bridge exactly once. Euler proved that there is no solution to this problem, and thus laid the foundations for graph theory.

Theorem 7.1 (Euler's Theorem). *A connected graph G possesses an Euler tour (Euler path) if and only if G contains exactly zero (exactly two) nodes of odd degree.*

The degree of a node is defined as the number of links connecting to/from it (links are undirected). Christofides proposed a heuristic algorithm that exploited this information to solve the TSP by constructing an Euler tour from a minimum spanning tree. [Algorithm 7.2](#) is shown here. Note that $L[\cdot]$ is a length operator applied to a set of links.

Algorithm 7.2: Christofides (1976). Algorithm to Find a Heuristic Solution for the TSP

Inputs: A complete graph $G(N, A)$ with link costs c_{ij} , $(i, j) \in A$

1. Find a minimum spanning tree for the set of nodes N . Call this tree T .
2. Let n_0 be the number of odd-degree nodes (which will always be even). Find a minimum length match between these nodes. Let the graph of pairwise matches be M . Define $H = M \cup T$.
3. From Euler's Theorem, H should have an Eulerian tour. Draw this tour.
4. Apply triangle inequality to remove links traversed twice.

Outputs: A tour H with worst-case bound of length $L[H] < \frac{3}{2}L[TST]$, where TST is the optimal tour.

The worst-case bound is proven by [Christofides \(1976\)](#) as follows. Suppose we know the TST. Since TST covers n nodes and visits each once, if we remove one link, we have a spanning tree (let us call it TST'). Since T is a minimum spanning tree, then the following relationship between T and TST' is true:

$$L[T] \leq L[TST'] < L[TST]$$

Now consider the set of odd-degree nodes n_0 set on TST . The case where the length of the matched odd-degree nodes (let us call that M') is maximum is if all the nodes are odd-degree. In that case, the length of all odd degree nodes is at most half of $L[TST]$: $L[M'] \leq \frac{L[TST]}{2}$. Since M is not restricted to be on TST , then $L[M] \leq L[M'] \leq \frac{L[TST]}{2}$. This is illustrated in [Fig. 7.6](#).

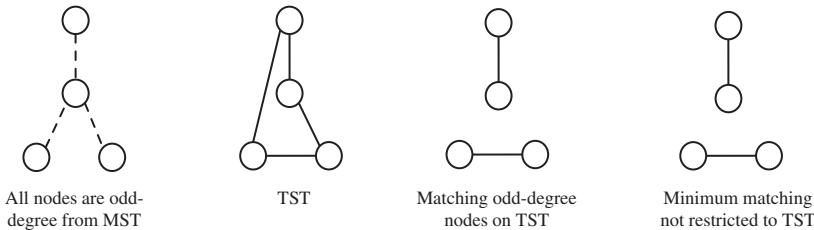


Fig. 7.6 Illustration of worst-case matched odd-degree node lengths.

Finally, since $L[H] = L[M] + L[T]$, where $L[M] \leq \frac{L[TST]}{2}$ and $L[T] < L[-TST]$, then the following is true:

$$L[H] = L[M] + L[T] < \frac{L[TST]}{2} + L[TST] = \frac{3}{2}L[TST]$$

The heuristic has a complexity of $O(n^3)$. It turns out to be a very practical method since it has a guaranteed performance bound. [Exercise 7.2](#) illustrates the algorithm and the model.

Exercise 7.2

Given the network shown in [Fig. 7.2](#), find the TST using integer programming and show that [Algorithm 7.2](#) satisfies the $\frac{3}{2}L[TST]$ worst-case bound.

The problem is small enough that it can be solved even in Excel using Solver. Note that “node 0” in the formulation in Eq. (7.3) corresponds to node 1 in [Fig. 7.2](#) and $n=7$. It takes 17 iterations of the branch and bound leading to the solution shown in [Fig. 7.7](#), which has a length of $L[TST]=21.16$.

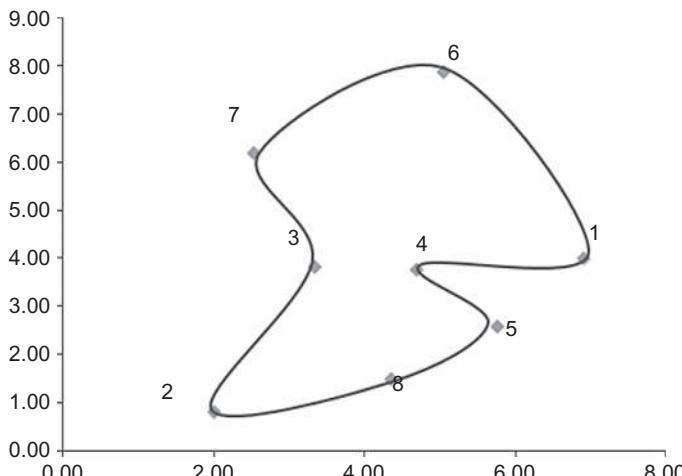


Fig. 7.7 Optimal solution to TSP using integer programming.

Now we apply the first three steps of [Algorithm 7.2](#). The first step is constructing the MST, which is shown in [Fig. 7.3](#). The second step is to identify the odd-degree nodes and to connect them pairwise in a minimal fashion. The odd-degree nodes are circled and the minimal matching subgraph is shown as the set of dashed lines in [Fig. 7.8](#).

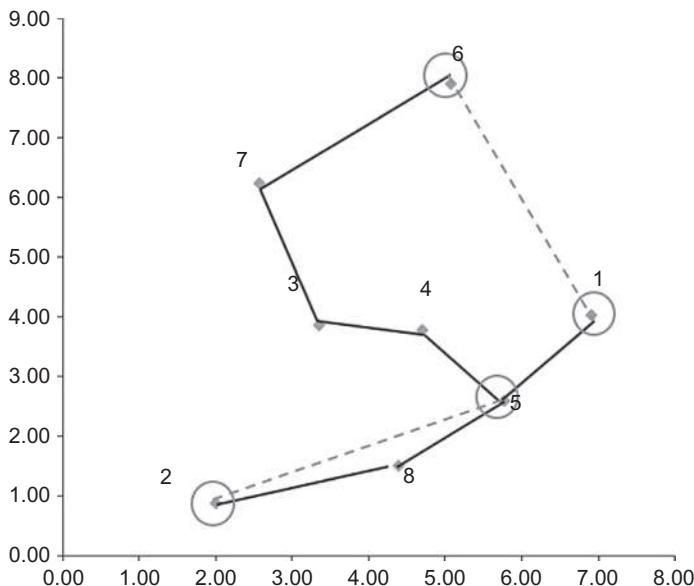


Fig. 7.8 Eulerian tour obtained using [Algorithm 7.2](#).

Based on the union of the two subgraphs, an Eulerian tour can be constructed: $H = (1, 5, 8, 2, 5, 4, 3, 7, 6, 1)$. While further improvements can be made in Step 4, Christofides' bound applies to H . As we can see, $L[H] = 22.99$, which is 108.6% of the $L[TST]$, well within the 150% bound.

Solution methods for TSP are not restricted to this heuristic or to IP solution methods. [Laporte \(1992a\)](#) provides a review of other algorithms in the literature. In addition to the Christofides' heuristic, there are various tour construction and improvement heuristics as well. Two construction heuristics are the nearest neighbor heuristic and the insertion heuristic. In the nearest neighbor heuristic, the next nearest nodes are iteratively added to the sequence without regard for where in the sequence to insert them. The insertion heuristic, on the other hand, adds nodes from a list and checks each segment of the sequence to find the lowest cost insertion. Several iterations of the nearest neighbor heuristic are illustrated using [Fig. 7.2](#) example in [Fig. 7.9](#).

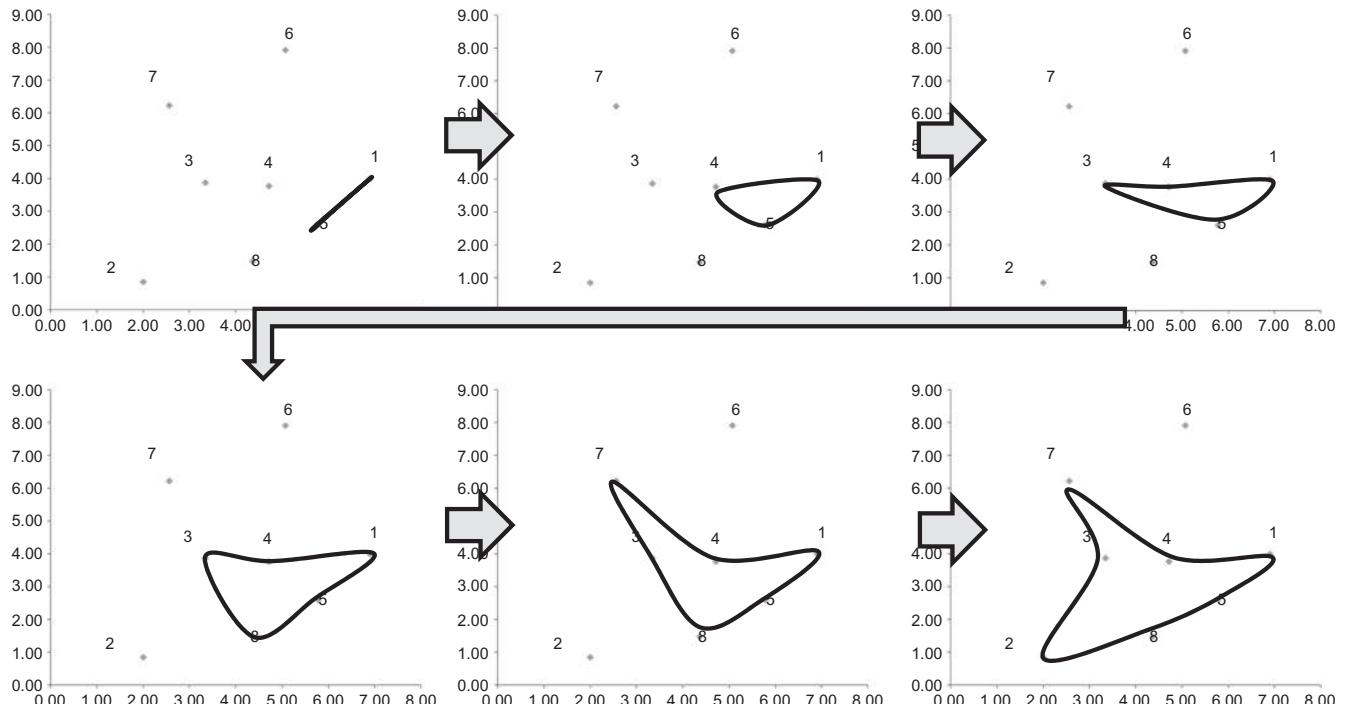


Fig. 7.9 Several iterations of an nearest neighbor heuristic on example from Fig. 7.2.

Another popular tour improvement heuristic is the k -opt heuristic (Lin, 1965; Lin and Kernighan, 1973). For a given tour, the algorithm iteratively checks k segments to consider swapping ends. For example, consider an existing tour $H=(1, 2, 3, 4, 5, 1)$. A 2-opt algorithm considers swapping (2, 3) with (4, 5). If $(1, 4, 5, 2, 3, 1)$ has a shorter length, a swap is made. This repeats through every possible combination of swaps.

There are numerous variants of TSP. One such is the m -TSP, which assigns m tours to serve the network. This can be done by creating m copies of the original depot node and linking them together with infinite cost links. The resulting problem becomes equivalent to a single-vehicle TSP and algorithms can be applied as before.

Another variant is the arc routing problem (also called the Chinese Postman Problem), which requires having a server visit every link on a connected graph at least once. This has many applications: waste pickup, parcel deliveries, and so on. Golden and Wong (1981) formulated the capacitated arc routing problem, and one of its more recent applications is in commercial drone deployment for traffic monitoring (Chow, 2016b).

Other increasingly important variants are the profitable tour problem (Feillet et al., 2005) and the generalized TSP (Srivastava et al., 1969). The profitable tour problem allows a server to visit only a subset of nodes where each visit entails some profit earned. Constraining the profits or the maximum length of the tour leads to other variants like the prize-collecting TSP and the orienteering problem. Generalized TSPs identify clusters of nodes such that visiting only one node in each cluster is sufficient. In urban mobility, activity recommendation systems for automated vehicles may operate as generalized profitable tour problems. Chow and Liu (2012) define a hierarchy of use cases based on the degree of complexity.

Use Case 1: shortest path—one origin and one destination

“How do I get to the In-n-Out at University Center, Irvine, California, for lunch?”

Use Case 2: activity selection problem—sequence is set but node needs to be selected

“Where can I get gas on my way home from work?”

Use Case 3: TSP—nodes are set but sequence needs to be selected

“What is the best sequence to visit Hollywood Boulevard, Santa Monica Pier, and Little Tokyo?”

Use Case 4: generalized prize collecting TSP—both nodes and sequence need to be set, but number of nodes is given

“I have time to visit three tourist attractions and then have Ethiopian food for dinner in Washington DC. Where should I go?”

Use Case 5: generalized profitable tour problem—number of nodes, their selection, and sequence all need to be set

“I am in Paris for a day and I want to see as many sights as possible.

Where should I go?”

For mobility services, another important variant is the TSP with pickups and drop-offs. In this case, a single vehicle needs to be routed such that each passenger has a pickup location as well as a drop-off location (Mosheiov, 1994). A feasible solution should satisfy the precedence constraints.

7.2.3 Vehicle Routing Problems

When the m -TSP includes vehicle capacities and arrival time constraints, it generalizes to a vehicle routing problem (VRP). Originally proposed as a “truck dispatching problem” (Dantzig and Ramser, 1959), the VRP has become one of the most studied topics in combinatorial optimization (Laporte, 1992b; Toth and Vigo, 2002). Many of the MaaS operations depend on solutions of VRPs or their variants to set routes, schedules, and dispatch decisions. Despite the applicability of the model, exact solution methods cannot consistently solve problems beyond 50–100 nodes or so.

A standard model formulation based on three indices (vehicle–origin–destination) for route flow based on Fisher and Jaikumar (1981) is shown in Eq. (7.4). Consider a set of nodes N for n customers with a depot at node 0 and K vehicles in the fleet.

Decision variables:

$X_{ijk} \in \{0, 1\}$ is 1 if vehicle k traverses route (i, j) , 0 otherwise

$Y_{ik} \in \{0, 1\}$ is 1 if vehicle k serves node i , 0 otherwise

$T_i \geq 0$ is arrival time at node i

Parameters:

u_k is the capacity of vehicle k

c_{ij} is the objective travel cost of route (i, j)

t_{ij} is the travel time of route (i, j)

d_i is the dwell time at node i

q_i is the demand at node i

M is an arbitrarily big number

$$\min_{X, Y, T} \phi = \sum_{i \in N} \sum_{j \neq i} \sum_{k \leq K} c_{ij} X_{ijk} \quad (7.4a)$$

Subject to

$$\sum_{i \in N \setminus 0} q_i Y_{ik} \leq u_k, \quad \forall k \leq K \quad (7.4b)$$

$$\sum_{k \leq K} Y_{ik} = 1, \forall i = 1, \dots, n \quad (7.4c)$$

$$\sum_{j \in N} X_{ijk} = Y_{ik}, \forall i \in N, k \leq K \quad (7.4d)$$

$$\sum_{j \in N} X_{jik} = Y_{ik}, \forall i \in N, k \leq K \quad (7.4e)$$

$$T_i - T_j \leq -t_{ij} - d_i + (1 - X_{ijk})M, \forall i, j = 1, \dots, n, k \leq K \quad (7.4f)$$

$$x_{ijk} \in \{0, 1\}, \forall i, j \in N, k \leq K \quad (7.4g)$$

$$Y_{ik} \geq 0, \forall i = 1, \dots, n, k \leq K \quad (7.4h)$$

$$T_i \geq 0, \forall i = 1, \dots, n \quad (7.4i)$$

Eq. (7.4b) enforces the vehicle capacity. Eq. (7.4c) makes sure every node is visited by some vehicle. Eqs. (7.4d) and (7.4e) ensure flow conservation. Eq. (7.4f) tracks the arrival times at each node and serves as subtour elimination constraints. Other constraints can also be considered. For example, hard time windows can be represented by Eq. (7.5), where (a_i, b_i) represent the time window for arrival (VRPTW).

$$a_i \leq T_i \leq b_i, \forall i = 1, \dots, n \quad (7.5)$$

Soft time windows (or goal arrival times, when $a_i = b_i$) with penalties can be implemented with the addition of the following term in the objective function: $\sum_i (p_i^e \tau_i^e + p_i^l \tau_i^l)$ and setting the following constraints in Eq. (7.6).

$$T_i + \tau_i^e \geq a_i, \forall i = 1, \dots, n \quad (7.6a)$$

$$T_i - \tau_i^l \leq b_i, \forall i = 1, \dots, n \quad (7.6b)$$

$$\tau_i^e, \tau_i^l \geq 0, \forall i = 1, \dots, n \quad (7.6c)$$

where τ_i^e and τ_i^l are the early and late deviations from the early or late time windows for each node, and p_i^e and p_i^l are the early and late penalty rates for deviation from time windows, indexed by node. For example, for user activity scheduling (see Chapter 4), late arrival to work nodes is generally much more consequential than late arrival to shopping.

Since the VRP is a generalization of the TSP, it is also NP-hard. Clarke and Wright (1964) proposed a heuristic based on a “savings method” in which pair-wise trip chains are ranked by how much they would save in travel costs. The ranked trip chains are then used to iteratively inform which node to sequence. The pair-wise chains act as a form of initial gradient to determine descent direction for the selection of next nodes. The heuristic is shown in Algorithm 7.3 and illustrated in Exercise 7.3.

Algorithm 7.3: (Clarke-Wright, 1964). Heuristic to Obtain a Solution to the VRP

Inputs: K vehicles, set of nodes N with travel costs c_{ij} , $(i,j) \in N \times N$, $i \neq j$, vehicle capacities u_k

1. Initiate with n vehicles to serve n customers directly from and back to a depot
2. For every pair of nodes (i,j) , compute savings: $s_{ij} = c_{0i} + c_{0j} - c_{ij}$ (for more complicated VRPs with other objective functions, this would have to be altered to represent the savings in the objective function)
3. Rank the node pairs (i,j) in descending order of savings s_{ij}
4. For each s_{ij} , add to a route if no route constraints are violated and if either of these three conditions are met:
 - a. Neither i nor j have already been assigned to a route, in which case a new route is initiated including both,
 - b. Exactly one of the two points (i or j) has already been included in an existing route and that point is not interior to that route, in which case link (i,j) is added to that route,
 - c. Both i and j have already been included in two different existing routes and neither point is interior to its route, in which case the two routes are merged.
5. If the list is not exhausted, return to start of Step 4. Otherwise, stop.
Any points not assigned to a route during Step 4 must be served as a direct route from node 0.

Outputs: a set of routes X_{ijk} and their computed arrival times T_i

Exercise 7.3

Given the instance in Fig. 7.10, solve the VRP for a 2-vehicle fleet using integer programming. Compare the result to the solution obtained using Algorithm 7.3.

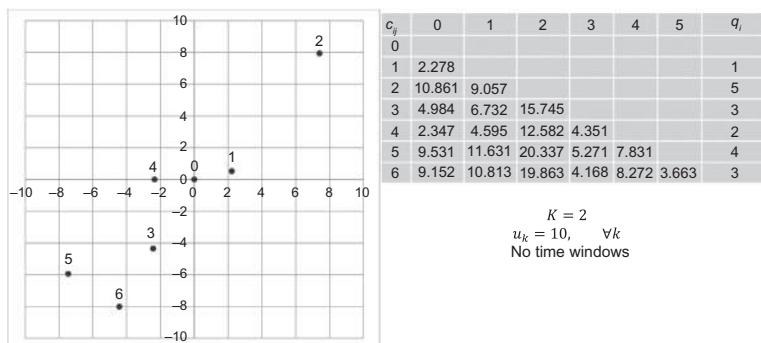


Fig. 7.10 Instance for Exercise 7.3.

Formulating this problem as an integer program as shown in Eq. (7.4), there are 84 binary variables, 20 continuous variables, 62 inequality constraints, and 35 equality constraints. The solution is shown in Fig. 7.11. The objective value is $\phi = 48.61$.

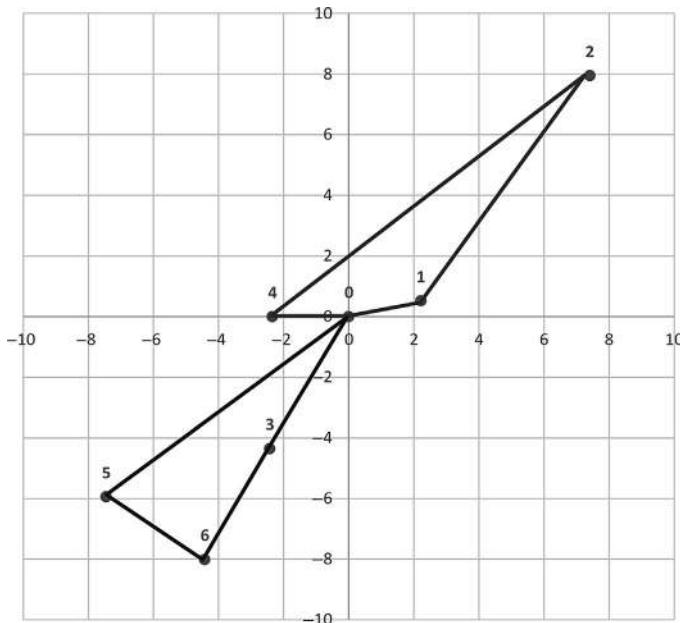


Fig. 7.11 Optimal solution obtained by integer programming.

Algorithm 7.3 is demonstrated. First, for each pair of nodes the savings is computed and sorted in descending order as presented in Table 7.3.

Table 7.3 Sorted savings for all node pairs in descending order

(i,j)	c_{0i}	c_{0j}	c_{ij}	s_{ij}
(5,6)	9.531	9.152	3.663	15.02
(3,6)	4.984	9.152	4.168	9.968
(3,5)	4.984	9.531	5.271	9.244
(1,2)	2.278	10.861	9.057	4.082
(4,5)	2.347	9.531	7.831	4.047
(4,6)	2.347	9.152	8.272	3.227
(3,4)	4.984	2.347	4.351	2.98
(2,4)	10.861	2.347	12.582	0.626

Continued

Table 7.3 Sorted savings for all node pairs in descending order—contd

(i,j)	c_{oi}	c_{oj}	c_{ij}	s_{ij}
(1,6)	2.278	9.152	10.813	0.617
(1,3)	2.278	4.984	6.732	0.53
(1,5)	2.278	9.531	11.631	0.178
(2,6)	10.861	9.152	19.863	0.15
(2,3)	10.861	4.984	15.745	0.1
(2,5)	10.861	9.531	20.337	0.055
(1,4)	2.278	2.347	4.595	0.03

1. Construct route 1: (5,6), with load of $4+3=7$
2. Add node 3 to route 1: (5,6,3), with load $7+3=10$. This route is at capacity.
3. Skip (3,5) since both nodes are already added.
4. Create new route 2: (1,2) with load $1+5=6$.
5. Skip (4,5), (4,6), (3,4): they all have one node that is not feasible to append to an existing route.
6. Add node 4 to route 2: (1,2,4) with load $6+2=8$. All nodes are added now.

The solution using [Algorithm 7.3](#) is optimal in this case.

There are a great number of variations on the VRP and its formulation. For example, [Balinski and Quandt \(1964\)](#) proposed a set partitioning formulation that is based on enumerated tours instead of constructing them from routes. This is expressed in Eq. (7.7) where J is the set of all possible routes for n nodes.

$$\min_{\gamma} \phi = \sum_{j \in J} c_j \gamma_j \quad (7.7a)$$

Subject to

$$\sum_{j \in J} a_{ij} \gamma_j = 1, \quad \forall i = 1, \dots, n \quad (7.7b)$$

$$\sum_{1 \leq i \leq n} a_{ij} q_i \gamma_j \leq u_j, \quad \forall j \in J \quad (7.7c)$$

$$\gamma_j \in \{0, 1\} \quad (7.7d)$$

In this formulation, c_j is the cost of operating tour $j \in J$, γ_j is equal to 1 if tour j is operated, u_j is the vehicle capacity, q_i is the demand at node i , and a_{ij} is an indicator parameter set to 1 if tour j covers node i . Eq. (7.7b) ensures that every node is covered by one tour. Eq. (7.7c) is the vehicle capacity constraint. Arrival times are predefined by the tours.

Although this alternative formulation requires tour enumeration, there are implicit enumeration methods that can efficiently generate a restricted tour set that converges in a finite number of iterations. One such approach is column generation with shortest path subproblems for m -TSP (Desrosiers et al., 1984) and for VRPTW and pickup and delivery problems (PDPTW) (Dumas et al., 1991).

Other algorithms include sweep algorithms and cluster first, route second (or vice versa) algorithms.

One of the most relevant variants of the VRP is the dial-a-ride problem (DARP), which is a subset of the pickup and delivery problem (PDP). DARP was first studied by Wilson et al. (1969), which they called CARS (Computer Aided Routing System) at the time. A formulation of DARP is shown in Eq. (5.4) in Chapter 5 already, so it is not included here. To illustrate DARP, consider Exercise 7.4.

Exercise 7.4

Given the same network in Fig. 7.10, assume now that the six nodes correspond to three individual passengers with pickups at $(1, 2, 3)$ and corresponding drop-offs at $(4, 5, 6)$. If vehicle capacity is 2 and fleet size is 2, $\alpha = \beta = 0$, $\gamma = 1$, $d_i = 0.5$, and $L = 1.5$, determine the optimal solution using integer programming.

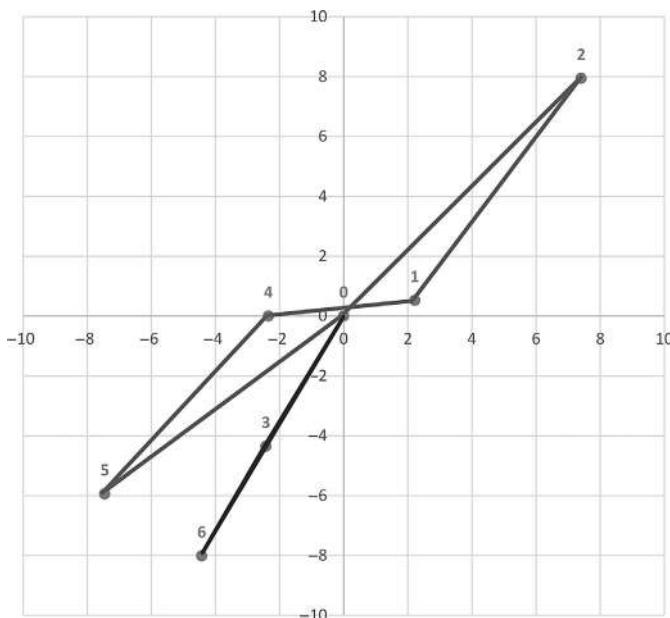


Fig. 7.12 Optimal solution to DARP example.

There are now 72 binary variables, 18 continuous variables, 23 equality constraints, and 156 inequality constraints. Excel Solver is unable to solve this problem, although MATLAB (through the `intlinprog` function on default options) can solve the problem. The solution is shown in Fig. 7.12 with a final objective value of $\phi=60.18$ with one vehicle serving route $(0, 2, 1, 4, 5, 0)$ and a second vehicle serving route $(0, 3, 6, 0)$.

A few other VRP variants are worth mentioning here. In parallel with the profitable tour problems, there are also selective VRPs (see Allahviranloo et al., 2014; Chow, 2016b). In these problems, only a subset of nodes may be visited depending on the profit earned. A second variant that is highly applicable to MaaS setting is the VRP with transfers (Cortés et al., 2010). As more multimodal services are implemented, transfers for passengers will become a crucial trade-off consideration. Related to this variant is the VRP with split deliveries, where the demand at a node can be met by multiple vehicles (Dror and Trudeau, 1989). When considering VRP over time (or multiple periods) with inventory costs, the inventory routing problem (Dror et al., 1985) is an important variant with applications in user activity scheduling (Chow and Nurumbetova, 2015) and sensor deployment (Chow, 2016b).

7.2.4 Facility Location Problems

The facility location problem deals with locating supply nodes in a network to serve nearby demand nodes in a way that minimizes access costs. Like the VRP, facility location problems have numerous applications in economics (locating businesses), emergency response, transport (idle vehicles, transit stations, freight terminals), sensor deployment, among others.

Also like the VRP, there are many different subclasses of facility location problems. Owen and Daskin (1998) provide a comprehensive review of the history and taxonomy of these problems. Its early foundations emerged from graph theory, where Hakimi (1964) showed that location problems can be solved by finding the solutions on the nodes of the connected graph (as opposed to anywhere in the space of the connected graph).

Theorem 7.2 (Hakimi, 1964). *An absolute median of a connected graph is always at a vertex.*

This understanding encouraged the study of location problems using graph theory and integer programming. Different subclasses of problems emerged.

1. Median problems—minimize the average distance traveled by users to get to the closest facility
2. Coverage problems—maximize coverage for a set maximum distance traveled by users to the closest facility
3. Center problems—minimize the maximum distance traveled by users to get to the closest facility

The p-median problem is shown as an integer program for a set of nodes N in Eq. (7.8) based on notation from [ReVelle and Swain \(1970\)](#).

x_j is 1 if locate at node j , 0 otherwise

γ_{ij} is 1 if demand at node i is served by node j , 0 otherwise

h_i is the demand at node i

d_{ij} is the distance between node i and j

P is the number of facilities

$$\min_{x, y} \phi = \sum_{i \in N} \sum_{j \in N} h_i d_{ij} \gamma_{ij} \quad (7.8a)$$

Subject to

$$\sum_{j \in N} x_j = P \quad (7.8b)$$

$$\sum_{j \in N} \gamma_{ij} = 1 \quad (7.8c)$$

$$\gamma_{ij} - x_j \leq 0, \forall i \in N, j \in N \quad (7.8d)$$

$$x_j, \gamma_{ij} \in \{0, 1\} \quad (7.8e)$$

On general networks the problem is NP-complete. For fixed values of P , the problem can be solved in polynomial time since there are $\binom{|N|}{P}$ combinations (see [Owen and Daskin, 1998](#)). Nevertheless, the computational burden can be costly, and as a result, heuristics have been introduced to solve p-median problems more efficiently. One greedy heuristic was introduced by [Teitz and Bart \(1968\)](#) and improved upon by [Larson and Odoni \(1981\)](#), which uses the information from updated 1-median solutions to iteratively insert facilities to the solution set. The heuristic is shown in [Algorithm 7.4](#) and illustrated in [Exercise 7.5](#).

Algorithm 7.4: (Teitz and Bart, 1968; Larson and Odoni, 1981).**Greedy Heuristic for p-Median Problem**

Inputs: a graph $G(N, A)$ with demand h_i , distances d_{ij} , and P facility budget

1. Let $m=1$. Find 1-median of the graph:
 - a. Multiply the i th row of the distance matrix by h_i to obtain $h_i d_{ij}$ matrix
 - b. For each column j , compute sum of all terms in column. The node j^* corresponding to the column with the minimum sum of terms is the location for the 1-median.
 Let 1-median be set at node j^* : set $x_{j^*}=1$
2. Facility addition. Add a new facility by choosing among the nodes where $x_j=0$ which maximizes the possible improvement in the objective function $\sum_{i \in N} \sum_{j \in N} h_i d_{ij} y_{ij}$. Let that be j^* , and set $x_{j^*}=1$. Let $m=m+1$.
3. Solution improvement. Consider substitution, one at a time, of each node in S with a node that is not in S .
4. If $m=P$, stop. Otherwise, go to step 2.

Outputs: x_j , y_{ij}

There are also the center-based location problems. One example is the set covering problem shown in Eq. (7.9) for a graph with node set N .

x_j is 1 if locate at node $j \in N$, 0 otherwise

c_j is the fixed cost of locating a facility at node j

d_{ij} is the distance between node i and j

s is the maximum acceptable service distance

N_i is the set of nodes j within an acceptable distance from node i , that is, $N_i = \{j \mid d_{ij} \leq s\}$

$$\min_x \phi = \sum_{j \in N} c_j x_j \quad (7.9a)$$

Subject to

$$\sum_{j \in N_i} x_j \geq 1, \quad \forall i \quad (7.9b)$$

$$x_j \in \{0, 1\} \quad (7.9c)$$

The formulation does not explicitly show coverage—it is hidden behind the definition of N_i . This means when setting this problem up, the set of N_i needs to be determined based on a given value of s for each node i . The larger the value of s , the more nodes a facility can cover, and as a result the less facilities needed to cover all nodes. This is illustrated in [Exercise 7.6](#).

Exercise 7.5

For the instance shown in Fig. 7.13, compare the integer programming solution to the p -median problem for $P=3$ and $P=2$. Solve the 3-median problem using Algorithm 7.4 and compare.

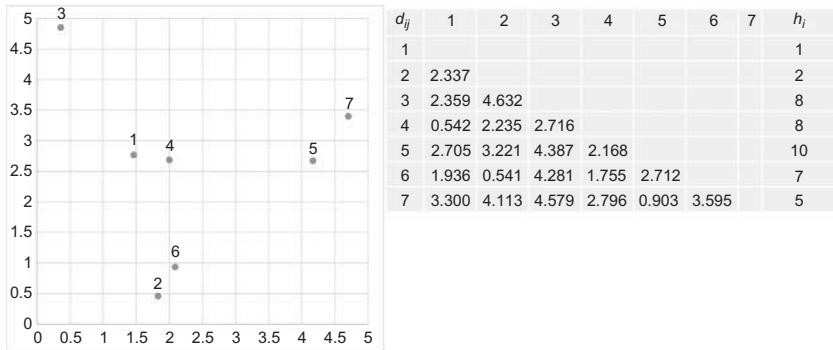


Fig. 7.13 Instance for Exercise 7.5.

The problem is formulated as an integer program for the two cases and solved using Excel Solver. The solutions are shown in Fig. 7.14, where the arrows are used to indicate the y_{ij} coverage variables and the circles are used to indicate the location decisions x_j . For $P=2$, $\phi=43.53$, and when $P=3$, $\phi=21.57$.

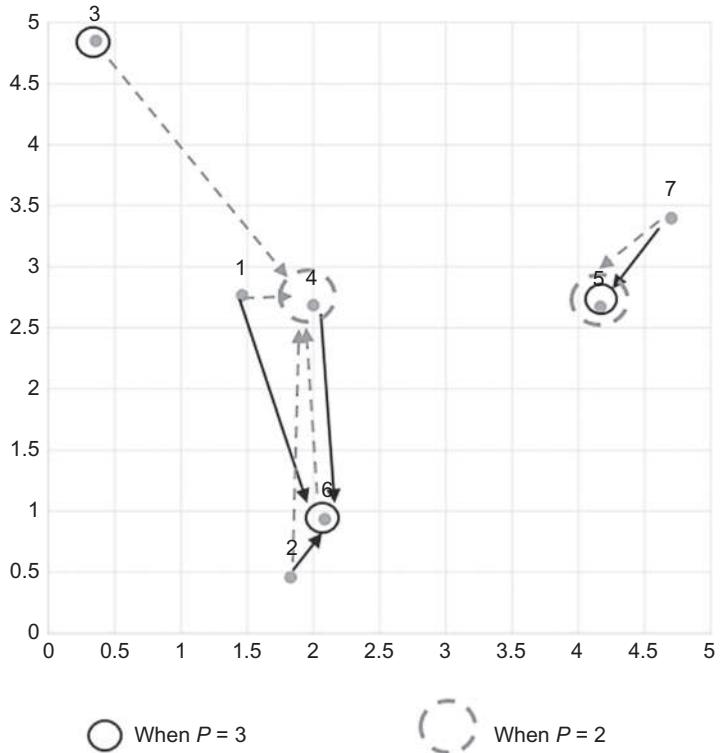


Fig. 7.14 Optimal solution via integer programming for two P values.

The solutions indicate that even switching from two facilities to three facilities can significantly alter the optimal configuration of the facilities. This illustrates why [Algorithm 7.4](#) is a heuristic that does not guarantee an optimal solution.

Using [Algorithm 7.4](#), three iterations are made. We start with locating the first facility, as indicated by the column with the bolded sum.

$h_i d_{ij}$	1	2	3	4	5	6	7
1	0.00	2.34	2.36	0.54	2.71	1.94	3.30
2	4.67	0.00	9.26	4.47	6.44	1.08	8.23
3	18.87	37.05	0.00	21.73	35.09	34.25	36.64
4	4.33	17.88	21.73	0.00	17.34	14.04	22.37
5	27.05	32.21	43.87	21.68	0.00	27.12	9.03
6	13.55	3.79	29.97	12.28	18.98	0.00	25.16
7	16.50	20.56	22.90	13.98	4.51	17.97	0.00
Sum	84.98	113.83	130.08	74.68	85.07	96.40	104.72

Summing across each column, the lowest objective value is obtained with a facility at node 4. We remove that column and update the table with $h_i \min[d_{ij}, d_{i4}]$.

$h_i \min [d_{ij}, d_{i4}]$	1	2	3	4	5	6	7
1	0.00	0.54	0.54		0.54	0.54	0.54
2	4.47	0.00	4.47		4.47	1.08	4.47
3	18.87	21.73	0.00		21.73	21.73	21.73
4	0.00	0.00	0.00		0.00	0.00	0.00
5	21.68	21.68	21.68		0.00	21.68	9.03
6	12.28	3.79	12.28		12.28	0.00	12.28
7	13.98	13.98	13.98		4.51	13.98	0.00
Sum	71.28	61.71	52.95		43.53	59.01	48.05

For the third facility, we update the table with $h_i \min[d_{ij}, d_{i4}, d_{i5}]$.

$h_i \min [d_{ij}, d_{i4}, d_{i5}]$	1	2	3	4	5	6	7
1	0.00	0.54	0.54			0.54	0.54
2	4.47	0.00	4.47			1.08	4.47
3	18.87	21.73	0.00			21.73	21.73
4	0.00	0.00	0.00			0.00	0.00
5	0.00	0.00	0.00			0.00	0.00
6	12.28	3.79	12.28			0.00	12.28
7	4.51	4.51	4.51			4.51	0.00
Sum	40.14	30.57	21.81			27.86	39.02

Using [Algorithm 7.4](#), the solution is $x_3 = x_4 = x_5 = 1$, which has an objective value of $\phi = 21.81$, 1.1% higher than the optimum.

Exercise 7.6

For the network in Fig. 7.13, assume that the fixed cost of all nodes is $c_j = 1$. Compare the solution when $s=1$ and when $s=2$.

When $s=1$, we have the following sets:

$$\begin{aligned} N_1 &= \{1, 4\}, N_2 = \{2, 6\}, N_3 = \{3\}, N_4 = \{1, 4\}, N_5 = \{5, 7\}, \\ N_6 &= \{2, 6\}, N_7 = \{5, 7\} \end{aligned}$$

When $s=2$, the sets change:

$$\begin{aligned} N_1 &= \{1, 4, 6\}, N_2 = \{2, 6\}, N_3 = \{3\}, N_4 = \{1, 4, 6\}, N_5 = \{5, 7\}, \\ N_6 &= \{2, 4, 6\}, N_7 = \{5, 7\} \end{aligned}$$

The solutions are shown in Fig. 7.15. When $s=1$, the optimal solution is to locate at $\{1, 2, 3, 5\}$, and when the threshold increases to $s=2$, the solution changes to $\{3, 5, 6\}$.

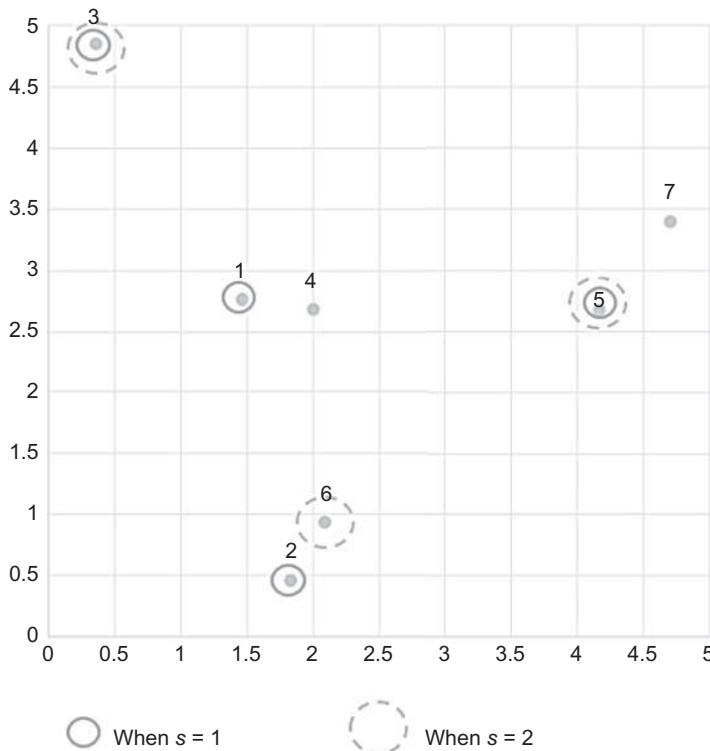


Fig. 7.15 Optimal solutions to set covering problem with $s=1, s=2$.

For urban areas with many demand nodes, it is not always cost effective to provide 100% coverage as required in the set covering problem. An alternative model is the maximal covering location problem (MCLP) proposed by [Church and ReVelle \(1974\)](#). In this model, the decision-maker is bounded by a budget on the number of facilities to deploy with the objective of maximizing coverage for a given distance threshold.

The model formulation is shown in Eq. (7.10).

x_j is 1 if a facility is located at node j , 0 otherwise

γ_i is 1 if a node i is covered, 0 otherwise

h_i is the demand at node i

P is the number of facilities available

$$\max_{x, \gamma} \phi = \sum_{i \in N} h_i \gamma_i \quad (7.10a)$$

Subject to

$$\gamma_i - \sum_{j \in N_i} x_j \leq 0, \quad \forall i \in N \quad (7.10b)$$

$$\sum_{j \in N} x_j \leq P \quad (7.10c)$$

$$x_j \in \{0, 1\}, \quad j \in N \quad (7.10d)$$

$$\gamma_i \in \{0, 1\}, \quad i \in N \quad (7.10e)$$

Eq. (7.10b) ensures that coverage is only possible if a facility within the threshold of a demand node is opened for service. Eq. (7.10c) is the budget constraint. N_i is defined in the same way as in Eq. (7.9) as $N_i = \{j \mid d_{ij} \leq s\}$. An illustration of the model is shown in [Exercise 7.7](#).

Exercise 7.7

For the instance in Fig. 7.13, considering $s = \{1, 2\}$ and $P = \{2, 3\}$, show how the solution of located facilities differs using the MCLP formulation.

The integer programming problem is solved for each of the four cases and presented in [Table 7.4](#).

The solutions show how sensitive the model is to threshold definitions and budgetary constraints.

Table 7.4 Solutions to MCLP for [Exercise 7.7](#)

s	P	$x_i = 1$	ϕ
1	2	{1, 5}	24
2	2	{5, 6}	33
1	3	{1, 2, 5}	33
2	3	{3, 5, 6}	41

The MCLP is known to be NP-hard ([Megiddo et al., 1983](#)) on general networks. [Church and Velle \(1974\)](#) proposed a greedy algorithm and a branch and bound algorithm for the integer programming formulation. [Geoffrion and Bride \(1978\)](#) proposed a Lagrangian relaxation method that has become widely used in location problems.

Location problems are highly applicable to relocation or rebalancing of empty or idle servers. For example, emergency services like positioning fire engines can improve their service times using relocation models ([Kolesar and Walker, 1974](#)). Airtankers for wildfires make use of relocation strategies ([Chow and Regan, 2011a](#)). Taxis need to relocate to serve new customers ([Sayarshad and Chow, 2017](#)). While relocation problems can be highly complex to involve look-ahead and real-time data, at the core it is about a fundamental trade-off between improving coverage/service by repositioning servers versus taking on the cost of the relocation.

The basic problem is quite similar to a standard location problem. One difference is that when the model is run, servers are already located on the network under a certain configuration. Based on new demand, a p-median objective may involve replacing Eq. [\(7.8a\)](#) with Eq. [\(7.11\)](#).

$$\min_{x, y, w} \phi = \sum_{i \in N} \sum_{j \in N} h_i d_{ij} y_{ij} + \theta \sum_{i \in N} \sum_{j \in N} r_{ij} w_{ij} \quad (7.11)$$

In this objective, the w_{ij} is a flow of idle servers from current locations x_i^0 to new locations x_j with relocation costs r_{ij} and a conversion factor θ to compare against service coverage costs. y_{ij} is a binary variable for whether a node i with demand h_i is covered by node j at distance d_{ij} . In addition to this change in objective, new transportation problem constraints need to be added as shown in Eq. [\(7.12\)](#). The relocation problem is illustrated in [Exercise 7.8](#).

$$\sum_{j \in N} w_{ij} = x_i^0, \quad \forall i \in N \quad (7.12a)$$

$$\sum_{i \in N} w_{ij} = x_j, \quad \forall j \in N \quad (7.12b)$$

Exercise 7.8

For the instance in Fig. 7.13, assume the demand is for the prior time interval with a current deployment of $x_{4t}=x_{5t}=1$. If the demand shows that it has now changed to the following, compare the solution with and without relocation costs (set $\theta r_{ij}=10$ for all (i,j)).

Node:	1	2	3	4	5	6	7
h_{it}	1	2	8	8	10	7	5
$h_{i,t+1}$	3	4	13	5	7	10	3

The values $x_{4t}=x_{5t}=1$ are treated as $x_{4,t+1}^0=x_{5,t+1}^0=1$. Solving the relocation problem without and with relocation costs using Excel Solver, the solution is presented in Table 7.5.

Table 7.5 Server locations at time t and $t+1$ (without and with relocation costs)

Node:	1	2	3	4	5	6	7
x_{jt}	0	0	0	1	1	0	0
$x_{j,t+1}$ ($r_{ij}=0$)	0	0	1	0	0	1	0
$x_{j,t+1}$ ($\theta r_{ij}=10$)	0	0	1	1	0	0	0

These results illustrate the sensitivity of the relocation modeling to different relocation costs relative to coverage costs. Without relocation costs, the decision-maker is free to locate the servers anywhere in the new time step. With relocation costs, however, it is more optimal to leave the server at node 4 in place.

$$w_{ij} \geq 0 \quad (7.12c)$$

There are many other variants to facility location problems. Location problems can be combined with routing problems as location routing problems (Perl and Daskin, 1985). In such problems the routing depends on the location of the depot(s), and vice versa. In transit route design, a problem called the maximum covering shortest path problem considers design of shortest paths such that the nodes covered by the shortest path also cover nearby demand nodes. Similar to that problem is the covering salesman problem in which tours are designed such that each node that they cover also covers nearby demand nodes (Current and Schilling, 1989).

For locating refueling or recharging stations, the flow interception problem assumes demand is not from nodes but from shortest paths

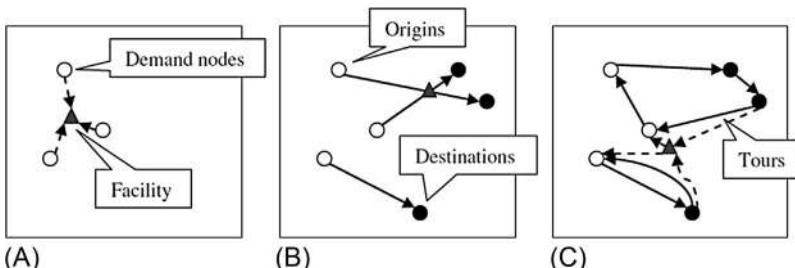


Fig. 7.16 Facility location based on (A) nodes, (B) flow, and (C) itinerary intercept. (Source: [Jung et al., 2014](#).)

between OD pairs. In this case, flow interception locates facilities that “intercept” as many paths as possible (Hodgson, 1990; Berman et al., 1992). Kim and Kuby (2012) relaxed the coverage requirement so that paths between OD pairs can deviate in a minimal manner to be served by the facilities. In Jung et al. (2014), a classification of different types of interception is made as shown in Fig. 7.16. The most complex version, itinerary interception, is tackled in Kang et al. (2013) and Kang and Recker (2014). Jung et al. (2014) solved the itinerary interception as a simulation-based optimization problem.

One last area of facility location that requires some discussion is the matter of queueing. So far, the location problem formulations presented all assume one facility can cover all demand without any capacity or congestion delay. This is not the case for many services: emergency medical services, idle taxis or bikeshare, and so on. In addition, queueing costs can be used to approximate future operating costs for dynamic relocation models with look-ahead (Sayarshad and Chow, 2017). When considering queueing, it implies that more than one server can be colocated at a node and that one server may be busy serving one or more customers when a new customer requires service. Facility location with queueing differs from multiserver queueing network analysis as the latter ignores coverage requirements for demand nodes.

Queue delay is a well-known issue, but earlier attempts to address it explicitly are computationally expensive. One example is the hypercube queueing model (Larson, 1974) which is a finite state continuous time Markov process. Berman and Odoni (1982) studied the relocation problem under this setting, modeling the service time as a generalized distribution (M/G/1 queue) (Berman et al., 1985).

Daskin (1983) proposed a simpler model extension of the MCLP that includes a likelihood (exogenously defined, however) of being busy for each server, called the maximum expected covering location model (MEXCLP). Marianov and ReVelle (1994, 1996) endogenized the likelihood by formulating equivalent integer linear programming problems. Marianov and Serra (2002) proposed a set covering version of the problem, which Sayarshad and Chow (2017) adapted to a median-based problem. That mixed integer linear programming formulation is shown in Eq. (7.13) as a relocation problem with queueing delay.

y_{ij} is 1 if customer arrivals in node i are served by node j

x_{jm} is 1 if there is an m th server located at node j

w_{ij} is the flow of servers from node i to node j

s_i is a dummy variable for the surplus of servers based in the current idle server configuration x_j^0

d_j is a dummy variable for the demand of servers due to the current server locations x_j^0

h_i is the arrival rate at node i assumed to follow a Poisson distribution

μ_j is a service rate for a server at node j , where the service time is assumed to be under an exponential distribution

c_{ij} is the access cost of a customer at node i to a server at node j

r_{ij} is the cost of relocating an idle server from node i to node j

θ is a conversion factor

C_j is the maximum possible number of vehicles at node j

P is the number of idle servers

$$\min_{x, y, w, d, s} \phi = \sum_{i \in N} \sum_{j \in N} h_i c_{ij} y_{ij} + \theta \sum_{i \in N} \sum_{j \in N} r_{ij} w_{ij} \quad (7.13a)$$

Subject to

$$\sum_j y_{ij} = 1, \quad \forall i \in N \quad (7.13b)$$

$$x_{jm} \leq x_{j, m-1}, \quad \forall j \in N, \quad m = 2, \dots, C_j \quad (7.13c)$$

$$y_{ij} \leq x_{j1}, \quad \forall i, j \in N \quad (7.13d)$$

$$\sum_{j \in N} \sum_{m=1}^{C_j} x_{jm} = P \quad (7.13e)$$

$$\sum_{j \in N} w_{ij} = s_i, \quad \forall i \in N \quad (7.13f)$$

$$\sum_{i \in N} w_{ij} = d_j, \quad \forall j \in N \quad (7.13g)$$

$$-d_j + \sum_{m=1}^{C_j} x_{jm} \leq x_j^0, \quad \forall j \in N \quad (7.13h)$$

$$-s_i - \sum_{m=1}^{C_j} x_{jm} \leq -x_j^0, \quad \forall j \in N \quad (7.13i)$$

$$\sum_{i \in N} h_i y_{ij} \leq \mu_j \left(x_{j1} \rho_{\alpha j 1} + \sum_{m=2}^{C_j} x_{jm} (\rho_{\alpha jm} - \rho_{\alpha j, m-1}) \right), \quad \forall j \in N \quad (7.13j)$$

$$x_{jm}, y_{ij} \in \{0, 1\} \quad (7.13k)$$

$$d_j, s_j, w_{ij} \geq 0 \quad (7.13l)$$

The original objective (and the measure to evaluate solutions by) should be Eq. (7.14). However, this objective is nonconvex. Eq. (7.13) circumvents the nonconvexity by acknowledging that intensity values are constant for a given number of servers and desired reliability measures α and b as shown in Eq. (7.15). The value of ρ can be solved for every possible value of m prior to setting up the model by minimizing ρ such that Eq. (7.15) remains satisfied.

$$\min_{x, y, w, d, s} \phi = \sum_{i \in N} \sum_{j \in N} h_i c_{ij} y_{ij} + \theta \sum_{i \in N} \sum_{j \in N} r_{ij} w_{ij} + \sum_{j \in N} \frac{\sum_i h_i y_{ij}}{\mu_j \sum_m x_{jm} - \sum_i h_i y_{ij}} \quad (7.14)$$

$$\sum_{k=0}^{m-1} \left(\frac{(m-k)m!m^b}{k!} \right) \left(\frac{1}{\rho^{m+b+1-k}} \right) \geq \frac{1}{1-\alpha} \quad (7.15)$$

Eq. (7.13b) ensures each demand node is served. Eq. (7.13c) requires that an m th server is located before the $(m+1)$ th is located there. Eq. (7.13d) requires that there is at least one server at node j before it can cover any nodes. Eq. (7.13e) is a budget constraint. Eqs. (7.13f)–(7.13i) are the transportation problem constraints for relocation. Eq. (7.13j) is a recursive, piecewise linearized computation of the intensity constraint for queueing delay. After a solution is obtained, its performance is measured using Eq. (7.14). An illustration is shown in [Exercise 7.9](#).

Exercise 7.9

(from Sayarshad and Chow, 2017). Consider an application with idle carshare relocation. A complete graph of four nodes with $P=2$, $\theta=0.2$, $x^0=(1, 1, 0, 0)$, relocation costs $r_{21}=0.043$, $r_{31}=0.032$, $r_{41}=0.049$, $r_{32}=0.030$, $r_{42}=0.073$, $r_{43}=0.077$, and service distances of $c_{21}=0.950$, $c_{31}=1.265$, $c_{41}=0.638$, $c_{32}=0.773$, $c_{42}=1.473$, $c_{43}=0.950$. The passenger arrival rates are $h=(4, 3, 5, 6)$. Suppose node 4 is shown to present the most opportunity for located servers to serve customers, due to a combination of customer origin-destination patterns (e.g., node 4 may have many short trips that start and end near node 4), resulting in the following service rates: $\mu=(8, 10, 9, 25)$. Compare objective value of Eq. (7.14) for the relocation model without queue delay and the model with queue delay ($\alpha=0.95$, $b=0$) in Eq. (7.13).

For the queue delay consideration, the values of ρ_{ajm} in Eq. (7.13j) need to be computed beforehand. For example, since $P=2$, we can solve $m=1$ and $m=2$:

$$m=1, \alpha=0.95, b=0 : \rho=0.2236$$

$$m=2, \alpha=0.95, b=0 : \rho=0.3162$$

These values of ρ are then input to Eq. (7.13). The solutions are shown in Fig. 7.17.

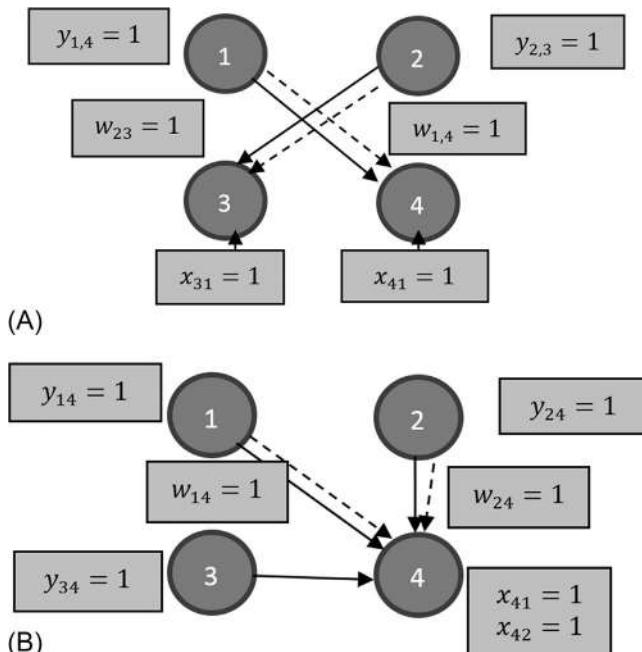


Fig. 7.17 Solutions to (A) relocation ignoring queue delay and (B) relocation with queue delay.

When ignoring queue delay, the objective is $\phi = 13.55$, of which the realized queue delay makes up a cost of 8.67. In this solution, the decision-maker does not anticipate the higher service rate for trips being made at node 4 and decides that the immediate costs (4.89) are more important, resulting in keeping one vehicle at node 3 and one at node 4.

When queue delay is accounted for, the objective value is now $\phi = 12.31$, of which queue delay cost is only 0.56 and most of the cost is due to immediate costs borne by the fleet (11.75). This is due to anticipating that node 4 will tend to have higher service rate and the fleet directs both idles vehicles there.

7.2.5 Line Planning Problems

Line planning problems are another class of network design problems that can be formulated without any congestion considerations in user routing (although variants also exist that consider transit passenger queue delay or crowding). These problems pertain to setting transit routes and their service frequencies on an existing network in which stations/stops are known in advance. An illustration of two feasible solutions to this problem for a sample instance is shown in Fig. 7.18, where q_{st} is demand from origin s to

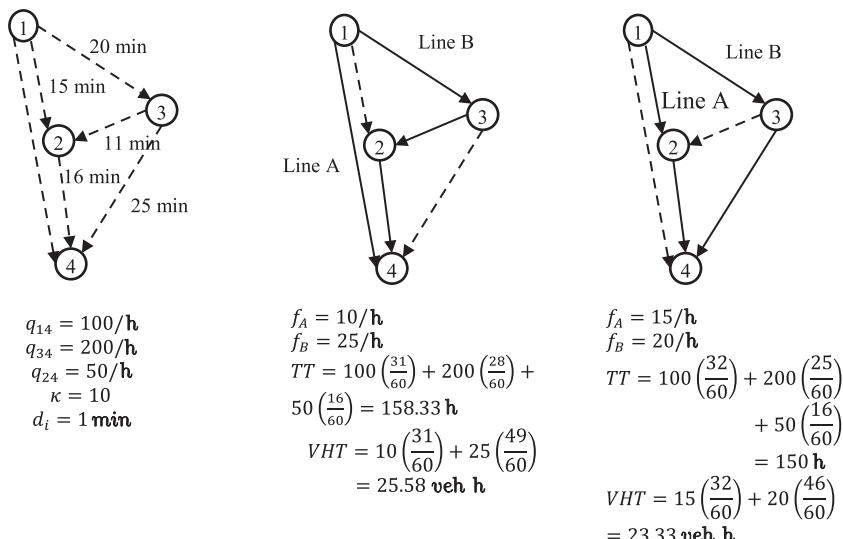


Fig. 7.18 Illustration of line planning problem.

destination t , κ is vehicle capacity, d_i is dwell time at node i , f is frequency (vehicles/h), TT is total travel time, and VHT is vehicle hours traveled.

In the example in Fig. 7.18, two alternative solutions have different routes and frequencies, resulting in different passenger travel times and vehicle hours traveled. In these solutions, passengers from node 1 may be split between lines A and B according to hyperpath assignment (see Chapter 3). However, the travel times between line A and line B are so far apart for both scenarios that there is never an advantage to take line B over line A from node 1. In some of the line planning literature (e.g., Borndörfer et al., 2007), the hyperpaths are referred to as “system splits.” A review of line planning problems is provided by Desaulniers and Hickman (2007) as “transit network design” and by Schöbel (2012). The model is shown to be NP-hard (Schöbel and Scholl, 2006).

Early studies in line planning problems, for example, Lampkin and Saalmans (1967) and Silman et al. (1974), tried to set shortest routes between endpoints. One general procedure that emerged to overcome the complexity was to split the problem into a two-step process to first identify a set of transit lines through a network and then to select frequencies. The initial skeleton route may be set using MST, TSP, or covering salesman problems like discussed in prior sections. Given a network, frequency may be set as shown in Eq. (7.16).

$$\min \sum_i \sum_j D_{ij} T_{ij}[f] \quad (7.16a)$$

Subject to

$$\sum_r R T_r f_r \leq F \quad (7.16b)$$

where D_{ij} is a fixed OD demand from i to j , T_{ij} is the travel time, $R T_r$ is the round-trip time of a route r , F is a maximum fleet size, and f_r is a frequency decision variable.

Hasselström (1981) proposed a 2-stage problem where the routes and frequencies are solved in the first stage and the passenger assignment is determined in the second stage. Demand is not fixed but is instead set as a function of travel cost as shown in Eq. (7.17).

$$\max \sum_i \sum_j D_{ij} \quad (7.17a)$$

Subject to

$$D_{ij} = K_{ij} e^{-\beta C_{ij}[f]}, \forall i, j \quad (7.17b)$$

$$\sum_r c_r f_r \leq \bar{C} \quad (7.17c)$$

$$\sum_r f_r \delta_{rs} \geq \Delta_s, \forall s \quad (7.17d)$$

$$f_r \in \mathcal{F} \quad (7.17e)$$

where K_{ij} is a constant demand parameter; β is a cost parameter for the demand function; C_{ij} is a generalized cost function that includes wait time, transfer time, and in-vehicle travel time; \mathcal{F} is a set of candidate frequencies; \bar{C} is an operating cost budget; c_r is the unit vehicle cost of operating route r ; and Δ_s is a minimum service frequency requirement for a zone s . The costs C_{ij} are dependent on passenger assignment for a given set of routes and frequencies.

Borndörfer et al. (2007) developed an exact solution method for the line planning problem with fixed demand based on column generation. The model formulation from that study is shown in Eq. (7.18). The model is defined on a multimodal graph $G(V, E)$, where $E = E_1 \cup \dots \cup E_M$ for M modes, and each edge $e \in E_i$ can be replaced with two antiparallel links a and \bar{a} . Demand exists for each OD pair $(s, t) \in W$.

y_k is the flow of passengers on a path $k \in K_{st}$, where K_{st} is the set of paths from origin s to destination t

f_l is the frequency of line $l \in L$

q_{st} is the demand from node s to node t

$x_l \in \{0, 1\}$ is a decision to use line $l \in L$

τ_k is the passenger traveling time on path k

c_i^f is the line fixed cost for mode i , and $c_l^f := c_i^f$ is the fixed cost of a line

c_e^o is the line operating cost on an edge $e \in E_i$, and $c_l^o := \sum_{e \in l} c_e^o$ is the operating cost of the line

Λ_e is the frequency bound for edge e

F_l is the frequency bound for a line

δ_{ak} is an indicator for whether a path k uses link a

γ_{al} is an indicator for whether a line l uses link a

$$\min_{x, y, f} \phi = \sum_{k \in K} \tau_k y_k + \sum_{l \in L} (c_l^f x_l + c_l^o f_l) \quad (7.18a)$$

Subject to

$$\sum_{k \in K_{st}} \gamma_k = q_{st}, \quad \forall (s, t) \in W \quad (7.18b)$$

$$\sum_{k \in K} \delta_{ak} \gamma_k - \sum_{l \in L} \gamma_{al} \kappa_l f_l \leq 0, \quad \forall a \in A \quad (7.18c)$$

$$\sum_l \gamma_{al} f_l + \sum_l \gamma_{\bar{a}l} f_l \leq \Lambda_e, \quad \forall e \in E \quad (7.18d)$$

$$f_l \leq F_l x_l, \quad \forall l \in L \quad (7.18e)$$

$$x_l \in \{0, 1\}, \quad \forall l \in L \quad (7.18f)$$

$$f_l \geq 0, \quad \forall l \in L \quad (7.18g)$$

$$\gamma_k \geq 0, \quad \forall k \in K \quad (7.18h)$$

The objective function minimizes passenger travel costs in the first term and operator costs in the second term. Eq. (7.18b) ensures passenger flows meet the demand. Eq. (7.18c) ensures there are enough lines to meet the passenger flows. Eqs. (7.18d) and (7.18e) ensure that the frequencies do not exceed an upper threshold at the edge and line levels. Eq. (7.18e) also acts to link the line selection with the frequencies. A line planning problem is illustrated in [Exercise 7.10](#).

Exercise 7.10

For the example in [Fig. 7.18](#), find the optimal solution using the formulation in Eq. (7.18).

Since no upper thresholds are specified, Eq. (7.18d) is not used in this problem. By arbitrarily setting a large value for the upper line frequencies $F_l = 1000$, the solution to this problem can be obtained.

Lines opened, frequencies:

$$(1, 4), 10/h$$

$$(1, 2, 4), 5/h$$

$$(1, 3, 4), 20/h$$

Passenger path flows: $\gamma_{1, 4}^* = 100$, $\gamma_{3, 4}^* = 200$, $\gamma_{2, 4}^* = 50$

Objective value: $\phi^* = 171.1167$ h

This result suggests it is optimal to run an express line from node 1 to node 4 while also running a local line (at a lower frequency) from node 1 through 2 to 4. If only the local line were operated, the frequency would for (1, 2, 4) would need to be 15, resulting in a higher objective value of 172.7833h.

To handle line planning problems, routes on which vehicles can be arranged should be constructed. Although this procedure is relatively simple in small networks and can be done manually as shown in Fig. 7.18, it is not as easy for more realistic networks. Ceder and Wilson (1986) proposed an enumeration method to generate such lines. The route generation method from Ceder and Wilson (1986) is provided here. The origin-based algorithm iteratively expands routes and inspects their feasibility based on the difference in travel time between the route and the shortest path from the origin. If this difference falls within a tolerable threshold, the route is elongated by a link and evaluated again, until the threshold is reached. The heuristic is shown in Algorithm 7.5.

Algorithm 7.5: (Ceder and Wilson, 1986). Route Generation Heuristic for Line Planning

Inputs: for a graph $G(N, A)$ with n nodes, demand W , travel times t_{ij} , allowable delay ratio D , direct passenger-hour matrix P , $P[i, j] = t_{ij}W[i, j]$, maximum allowable travel time M , $M[i, j] = t_{ij}D[i, j]$, set of possible route terminals T

1. Initiate: Rank terminals in descending order by $\sum_{j \in N} P[q, j]$ for each $q \in T$, route set $K_q := \{ \}$
2. For $q = 1 : |T|$,
 - a. For each node $n \in N$ from increasing order of adjacency from q ,
 - i. If n is adjacent to q ,
 1. Add route $k_{qn} := K_q \cup k_{qn}$ if route travel time $c[k_{qn}] \leq M[q, n]$
 - ii. Else
 1. For each existing route ending at a node n' adjacent to n
 - a. Extend $k_{qn'}$ to create a new route k_{qn} if travel time $c[k_{qn}]$ is within the maximum allowable travel time: $c[k_{qn}] \leq M[q, n]$
 - b. Add route $k_{qn} := K_q \cup k_{qn}$

Outputs: $\{K_q\}_{q \in T}$

Consider Exercise 7.11.

Exercise 7.11

Consider the network shown in Fig. 7.19. Use Algorithm 7.5 to construct routes for a single terminal at node 3 if $D[i,j] = 1.3$ for all (i,j) .

		Shortest travel time matrix								
<i>O\D</i>		1	2	3	4	5	6	7	8	9
1	0	10	17	10	25	26	25	31	34	
2	5	0	7	15	15	16	30	30	24	
3	20	15	0	28	21	9	42	28	17	
4	6	16	23	0	15	27	15	21	30	
5	13	11	18	7	0	12	22	15	20	
6	25	23	8	19	12	0	33	19	8	
7	20	25	29	14	14	21	0	6	15	
8	21	19	23	15	8	15	14	0	9	
9	31	29	14	25	18	6	25	11	0	

		Demand matrix								
<i>O\D</i>		1	2	3	4	5	6	7	8	9
1	0	189	341	147	111	164	321	350	431	
2	489	0	440	357	103	294	118	380	144	
3	472	297	0	204	281	272	484	196	271	
4	214	150	150	0	483	192	197	464	408	
5	316	434	407	389	0	240	229	156	151	
6	285	339	183	258	188	0	430	464	488	
7	355	158	108	101	243	253	0	441	466	
8	353	423	116	427	447	474	463	0	240	
9	401	414	439	334	253	397	211	118	0	

Fig. 7.19 Input data for Exercise 7.11.

Running the algorithm produces Table 7.6, where the “Directness measure” for each (q, n) is the excess passenger hours beyond $P[q, n]$. It produces 14 feasible routes that fit within the 1.3 ratio.

Table 7.6 Solution to Exercise 7.11

Terminal	1st node	2nd node	3rd node	4th node	Directness measure
3	2				0
3	6				0
3	2	1			0
3	6	5			0
3	6	9			0
3	2	1	4		408
3	6	5	4		0
3	6	5	8		5280
3	6	9	8		0
3	2	1	4	7	1860
3	2	5	4	7	12,530
3	6	5	4	7	914
3	6	5	8	7	14,195
3	6	9	8	7	0

7.3 BILEVEL NETWORK DESIGN

Among NDPs there is a notorious subclass of problems that is even more complex: bilevel network design problems. These problems deal with two explicit sets of decision-makers: an operator making changes to a network, and a set of travelers who are acting selfishly in response to those changes. This is called a Stackelberg game in which there is a leader and a set of followers.

A Stackelberg game differs from a Cournot-Nash game in which all the players are acting simultaneously to select a quantity without any knowledge of the other player's decision. To appreciate the impact of this asymmetric information flow (allowing the followers to know in advance the decisions of the leader, and for the leader to anticipate that), consider the [Exercise 7.12](#) from [Gibbons \(1992\)](#).

Exercise 7.12

([Gibbons, 1992](#)). Consider a duopoly of two firms i and j producing the same product and seeking to maximize profits. The price is a function of the total quantity produced by both firms: $p[q_1, q_2] = a - q_1 - q_2$ if $0 < q_1 + q_2 < a$, and 0 otherwise. Unit costs are the same: $c_i[q_i] = cq_i$, where $c < a$. Compare the Cournot-Nash Equilibrium to the Stackelberg equilibrium where firm 1 is a leader and firm 2 is a follower.

In the Cournot-Nash equilibrium, each firm is optimizing their profit subject to the optimal decision of the other player. This is shown as:

$$\max_{q_i} q_i \left(a - \left(q_i + q_j^* \right) - c \right)$$

where q_j^* is constant in that expression. The optimum can be found using first-order condition:

$$a - 2q_i^* - q_j^* - c = 0$$

$$q_i^* = \frac{1}{2} \left(a - q_j^* - c \right)$$

Since the two firms are identical, there is a system of two equations with two unknowns. We can substitute in the q_2^* into the equation for q_1^* and solve for it.

$$q_1^* = \frac{1}{2} \left(a - \frac{1}{2} (a - q_1^* - c) - c \right)$$

$$q_1^* = q_2^* = \frac{a - c}{3}$$

In the case of a Stackelberg equilibrium, now firm 2 observes q_1^* prior to choosing q_2^* . This can be determined by backward induction. Firm 2 seeks to maximize the following:

$$\max_{q_2} q_2(a - (q_1 + q_2) - c)$$

This leads to an optimal response function depending on q_1 :

$$R_2[q_1] = \frac{a - q_1 - c}{2}$$

In this case, there is no equilibrium of repeated best responses; it stops at the best response of the leader:

$$\max_{q_1} q_1 \left(a - \left(q_1 + \frac{a - q_1 - c}{2} \right) - c \right) = \frac{q_1(a - q_1 - c)}{2}$$

The optimum is:

$$q_1^* = \frac{a - c}{2}, q_2^* = R_2[q_1^*] = \frac{a - c}{4}$$

By requiring firm 2 to act as a follower to firm 1, firm 2 becomes worse off than in the Cournot-Nash equilibrium.

Many public resource allocation problems can be regarded as Stackelberg games, particularly in urban transport with congestion effects. Examples of transport network design problems include road capacity expansion, toll pricing, allocation of managed lanes, electric charging infrastructure, and more. Such problems have been studied early on (Steenbrink, 1974; LeBlanc, 1975). Several major subclasses of these problems have since evolved, some of which are comprehensively discussed in Yang and Bell (1998). Before reviewing these subclasses, a more in-depth discussion of the complexity of Stackelberg games in a network setting, and general trends for solution strategies, is warranted.

A Stackelberg game can be represented by a bilevel programming problem (Marcotte, 1986). Consider the following generic formulation shown in Eq. (7.19), based on Bard (2013).

$$\min_x F[x, y] \quad (7.19a)$$

Subject to

$$G[x, \gamma] \leq 0 \quad (7.19b)$$

$$\min_{\gamma} f[x, \gamma] \quad (7.19c)$$

Subject to

$$g[x, \gamma] \leq 0 \quad (7.19d)$$

$$x, \gamma \geq 0 \quad (7.19e)$$

This model class is shown to be NP-hard (Bard, 1991) and nonconvex (Bard and Moore, 1990). Worse yet, it is even possible for a problem to have no solution at all, as illustrated in the following problem in Eq. (7.20) from Bard (2013).

Upper level:

$$\min_x F = (2x_1 + 4x_2)\gamma_1 + (3x_1 + x_2)\gamma_2$$

Subject to

$$\begin{aligned} x_1 + x_2 &= 1 \\ x_1 \geq 0, x_2 &\geq 0 \end{aligned} \quad (7.20a)$$

Lower level:

$$\min_{\gamma} f = (-x_1 - 3x_2)\gamma_1 + (-4x_1 + 2x_2)\gamma_2$$

Subject to

$$\begin{aligned} \gamma_1 + \gamma_2 &= 1 \\ \gamma_1 \geq 0, \gamma_2 &\geq 0 \end{aligned} \quad (7.20b)$$

The lower level problem can be written with $y[x]$, and then substituted into the upper level problem. The result is the following function shown in Fig. 7.20, of which there is no minimum point. From the right-hand side, the solution suggests a minimum occurs at $x_1 = 0.25$. However, when it is exactly at that point, the value of F depends solely on the follower's choice, who is indifferent to any point of $0 \leq \gamma_1 \leq 1$.

For bilevel NDPs, a typical expression for the upper level objective to minimize is shown in Eq. (7.21), while the lower level problem is a user equilibrium as shown in Eq. (3.3) in Chapter 3. The control variables in Eq. (7.19) may impact the definition of the links ($A[u]$) or the link cost functions ($c_a[x_a; u]$). The link flows x_a are based on the lower level problem. Some models use a budget constraint, for example, $\sum_a d_a u_a \leq B$, while others may

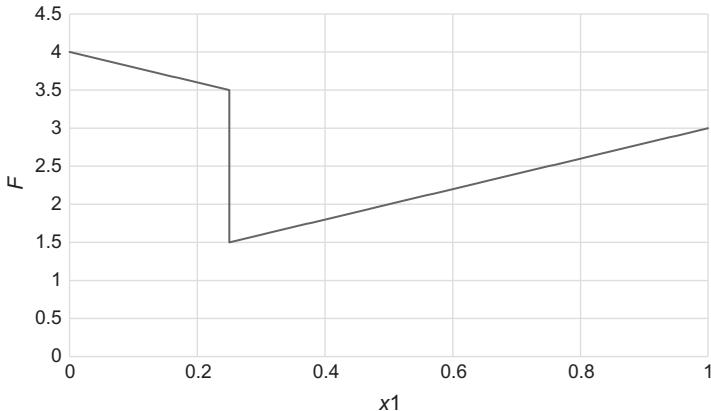


Fig. 7.20 Illustration of a bilevel problem in Eq. (7.20) with no solution.

incorporate the budget into the objective, for example, $g_a = \lambda \sum_a d_a u_a$, where λ is a multiplier to convert the budget cost to the objective value.

$$F = \sum_{a \in A[u]} (c_a[x_a; u]x_a + g_a) \quad (7.21)$$

Researchers have searched for ways to address bilevel problems. One common technique is applicable to bilevel linear problems where there are no integer decisions in the lower level problem. In such cases, one strategy is to identify the optimality conditions for the lower level problem and add those conditions as constraints to the upper level problem, effectively collapsing the problem into a single level problem (Candler and Townsley, 1982; Hansen et al., 1992). In the case that the lower level problem is nonlinear but still continuous, KKT optimality conditions can be used to reformulate the model as a single level problem (Bard and Moore, 1990). For nonlinear lower level problems whose solutions can be captured with such complementary slackness conditions (e.g., traffic assignment problem), a penalty or gap function approach can be used. The complementary slackness of the lower level problem is added to the upper level problem objective with a penalty (Marcotte and Zhu, 1996). In the case where the lower level problem involves integer programming, one approach is to use the convex hull of the lower level problem solution space as a continuous approximation (Gümüş and Floudas, 2005). Another approach is to decompose the lower level problem into subproblems (Saharidis and Ierapetritou, 2009).

In addition to these techniques, many metaheuristics have been used for bilevel network design problems: simulated annealing (Friesz et al., 1992), genetic algorithm (Chen and Yang, 2004), ant colony optimization (Poorzahedy and

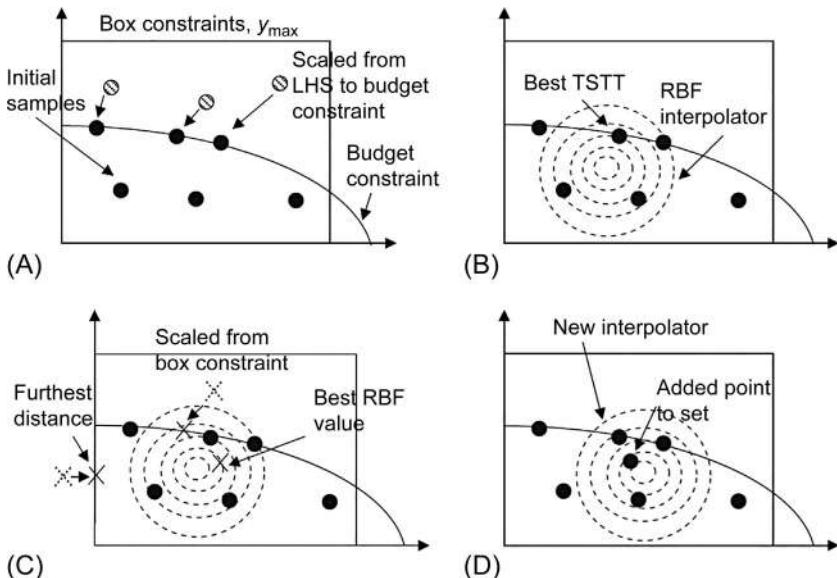


Fig. 7.21 Four steps to surrogate-based NDP: (A) generate sample solutions, (B) fit a surrogate model to the samples, (C) ensure feasibility of sample solutions, (D) select new solution using surrogate model. (Source: [Chow et al., 2010a](#).)

[Abulghasemi, 2005](#)), surrogate-based optimization methods ([Chow et al., 2010a](#)), among others. Surrogate-based optimization is a computationally efficient way to use surrogate models (e.g., radial basis functions) to approximate the objective function of the original NDP so that descent directions can be cheaply evaluated. The procedure is graphically illustrated in Fig. 7.21.

The following sections only serve to introduce the problems and their nuances; for a comprehensive review, readers are referred to [Farahani et al. \(2013\)](#).

7.3.1 Discrete Network Design Problems

In the discrete network design problem, the upper level is a discrete optimization problem of setting binary variables $u_a \in \{0, 1\}$ for a link $a \in A$ such that link flow is constrained as $x_a \leq M u_a$. This type of problem was first solved by [LeBlanc \(1975\)](#) using a branch and bound method. Due to the presence of Braess' Paradox, lower bounds have to rely on a system optimal solution, which is not a tight bound. Furthermore, as the problem is nonconvex due to the bilevel structure, the solution is only a local optimum ([Gao et al., 2005](#)). Global solution methods rely on approximation techniques, such as one using system optimal behavior to generate increasingly tighter

Algorithm 7.6: (LeBlanc, 1975). A Branch and Bound Algorithm for Discrete Network Design

Inputs: directed graph $G(N, A)$, link cost functions $c_a[x_a]$, $a \in A$, set of potential links S , budget B , link investment costs d_a , $a \in S$

0. Initiate with $i=1$, $I_i^0 = \{ \}$, $I_i^1 = \{ \}$, $I_i = S$, and $\bar{z} = \infty$, fathomed branches $R = \{ \}$
1. Solve the lower bound (SO with $u_a = 0$, $a \in I_0$, $u_a = 1$, $a \in I_1 \cup R$), set at z_i , where i is the node number in the branch and bound tree (with $i=1$ as the root, and its children are $2i$ and $2i+1$)
2. Bound: If $\bar{z} \leq z_i$, fathom this node's branches
3. Solve UE at node i to obtain link flows x_i , and set objective value $z = c_i[x_i]x_i$. If $z < \bar{z}$, update $\bar{z} = z$.
4. Branch: If all nodes are fathomed, stop; Else, for an unclosed node i , pick a next binary variable $a \in I_i$, and set $I_{2i}^0 := I_i^0 \cup \{a\}$, $I_{2i}^1 := I_i^1$, $I_{2i+1}^0 := I_i^0$, $I_{2i+1}^1 := I_i^1 \cup \{a\}$, and $I_{2i} := I_i \setminus \{a\}$, $I_{2i+1} := I_i \setminus \{a\}$. If budget is exceeded for branch j , fathom that branch: $R := R \cup \{j\}$. Proceed down an open branch, set that new node as i , and go to step 1.

Outputs: link investments u_a^* , $a \in S$, resulting user equilibrium link flows x_a^* .

lower bounds (Wang et al., 2013). LeBlanc's branch and bound algorithm is introduced in [Algorithm 7.6](#) and illustrated in [Exercise 7.13](#).

Exercise 7.13

Illustrate two layers of branches of [Algorithm 7.6](#) on the network shown in [Fig. 7.22](#).

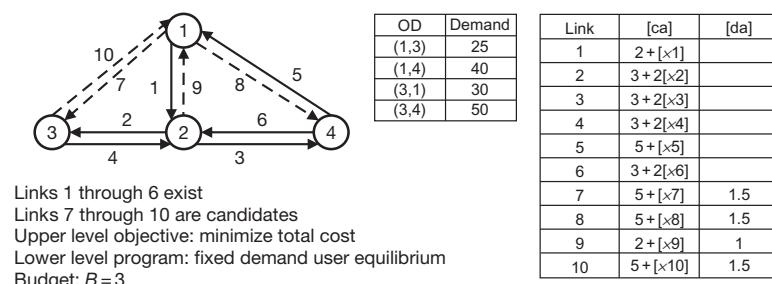


Fig. 7.22 Network used for [Exercise 7.12](#).

Initiate with $i=1$, $I_i^0 = \{ \}$, $I_i^1 = \{ \}$, $I_i = S$, $\bar{z} = \infty$, $R = \{ \}$.

At the root node, we just solve the lower bound SO (with all links invested) and UE (with no links invested) first. We obtain the SO solution with all links open as $z_1 = 323.33$.

When all potential links are closed, the UE solution is $\bar{z} = 1120$.

Since there is a gap, we branch out to nodes 2 and 3 using link 7: $u_7 = 0$ for $i=2$ and $u_7 = 1$ for $i=3$.

At $i=2$, the lower bound with $u_7 = 0$ is $z_2 = 389.48$. UE is the same as $i=1$. Branches for 4 and 5 are created for link 8.

At $i=3$, the lower bound with $u_7 = 1$ is the same as for $i=1$: $z_3 = 323.33$, while the objective under UE flows updates the upper bound to $\bar{z} = 990$.

At $i=4$, the lower bound with $u_7 = u_8 = 0$ is $z_4 = 599.73$.

At $i=5$, the lower bound is $z_5 = 389.48$, while the upper bound is $\bar{z} = 720$, which we set as the new bound.

At $i=6$, the lower bound is $z_6 = 469.73$.

Finally, at $i=7$, the lower bound remains at $z_7 = 323.33$ while the upper bound is newly set at $\bar{z} = 663.75$.

At this point, node 7 is already at the budget constraint and no further branches can proceed from here. So far no lower bound has been fathomed, so nodes 4, 5, and 6 can all be branched. The current solution after seven nodes has an objective value $\phi = 663.75$ with a decision to invest in links 7 and 8. We also know this solution can be improved at most by a 41.3% reduction in cost by branching from node 5. This is one of the criticisms of this method, that the gap from the defined lower bound is generally not a tight bound. A summary of the branching is shown in Fig. 7.23.

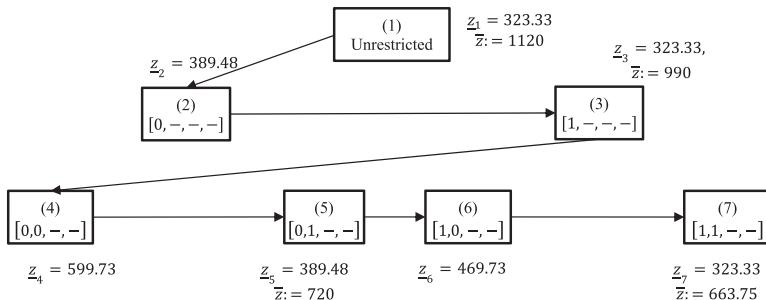


Fig. 7.23 Two layers of Algorithm 7.6 on Exercise 7.13.

7.3.2 Continuous Network Design Problems

Continuous NDPs (CNDPs) involve continuous investment decision variables, which may be nonlinear objective functions. CNDPs may also involve discrete variables, in which case they are called mixed NDPs. Two examples of continuous decision variables are shown in Eq. (7.22). In this equation, a link performance function is defined based on the BPR function (see Chapter 3). It is augmented to allow for changes to the function through a decrease in congestion rate by increasing highway capacity (u_a) or by applying a toll (v_a). Other continuous variables include signal control, speed control, inflow control, reserve capacity allocation, and more.

$$c_a = c_{a0} \left(1 + 0.15 \left(\frac{x_a}{K_a + u_a} \right)^4 \right) + v_a \quad (7.22)$$

[Steenbrink \(1974\)](#) proposed one of the earliest algorithms to solve the CNDP by iteratively fixing the value of the upper level and lower level decision variables and solving for the other until a fixed set of variables is reached. This method, called the Iterative Optimization Algorithm, converges to a Cournot-Nash equilibrium, however [\(Yang and Bell, 1998\)](#). [Abdulaal and LeBlanc \(1979\)](#) formally defined the CNDP and proposed a Hooke-Jeeves heuristic to obtain a solution. [Suwansirikul et al. \(1987\)](#) proposed an “equilibrium decomposed optimization” heuristic which is found to be more effective than the Hooke-Jeeves method. [Yang et al. \(1994\)](#) proposed a sensitivity analysis-based heuristic to obtain a local optimal solution.

More recently, global optimization approaches have been proposed to solve CNDP. [Gao et al. \(2007\)](#) converted the CNDP into a single level problem by expressing the upper level objective as a gap function based on the variational inequality converted from the lower level problem. A Method of Successive Averages was used to reach the global optimum. [Li et al. \(2012a\)](#) also applied a gap function to convert the CNDP into a single level problem but apply cutting plane algorithms to solve the reformulated problem. The MSA-based gap function method from [Gao et al. \(2007\)](#) is shown in [Algorithm 7.7](#).

Algorithm 7.7: (Gao et al., 2007). A Gap Function Algorithm for Continuous Network Design

Inputs: directed graph $G(N, A)$, link cost functions $c_a[x_a]$, $a \in A$, set of potential links S , budget parameters G_a

1. Initiate with u^0 and u^1 , and let $n=1$. Determine $x^*[u^0]$.
2. Determine upper level objective gradient. Fix the upper level variables $u=u^n$, solve the following problem to obtain $x^*[u^n]$ by implementing the user equilibrium assignment procedure in Eq. (7.23)

$$w[u^n] = \min_x \sum_{a \in A} \int_0^{x_a[u]} c_a[v, u_a] dv \quad (7.23a)$$

Subject to

$$\sum_k h_k^{rs} = q_{rs}, \quad \forall (r, s) \in W \quad (7.23b)$$

$$h_k^{rs} \geq 0, \quad \forall (r, s) \in W, k \in K_{rs} \quad (7.23c)$$

$$x_a = \sum_{(r, s) \in W} \sum_{k \in K_{rs}} \delta_{ak}^{rs} h_k^{rs}, \quad \forall a \in A \quad (7.23d)$$

where Eqs. (7.23b)–(7.23d) are the user equilibrium constraints corresponding to Eq. (3.3). Compute $\nabla_u w[u^n]$ based on Eq. (7.24).

$$\frac{\partial w}{\partial u_a} = \sum_{a \in A} \int_0^{x_a^*[u]} \frac{\partial}{\partial u_a} c_a[v, u_a] dv \quad (7.24)$$

Compute $\nabla_u F[u^n, x^*[u^n]]$ based on Eq. (7.25). The second term on the right-hand side is identical for each u_a . $\frac{\Delta x_a[u]}{\Delta u_a}$ is obtained from the current and prior iterations.

$$\frac{\partial F}{\partial u_a} = \frac{\partial w}{\partial u_a} + \sum_{a \in A} \left(x_a^*[u] \frac{dc_a[x_a, u_a]}{dx_a} \Big|_{x_a=x_a^*[u]} \frac{\Delta x_a[u]}{\Delta u_a} \right) + \int_0^{x_a^*[u]} v \frac{d^2 c_a[v, u_a]}{dv du_a} dv \quad (7.25)$$

3. Solve the LP in Eq. (7.26) to obtain an auxiliary solution \tilde{u}^n .

$$\min_{\tilde{u}^n} \nabla_u F[u^n, x^*[u^n]]^T (\tilde{u}^n - u^n)$$

Subject to

$$\begin{aligned} \sum_{a \in S} G_a[u_a] &\leq B \\ u_a &\geq 0, \quad \forall a \in S \end{aligned} \quad (7.26)$$

4. Convergence check and update. If $\nabla_u F[u^n, x^*[u^n]]^T [\tilde{u}^n - u^n] < \epsilon$, stop.

Otherwise, let $u^{n+1} = u^n + \frac{1}{n+1} (\tilde{u}^n - u^n)$, $n=n+1$, and go to step 2.

Outputs: continuous link decision variables u_a^* and link flows x_a^* .

Let us illustrate [Algorithm 7.7](#) in [Exercise 7.14](#) with the network from the prior example.

[Algorithm 7.7](#) seems effective with capacity expansion. Initial attempts to apply this method to toll pricing were not successful.

Exercise 7.14

Consider a simple example with one OD with demand $q=10$ and two parallel links with average costs $c_1 = 2 + \frac{2x_1}{1+u_1}$ and $c_2 = 5 + \frac{x_2}{1+u_2}$, where (u_1, u_2) are capacity expansions that reduce the rate of congestion. Without any capacity expansion, the total cost is $\phi_0=106.6667$. Apply one iteration of [Algorithm 7.6](#) from initial points of $(u_1^0, u_2^0)=(0, 0)$ and $(u_1^1, u_2^1)=(2, 3)$ with a budget of $u_1 + u_2 \leq 5$.

Iteration 1:

Initiate with $u^0=(0, 0)$, $u^1=(2, 3)$. Let $n=1$.

Solve the UE to obtain the link flows. We get $x^1=(6, 4)$ and $\phi_1=60$.

$$\frac{\partial c_1[x_1, u_1]}{\partial u_1} = -\frac{2x_1}{(1+u_1)^2}$$

$$\frac{\partial c_2[x_2, u_2]}{\partial u_2} = -\frac{x_2}{(1+u_2)^2}$$

$$\frac{\partial w}{\partial u_1} = -\frac{x_1^2}{(1+u_1)^2} = -\frac{6^2}{(1+2)^2} = -4$$

$$\frac{\partial w}{\partial u_2} = -\frac{x_2^2}{(1+u_2)^2} = -\frac{4^2}{2(1+3)^2} = -0.5$$

$$\frac{dc_1[x_1, u_1]}{dx_1} = \frac{2}{1+u_1} = \frac{2}{3}$$

$$\frac{dc_2[x_2, u_2]}{dx_2} = \frac{1}{1+u_2} = \frac{1}{4}$$

From u^0 we can compute a difference:

$$\frac{\Delta x_1[u_1]}{\Delta u_1} = \frac{6 - 4.33}{2 - 0} = 0.83$$

$$\frac{\Delta x_2[u_2]}{\Delta u_2} = \frac{4 - 5.67}{3 - 0} = -0.56$$

$$\int_0^{x_1^*[u]} v \frac{dc_1[v, u_1]}{dv du_1} dv = - \int_0^6 \left(\frac{2v}{(1+u_1)^2} \right) dv = -\frac{36}{9} = -4$$

$$\int_0^{x_2^*[u]} v \frac{dc_2[v, u_2]}{dv du_2} dv = - \int_0^4 \frac{v}{(1+u_2)^2} dv = -\frac{8}{16} = -0.5$$

This is plugged into the gradient approximation:

$$\frac{\partial F}{\partial u_1^1} = -4 + \left(6 \left(\frac{2}{3} \right) (0.83) + 4 \left(\frac{1}{4} \right) (-0.56) \right) - 4 = -5.22$$

$$\frac{\partial F}{\partial u_2^1} = -0.5 + \left(6 \left(\frac{2}{3} \right) (0.83) + 4 \left(\frac{1}{4} \right) (-0.56) \right) - 0.5 = 1.78$$

The following LP is solved:

$$\min_{\tilde{u}^1} -5.22\tilde{u}_1^1 + 1.78\tilde{u}_2^1 : \tilde{u}_1^1 + \tilde{u}_5^1 \leq 5, \tilde{u}_a^1 \geq 0$$

The optimum solution is $\tilde{u}_1^1 = 5, \tilde{u}_5^1 = 0$

MSA update: $u_1^2 = 2 + \frac{1}{2}(5-2) = 3.5, u_2^2 = 3 + \frac{1}{2}(0-3) = 1.5$, resulting in $x_1^2 = 8.29, x_2^2 = 1.71, \phi_2 = 56.84$.

Due to the computational cost of CNDPs, surrogate-based methods are especially effective because they are designed for high-dimensional problems with continuous decision variables (Regis and Shoemaker, 2007). For example, the algorithm proposed in Chow et al. (2010a) is shown to take only 35% of the time of a genetic algorithm to reach the best solution found for the Anaheim network with 31 investment links. Other surrogate-based methods have also been proposed for mixed NDPs that have both continuous and discrete variables (Chen et al., 2015; Rodriguez-Roman, 2018).

7.3.3 Activity-Based Network Design

The last class of bilevel NDPs was proposed more recently. Kang et al. (2013) argue for the importance of evaluating network designs where a heterogeneous population of users have demand to schedule their daily activities (see Chapter 4). Methodologically, activity-based NDPs (AB-NDPs) are like location routing problems in the location and routing literature (Perl and Daskin, 1985). Discrete decisions are made by a network design-maker,

followed by responses in routing to reflect sensitivities in users' activity scheduling.

AB-NDPs are important in many applications: locating limited fueling infrastructure ([Kang and Recker, 2014](#)), designing multimodal transport facilities ([Chow and Djavadian, 2015](#)), and designing vehicle-to-grid (V2G) pricing schedules ([Nourinejad et al., 2016](#)), to name a few.

The structure of an AB-NDP is like a discrete NDP. The upper level is a discrete optimization problem while the lower level is a measure of population scheduling responses. The lower level problem can be represented in several ways.

- Representative sample schedule responses: if there are no capacity or congestion effects, and only heterogeneity of travelers is needed, a set of mHAPP models (see [Chapter 4](#)) can be solved for each sample household or individual. An example is shown in [Kang et al. \(2013\)](#).
- Population-level schedule responses with homogeneous taste preferences: in the case congestion or capacity effects at certain bottlenecks are desired, a quasidynamic user equilibrium model like in [Lam and Yin \(2001\)](#) can be applied. An example is shown in [Nourinejad et al. \(2016\)](#).
- Population-level schedule responses with heterogeneous taste preferences: in the case where capacity effects are desired as well as individual variations in scheduling or travel preferences, a mixed logit model of schedule choice that is calibrated using mHAPP models can be used. This has not been applied yet, but it was discussed in [Chow and Djavadian \(2015\)](#).

We expand on the last method. A modeler is assumed to have calibrated a set S of mHAPP models beforehand. The AB-NDP then involves finding an upper level solution such that the followers in aggregate behave within the capacity effects. A branch and bound method like [LeBlanc's \(1975\)](#) can be applied here as well; the difference is that since Braess' Paradox does not exist when there is no congestion effect, the lower bound should be tighter than in the discrete NDP. The customized method is presented in [Algorithm 7.8](#), although more rigorous testing will need to be done in future research.

Algorithm 7.8: A Branch and Bound Algorithm for Activity-Based Network Design With Heterogeneous Population

Inputs: directed activity and infrastructure graph $G(N, A)$, $N = Z \cup \Lambda$ where Z are activity zones and $\Lambda = \Lambda_1 \cup \dots \cup \Lambda_M$ are M transport nodes, link cost functions $c_a[x_a]$, $a \in A$, capacities κ_a on links serving transport nodes, J schedule alternatives, set S of individuals, set of potential discrete multimodal projects P (can alter capacity, add links, etc.), budget B , link investment costs d_p , $p \in P$

0. Initiate with $i=1$, $I_i^0 = \{ \}$, $I_i^1 = \{ \}$, $I_i = S$, and $\bar{z} = -\infty$, fathomed branches $R = \{ \}$
1. Solve the upper (since this is utility maximization) bound (ϕ with $u_p = 0$, $p \in I_0$, $u_p = 1$, $p \in I_1 \cup I$, set at \bar{z}_i , where i is the node number in the branch and bound tree (with $i=1$ as the root, and its children are $2i$ and $2i+1$). Objective can be defined as maximizing consumer surplus (see Small and Rosen, 1981) in Eq. (7.27):

$$\max \phi = \sum_{n \in S} \frac{1}{\alpha_n} \ln \sum_{i \in J} \exp [V_{in}[u_p]]$$

Subject to

$$\sum_{p \in P} d_p u_p \leq B \quad (7.27)$$

where the values of $V_{in}[u_p]$ are sampled from the mHAPP models. Because parameters β_n are defined for each sample, there is no need to simulate probability.

2. Bound: If $\bar{z}_i \leq z$, fathom this node's branches
3. Solve Eq. (7.27) with only the present projects I_i^1 confirmed at node i . If $z < z$, update $z := z$.
4. Branch: If all nodes are fathomed, stop; Else, for an unclosed node i , pick a next binary variable $a \in I_i$, and set $I_{2i}^0 := I_i^0 \cup \{a\}$, $I_{2i}^1 := I_i^1$, $I_{2i+1}^0 := I_i^0$, $I_{2i+1}^1 := I_i^1 \cup \{a\}$, and $I_{2i} := I_i \setminus \{a\}$, $I_{2i+1} := I_i \setminus \{a\}$. If budget is exceeded for branch j , fathom that branch: $R := R \cup \{j\}$. Proceed down an open branch, set that new node as i , and go to step 1.

Outputs: link investments u_p^* , $p \in P$, resulting schedule flows x_j^* , $j \in J$

The algorithm is demonstrated in the [Exercise 7.15](#).

Exercise 7.15

Consider the example in [Fig. 7.24](#). There is a budget to locate two identical facilities among three candidate locations (5, 6, 7) each with a capacity of 100 visitors. Visiting these facilities yield a utility. Tours originate from two nodes, 1 (pop. 700) and 2 (pop. 300), with a mandatory visit to node 3 and optional visit to node 4 (no capacity). There are two classes of users, with 50/50 split living in node 1 and a 20/80 split in node 2. Run the initial iteration of [Algorithm 7.8](#) to seek the welfare maximizing allocation of facilities.

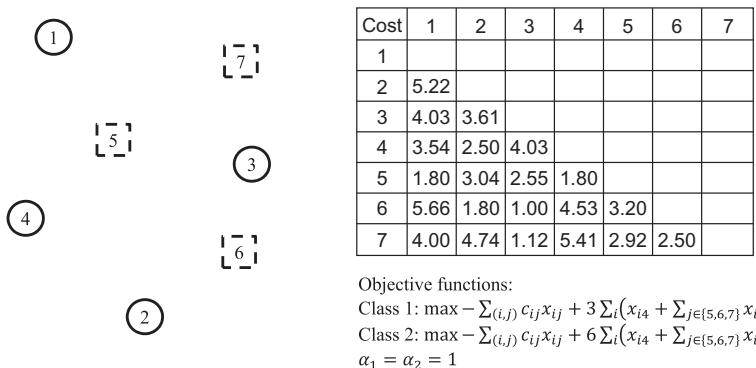


Fig. 7.24 Example for [Exercise 7.15](#).

This is a stylized version of activity-based NDP. Arrival times are ignored; the focus is only on tours. Node 3 represents a downtown work neighborhood while nodes 4–7 represent optional activities that add value to a user's tour. Although there are only two classes representing the population of 1000, in practice there should be hundreds or thousands of sample trajectories from many different home zones representing a population. The two classes differ in the value that the users place in those secondary activities. Although they are all treated the same for one class in this example, in practice different activity destinations may have different values to different people.

First we solve the base alternative where none of the facilities are available to users.

Node 1, class 1: (1, 3, 1), $V_{131,1} = -8.06$, population: 350

Node 1, class 2: (1, 3, 4, 1), $V_{1341,2} = -5.6$, population: 350

Node 2, class 1: (2, 3, 4, 2), $V_{2342,1} = -7.14$, population: 60

Node 2, class 2: (2, 3, 4, 2), $V_{2342,2} = -4.14$, population: 240

Lower bound: $\sum_n P_n \ln \sum_i \exp(V_{in}) = -6203$

Now we solve the upper bound where all three facilities are available and the alternative routes are added to the base scenario. The capacity-constrained mixed logit solution is shown as follows.

Node 1, class 1:

$(1, 3, 1), V_{131,1} = -8.06$, population: 350

Node 1, class 2:

$(1, 3, 4, 1), V_{1341,2} = -5.6$, population: 350

Node 2, class 1:

$(2, 3, 4, 2), V_{2342,1} = -7.14$, population: 60

Node 2, class 2:

$(2, 3, 4, 2), V_{2342,2} = -4.14$, population: 140

$(2, 4, 5, 7, 3, 6, 2), V_{2457362,2} - w_5 - w_6 - w_7 = 12.86 - 9.44 - 4.08 - 3.48 = -4.14$, population: 100 (binding)

Upper bound: $\sum_n P_n \ln \sum_i \exp(V_{in}) = -4503$

The upper bound consumer surplus assumes that a subset of the 140 population has access to both $(2, 3, 4, 2)$ and $(2, 4, 5, 7, 3, 6, 2)$. This subset is based on the outcome that 100 of them would choose $(2, 4, 5, 7, 3, 6, 2)$. Because the market share is practically 100%, the subset itself is also 100.

The lower bound and upper bound solutions after the initial iteration are shown in Fig. 7.25. Due to the presence of capacities in the three new nodes, only the class 2 population (with the higher utility per visit) can use the three nodes. Furthermore, only the class 2 population residing in node 2 has the convenience to chain node 4 and all the other nodes together. In solving the lower level problem in the upper bound, we also produce dual prices for each of the added facility capacities: $w_5 = 9.44$, $w_6 = 4.08$, $w_7 = 3.48$. This reflects the popularity of node 5 in more of the routes due to its centrality in location.

One layer of branches would have one branch with $\gamma_5 = 0$ and another with $\gamma_5 = 1$, where γ_j is the decision to invest in node j .

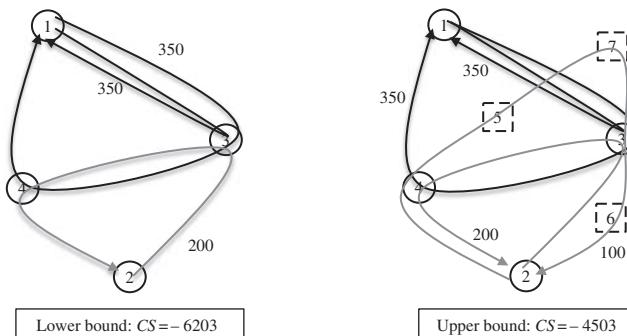


Fig. 7.25 Lower (0,0,0) and upper bound (1,1,1) solutions at the root node.

7.4 NETWORK DESIGN UNDER COEXISTING SYSTEMS

The prior network design models assume network improvements are made by a single decision-maker without considering other operators in the same region. This is an important problem in a MaaS system since its definition implies interactions between multiple operators. Other operators that either act in direct competition or provide complementary mobility service impact the decisions of the operator. For example, expansion of a bike-sharing system can depend on the design of a nearby public transit service even if they are not directly competing for service. On the other hand, two e-hail services operating in the same region would have to consider the strategies of their adversary. This section covers different aspects of this setting. Consider the following definitions of coexistence and interaction.

Definition 7.1 A set G of g networks $G_n[N_n, A_n]$, $1 \leq n \leq g$, serving commodities W_n , **coexist** if there is (a) a nonempty set of commodities $\Gamma \subseteq W_1 \cap \dots \cap W_g$ for each network, (b) a nonempty set of nodes $\Lambda \subseteq N_1 \cap \dots \cap N_g$, or (c) a nonempty set of links $\Upsilon \subseteq A_1 \cap \dots \cap A_g$.

Definition 7.2 Coexisting networks **interact** if the performance of a network, ϕ_n , $1 \leq n \leq g$, depends on the strategies of other networks $\phi_{-n}[y_{-n}]$, where y_{-n} denotes decision variables of networks other than n .

Definition 7.1 implies three classes of mechanisms for interaction between coexisting networks, either based on shared demand, shared nodes, or shared links. For example, multimodal travelers using multiple transport services are examples of the first class. Airports where freight and passenger flows share the same facility are examples of the second class. Link-based interactions include railways that run freight and passenger services on the same track, or roadway used by both passenger vehicles and buses. **Definition 7.2** provides a broad definition of interaction that does not depend on the availability of information to each operator or timing of decisions made.

Research on interactions between transport network operators emerged in the 1980s as a period of deregulation in some industries. For example, [Harker \(1988\)](#) devised a generalized Nash equilibrium model to evaluate the feasibility of private mass transit markets in the United States. Other types of interactions have also been considered. [Yang and Woo \(2000\)](#) and [Zhang et al. \(2011b\)](#) proposed Nash and Stackelberg game models to evaluate competition between two toll pricing firms. [Adler et al. \(2010\)](#) examined the interactions between airlines and high-speed rail in intercity travel. [Logi and Ritchie \(2002\)](#) tackled coordination between multiple traffic management centers.

[Chow and Sayarshad \(2014\)](#) introduced the concept of symbiotic network design in which case two or more agents are not necessarily in direct competition but their operations do impact one another.

The first part introduces general problems dealing with multiple coexisting operators and classifies different categories of problems that arise. Network design games are then introduced as applications of game theory in a network setting. Lastly, when multiple operators need to work together, network design problems need to consider privacy awareness in sharing information between the operators.

7.4.1 Symbiotic Network Design

Broadly speaking, two or more network operators may coexist on the same network such that their decisions impact one another. Depending on the type of interaction, different games may ensue. For example, two operators that compete within the same time horizon for passengers may be viewed as a Nash game. On the other hand, there may be interactions where there is a sequence of multiple decisions, different time scales in decisions (e.g., one decision-maker makes one tactical decision while another makes multiple decisions in that same time frame), different degrees of information sharing, and more, that makes it difficult to have consistent approach to analyzing policies that relate to these interactions.

[Chow and Sayarshad \(2014\)](#) proposed a more general framework in which the impacts of interactions between operators can be quantified so that, even without having to evaluate the interdependencies of the decisions, one can classify those decisions and provide guidance on policies such as subsidization, integrated fares, integrated schedules and transfers, shared user information, or shared facilities, among others.

The framework is based on the use of multiobjective optimization and Pareto optimality techniques to classify operator strategies under a framework of symbiosis. Multiobjective optimization is normally applied to a single decision-maker facing multiple objectives in an optimization problem, resulting in the need for dominance criteria like Pareto optimality to compare solutions (see [Current and Marsh, 1993](#); [Mordukhovich, 2004](#); [Chow and Regan, 2014](#)). Symbiosis is a biological phenomenon that refers to the coexistence of two different living organisms that form persistent associations (see [Douglas, 2010](#)). Examples of symbiosis in nature include sea anemones and hermit crabs, African oxpeckers with herd mammals, and bees and orchids. In the context of network design, symbiosis takes on the following interpretation.

Definition 7.3 Coexisting networks are **symbiotic** if changing the design of one network from one state to another changes the performance of the other coexisting networks. Symbiotic networks are purely mutualistic if changing the design of one network to improve (degrade) its performance results in an improvement (degradation) in the other coexisting networks. Symbiotic networks are parasitic if changing the design of one network to improve (degrade) its performance results in a degradation (improvement) in the other coexisting networks.

The symbiotic extension of Eq. (7.1) is shown in Eq. (7.28).

$$\begin{aligned} & \min_{x, y} \phi_1[x, y_1; y_2, \dots, y_g] \\ & \dots \\ & \min_{x, y} \phi_g[x, y_g; y_1, \dots, y_{g-1}] \end{aligned} \quad (7.28a)$$

Subject to

$$\sum_{j \in N_n} x_{ij}^m - \sum_{j \in N_n} x_{ji}^m = \begin{cases} w_m, & i = O[m] \\ -w_m, & i = D[m], \forall m \in M \\ 0, & \text{otherwise} \end{cases} \quad (7.28b)$$

$$x_{ij} \equiv \sum_{m \in M} x_{ij}^m \leq K_{ij} y_{ijn}, \quad \forall (i, j) \in A, 1 \leq n \leq g \quad (7.28c)$$

$$\left(x, \{y_n\}_{1 \leq n \leq g} \right) \in S \quad (7.28d)$$

$$x_{ij}^m \geq 0, \quad y_{ijn} \in \{0, 1\}, \quad \forall (i, j) \in A, m \in M \quad (7.28e)$$

Interactions within this framework that are interpreted as specific types of games have specific constraints sets S . The symbiotic framework can then be used to identify the type of relationship that a strategy falls under, as illustrated in the regions of the Pareto optimal set in Fig. 7.26.

In the figure, two symbiotic operators are illustrated, designated as a Host and a Guest, both seeking to minimize objective values from an initial state that they share. A design strategy proposed by the Guest can be parasitic to the Host if it improves the Guest at the Host's expense. Solutions on the mutualistic strategies are attainable in a cooperative environment. If the two operators are noncooperative and information is not readily shared, then dominated strategies may be produced. As an example, in the Prisoner's Dilemma the Nash equilibrium constraint imposed on the choices of the two prisoners results in a Pareto optimal solution at the worst option. This is because the other three choices are simply not feasible solutions as defined.

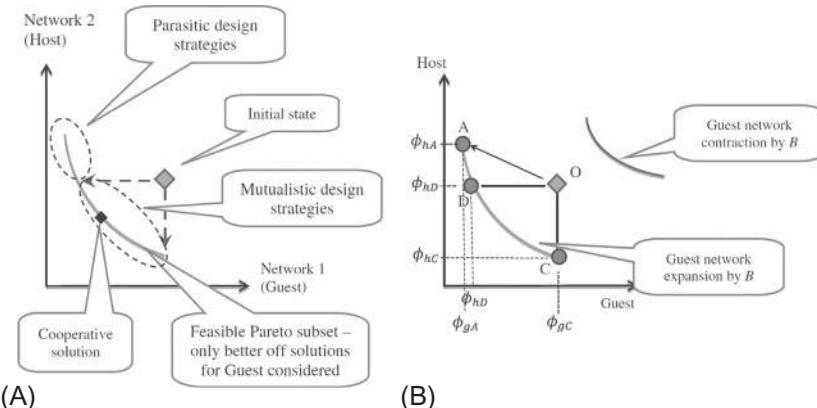


Fig. 7.26 (A) Regions for classifying symbiotic network design strategies and (B) impact of network expansion and contraction. (Source: [Chow and Sayarshad, 2014](#).)

The framework allows network operators to negotiate. If the Guest operator considers a parasitic strategy, the Host operator may counter with modifications to that strategy to keep the solution mutualistic. These counters include infusing the investment with a subsidy or in proposing a fare bundling policy where a mutualistic strategy can be attained. The framework is illustrated in Exercise 7.16 from [Chow and Sayarshad \(2014\)](#).

Exercise 7.16

Consider the example in Fig. 7.27 originally analyzed by [Zhang et al. \(2011b\)](#). There are two operators, α and β , each managing three nodes. Four pairs of OD demand are served: (2, 1), (2, 6), (5, 1), (5, 6). If both operators seek to maximize social welfare of their own residents by imposing tolls—operator α on link 2, operator β on link 7—there are four scenarios that arise. When there are no tolls, the social welfare (SW_α, SW_β) is $(7.398 \times 10^7, 1.137 \times 10^8)$. If only operator α implements an optimal toll for itself, the social welfare is $(7.687 \times 10^7, 1.127 \times 10^8)$. If both operators implement tolls competitively, the unique Nash equilibrium lies at $(7.671 \times 10^7, 1.164 \times 10^8)$. If the two operators cooperate, they get $(7.670 \times 10^7, 1.165 \times 10^8)$. Plot these strategy objective values in the multiobjective objective space and discuss using the symbiotic framework.

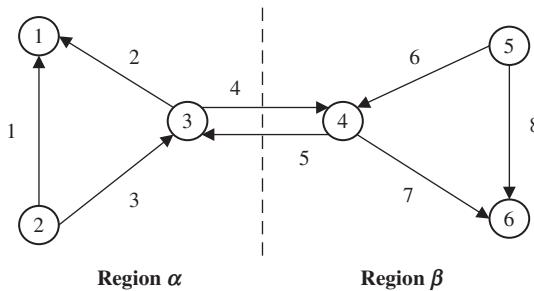


Fig. 7.27 Example network for Exercise 7.16.

The solutions are plotted in the objective space and shown in Fig. 7.28. Several conclusions can be made. When considering only the toll pricing of one operator, the strategy is parasitic to the other operator. If operator α were to consider this strategy, operator β may counter either by setting their own toll or by some other strategy. Two-region competition is mutualistic in this example and quite close to the cooperative strategy.

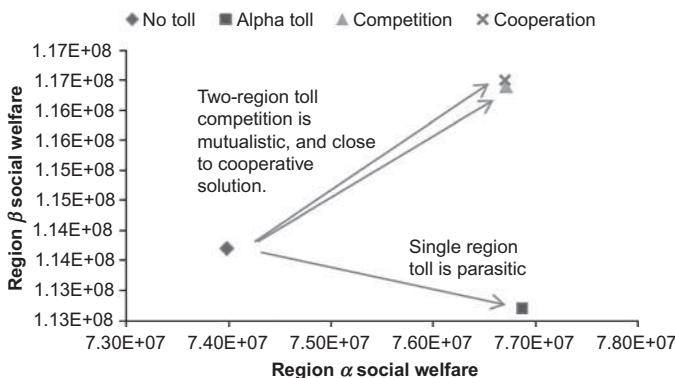


Fig. 7.28 Objective space of the alternative solutions. (Source: [Chow and Sayarshad, 2014](#).)

7.4.2 Network Design Games

Network design games refer to the diverse group of problems within network design involving the equilibrium of cooperative or noncooperative interactions between operators. Due to the fragmented nature of different NDPs, game variants have emerged specifically tuned to those subproblems. For example, competitive facility location was studied as early as in [Hotelling \(1929\)](#). These games

involve multiple decision-makers setting facilities acknowledging that competitors' decisions will impact the outcome. Ghosh and Craig (1983) considered competitive facility location under a dynamic setting where competitors relocate in response to new strategic locations. De Palma et al. (1989) added random utilities to the demand. Mallozzi (2007) considered multiple competitors with multiple facilities. In a survey by Plastria (2001), the difference between simultaneous entry games and sequential entry games is distinguished.

On the spanning tree front, many of the studies focus on cooperative games. Bird (1976) applied cooperative game theory to allocate costs to different network operators choosing to share their built links with the coalition. Cardinal et al. (2011) considered a Stackelberg game variant where one player sets prices to lease out usage of their portions of a tree while the follower player then optimizes a minimum spanning tree in response. Cost allocation on Steiner tree games has also been considered (Megiddo, 1978).

A similar history of research occurred for network flow problems, where cooperative games were proposed for networks in which one multiple operators owned different links in the network. A coalition that forms between these operators is used to allow flow from an origin to destination. Examples include Kalai and Zemel (1982), Granot and Granot (1992), Markakis and Saberi (2005), and Potters et al. (2006). The focus of these studies has been on tight regions for stability in such games, such as the nucleolus, and on expanding it to different variations of network flow problems.

In the case of bilevel network design games, Harker (1988) showed that noncooperative games between multiple transit operators cannot guarantee unique equilibria. Fernandez and Marcotte (1992) used a quasivariational inequality formulation to obtain solutions. Applications have been studied for frequency setting (Zubieta, 1998), fare setting (Zhou et al., 2005), fleet sizing (Li et al., 2008), and market entry and exit strategies (Li et al., 2012b).

A good example to illustrate the complexity of the noncooperative NDP is in competitive toll road capacity and price design for two operators m and n handling single links each, in Yang and Woo (2000). Suppose the toll price τ_i for operator i 's link is incorporated into the elastic demand variant of the Beckmann formulation as shown in Eq. (7.29), where t_i is the travel time on operator i 's toll road, γ_i is a capacity parameter (e.g., the denominator in the BPR function), x_i is the link flow, β is a conversion of tolls to travel time savings, q_w is a demand for OD $w \in W$, D_w^{-1} is an inverse of the demand function $D_w[u_w]$, and u_w is the generalized travel cost for OD pair $w \in W$.

$$\min \sum_a \int_0^{x_a} t_a[\omega] d\omega + \sum_i \int_0^{x_i} (t_i[\omega, \gamma_i] + \beta \tau_i) d\omega - \sum_w \int_0^{q_w} D_w^{-1}[\omega] d\omega \quad (7.29)$$

Operator i 's objective function and capacity construction cost function are captured by Eqs. (7.30a) and (7.30b), respectively. The term κ is a proportionality parameter.

$$\max_{\tau_i, y_i} \pi_i[\tau, y] = x_i[\tau, y]\tau_i - \alpha I_i[y_i] \quad (7.30a)$$

$$I_i = \kappa t_i^0 y_i \quad (7.30b)$$

Under Nash equilibrium for the two operators, Eq. (7.26) would impose the user equilibrium constraints in set S as well as first-order conditions for each firm obtained from their individual objective functions (Eq. (7.26a)). In this example shown in [Exercise 7.17](#), a base network may be altered to have either toll roads operated by the competitors in a “substitutable” fashion or in a “complementary” fashion.

Exercise 7.17

([Yang and Woo, 2000](#)). For each of the two network scenarios shown in [Fig. 7.29](#), identify one set of stable equilibria capacities, tolls, resulting profits, and welfare gain.

Link	t0_a [h]	C_a [vph]
a	0.4	3000
b	0.6	3000
c	1	4000
m1	0.8	—
n1	0.4	—
m2	0.3	—
n2	0.4	—

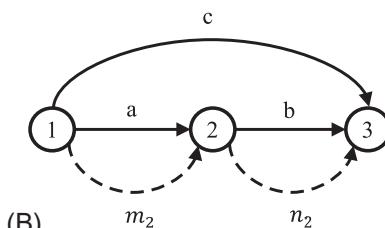
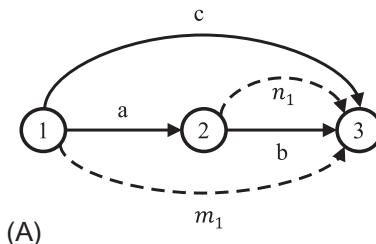


Fig. 7.29 Example with (A) substitutable toll roads and (B) complementary toll roads.

One set of equilibria is shown for each scenario in [Table 7.7](#). As this solution shows, the profits of the operators can vary significantly

depending on the design. Neither operator is at their desired maximum profit and the suboptimal welfare gain due to competition is also noted.

Table 7.7 Example equilibrium solutions

		Substitutable (A)	Complementary (B)
Capacity	Operator m	6401.00	3798.10
	Operator n	5000.10	7301.00
Toll	Operator m	16.00	9.20
	Operator n	17.90	18.70
Profit	Operator m	18,907.85	12,908.96
	Operator n	39,463.99	62,711.04
Welfare gain (%SO)		229,904.30 (79%)	296,352.00 (98%)
Social optimum (SO)		291,813.20	302,840.70

7.4.3 Privacy-Aware Network Design

In a symbiotic network setting such as smart cities, collaboration between operators without sacrificing competitive privacy of one's data is an important topic. Network design problems are only as useful as the input parameters. Those parameters, in turn, depend on distributed input data shared by other private operators in a MaaS setting. With competitive privacy awareness, data may be filtered through a third-party platform to other parties in exchange for shared usage of the platform for feeding one's own operations. Data would be treated as secure commodities that operators trade with each other in an anonymized format, using systems similar to cryptocurrency platforms. Learning and privacy control models (see [Chapters 5 and 6](#)) may be used to maximize usage of the data without sacrificing a platform contributor's privacy. An example is to use the private data shared by multiple operators to train a neural network model without sharing that data across the platform to other operators. In turn, each operator makes use of the trained model as their data inputs (e.g., [Chen and Zhong, 2009](#); [Bansal et al., 2011](#)). It has been shown to have significant application in the healthcare industry due to privacy concerns in medical records.

As far as methods for privacy-aware network design, this remains an open problem. However, given its importance, this section is used to highlight some near-term research questions that should be addressed.

- Since privacy control for sharing data from private operators depend on needs of public agencies (the Δ in [Chapter 6](#)), there needs to be an integrated network design model where input parameters are functions of distributed databases with the Δ as a decision variable.
- There also needs to be an understanding of the value of privacy as this varies by operator and user. Some initial efforts ([Savage and Waldman, 2015](#); [Hirschprung et al., 2016](#)) suggest a wide range of values that also depend heavily on context.
- For multiple operators considering sharing data, there needs to be a cooperative game model in which data is explicitly valued. In such a scenario, the benefit to having a shared database formed from a data sharing coalition may be greater than the risk of loss of data privacy.
- Cryptocurrency system designs should be investigated for constructing a data-oriented platform for MaaS. The platform should include a learning model so that the sum is greater than the parts.
- Privacy-aware network design algorithms need to be evaluated in terms of statistical reliability as it scales with availability of data, similar to how computational performance is measured.

RESEARCH AND DESIGN CHALLENGES

The following are open-ended challenges that sometimes refer to real data sets. Readers are encouraged to try to tackle the problems, and post their designs or solutions on the following GitHub page, <https://github.com/BUILTNYU/Elsevierbook/tree/master/Chapter%207>.

- (7.1) Download the set of stops from a local bus network of your choice (<https://transitfeeds.com/>). Write a program for [Algorithm 7.1](#) and apply it to the bus stops. Create a set of routes from the MST. Find ridership numbers for the stops (e.g., in NYC the daily route level bus ridership can be found at http://web.mta.info/nyct/facts/ridership/#chart_b, from which estimated stop-level ridership can be assumed). For these routes, assign frequencies using Eq. (7.16). Compare the networks and sample travel times for random stop pairs. For automated travel time queries, you can use the following code: <https://github.com/BUILTNYU/Google-Map-API-Query-Program-and-Documentation>.
- (7.2) Write a program for [Algorithm 7.2](#) to solve TSPs. Test it on a tour of the 59 national parks in the United States (<http://www.randalolson.com/2016/07/30/the-optimal-u-s-national-parks-centennial-road-trip/>).

- (7.3) Pick out a food type of your preference (Chinese, Italian, etc.). Use Yelp (<https://www.yelp.com/>) to query the top 10 locations in your vicinity and 10 dessert locations. Formulate a problem to visit one of each set of destinations to minimize travel time while maximizing accrued Yelp ratings. Try different weight values between the two objectives and interpret the resulting solutions.
- (7.4) Implement [Algorithm 7.3](#) to solve VRPs. Test the code on several of Solomon's test instances for VRPs (one host site is at SINTEF: <https://www.sintef.no/projectweb/top/vrptw/solomon-benchmark/100-customers/>).
- (7.5) Randomly generate 8 nodes on a 10 by 10 unit Euclidean space and randomize the demand from each node between 1 and 10 units. Find a p-median solution for three facilities using the integer programming formulation and compare that to [Algorithm 7.4](#).
- (7.6) For the solution of the randomly generated example in Challenge 7.5, suppose in the next period the demand for the 8 nodes are changed to a new set of randomly generated values between 1 and 10. Find an optimal relocation plan using the formulation in Eqs. (7.11) and (7.12) where θ_{rj} is set equal to the Euclidean distance.
- (7.7) Develop an online program using Eq. (7.13) to direct idle e-hail vehicles to zones to minimize the amount of time it takes to match with a customer. For a given set of zones, request data in each zone is available from prior periods. This information is used to approximate the demand in a current period. Service rate μ_j is the number of passenger pickups in each zone per period. The vehicles in each zone are treated as servers. As the size of the problem may be quite large, assume an LP relaxation of the model (see [Sayarshad and Chow, 2017](#)).
- (7.8) Based on the smart card data in <http://www-users.cs.umn.edu/~tianhe/BIGDATA/>, use the OD demand data to design frequencies using the same total fleet size. Is the solution the same?
- (7.9) For the same data set in Challenge 7.8, apply [Algorithm 7.5](#) to generate a new set of routes and obtain a solution to Eq. (7.18).
- (7.10) Program [Algorithm 7.6](#) and run it to solve the discrete network design problem for the Sioux Falls network (<https://github.com/bstabler/TransportationNetworks>) in the same way conducted by LeBlanc.

- (7.11) For the same Sioux Falls network with continuous design variables, compare the solution of the IOA approach from [Steenbrink \(1974\)](#) with [Algorithm 7.7](#). Show how one solution is a Cournot-Nash equilibrium and the other is a Stackelberg equilibrium.
- (7.12) For the synthesized region in Challenge 4.9, formulate an NDP that either improves capacity or adds new links and/or station to a system. Use [Algorithm 7.8](#) to optimize the AB-NDP.
- (7.13) Design one coexisting networks example involving parking garages setting their capacities and prices along with a bus operator. Is the relationship between the two mutualistic or parasitic? Propose one interaction strategy between the two and evaluate the strategy.
- (7.14) Consider the PATH train and the NYCT subway lines that connect directly to the PATH train in NYC. Model their timetabling decisions and integrated fare package as a network flow game that takes transfers into account. Is there a stable outcome for them to cooperate?
- (7.15) Formulate a privacy-aware NDP and construct an example to illustrate its solution.

CHAPTER 8

Network Portfolio Management

8.1 INTRODUCTION

Chapter 7 introduced network design methods, which are essentially resource allocation problems in a network setting. While resource allocation problems are notoriously difficult in a network because of the inherent dependencies between allocation decisions, they become even more complex under a setting of uncertainty. Nevertheless, with smart cities and real-time information afforded by the Internet of Things (IoT), the ability to effectively allocate resources in a network using real-time data under uncertainty is more urgently demanded than ever before. Prominent examples in urban transport include fleet management for MaaS (routing, dispatch, idle vehicle rebalancing), adaptive traffic control, bus bunching operations, and urban freight information systems. This chapter deals with such decisions, often made dynamically, under an uncertain environment.

Systematic management of multiple projects over time is called “portfolio management.” More specifically, Cooper et al. (1998) define portfolio management as follows.

Definition 8.1 (Cooper et al., 1998). *Portfolio management is a dynamic decision process, whereby a business’s list of active new products (and R&D) projects is constantly updated and revised. In this process, new projects are evaluated, selected, and prioritized; existing projects may be accelerated, killed, or deprioritized; and resources are allocated and reallocated to the active projects. The portfolio decision process is characterized by uncertain and changing information, dynamic opportunities, multiple goals and strategic considerations, interdependence among projects, and multiple decision-makers and locations.*

Portfolio management and dynamic decision processes, in general, are highly complex. Due to this complexity and reach, different disciplines have focused on aspects of this problem. For example, mechanical and electrical engineers study control theory, which focuses on continuous time and decisions with stationary random variables. In finance and project management, real options theory emphasizes timing and value of decisions with nonstationary random variables. In operations research, stochastic optimization

emphasizes large-dimensional decision variables but tends to limit the random processes to simple discrete distributions (Powell, 2011).

Many of these subfields in portfolio management also assume that information is exogenous to the dynamic decision-making. However, this is not the case in smart cities. Autonomous vehicles act as both service providers and sensors; dispatch and routing decisions therefore should affect information availability in the future. Similarly, many data-driven services (e.g., location-based services or mobility-on-demand) rely on data from users, but allocating too much of the effort to exploration could limit the effectiveness of providing good service to those users. For example, a mobility-on-demand service needs to predict a user's travel preferences, and efforts to learn those preferences by using surveys or recommender systems (Adomavicius and Tuzhilin, 2005) to offer options to users may require testing out unknown preferences at the risk of poor performance. This is due to the fundamental trade-offs that need to be made between exploration and exploitation (Powell and Ryzhov, 2012). This chapter touches upon that subject as well.

8.2 DECISION-MAKING UNDER UNCERTAINTY

Uncertainty deals with variables where partial or incomplete information is available, at least up until some point in time, and typically involves decisions that need to be made under that setting. It includes fuzzy variables that deal with the possibility of events and stochastic processes that deal with probabilities (see Allahviranloo et al., 2014). Given such an immense topic, we focus only on stochastic processes and pertaining to the decision theory. An example of a stochastic variable is shown in Fig. 8.1.

Many decisions are made prior to the realization of uncertainty. Fig. 8.1A and B shows the probability density function and cumulative distribution functions of a random variable that has a triangular distribution. More complex variants involve functions of random variables and the outcome of prior decisions.

A classic example of decision-making based on a stochastic variable is the newsvendor problem (Arrow et al., 1951). In the problem, demand D is a random variable with a cumulative distribution function $F_D[d]$, q is the decision for stock quantity, c is the unit cost of production, and p is the price sold. If the realized demand d is less than the stock, then the revenue is pd . When demand is greater than the stock, the revenue is pq . The expected profit is $E[P] = E[p\min[q, D]] - cq$.

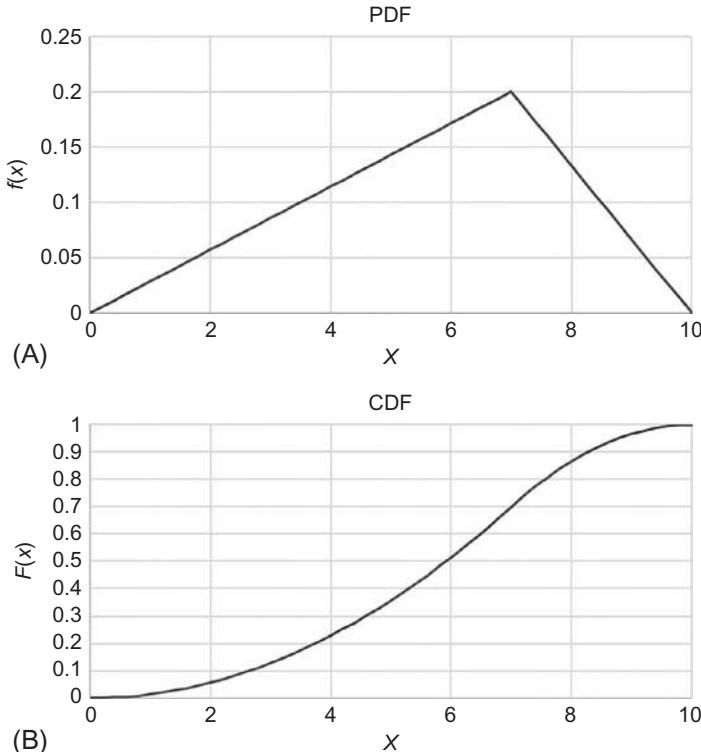


Fig. 8.1 Examples of stochastic variables: (A) a probability density function of a random variable with a triangular distribution and (B) corresponding cumulative distribution function of triangular distribution.

In the marketing literature the optimal stock quantity is known to be set based on the “critical fractile” as shown in Eq. (8.1). Exercise 8.1 illustrates this concept.

Exercise 8.1

Consider the example CDF in Fig. 8.1B. If that represents demand F_D , and $c=\$10$ and $p=\$15$, determine the optimal stock and expected profit.

The optimal stock quantity should be:

$$q^* = F_D^{-1} \left[\frac{1}{3} \right] = \sqrt{\left(\frac{1}{3} \right) (70)} = 4.83$$

The expected profit when $f_D[x]$ is the triangular distribution from 0 to 10 with peak at 7 and $q^* < 7$ is:

$$E[P^*] = \int_0^{q^*} (px - cq^*) f_D[x] dx + \int_{q^*}^{10} (pq^* - cq^*) f_D[x] dx = 11.961$$

The first term captures the case when there is enough stock to cover demand (and turns out to equal zero in the optimum), and the latter term is for the case when demand exceeds supply.

$$q^* = F_D^{-1} \left[\frac{p-c}{p} \right] \quad (8.1)$$

Any decision problem needs to deal with three elements: the random **state variables** that describe the state of the system, **control variables** (actions) that the decision-maker sets, and a **policy** that the decision-maker uses to set control variables based on the state of the system. An optimal policy is one where the expected value of the objective is optimized over all possible states and likelihoods. In the example in [Exercise 8.1](#), the stock quantity is the control variable, the demand is a state variable, and the optimal policy is to set the stock quantity based on Eq. (8.1). A suboptimal policy for this example could have been “ $q = \frac{10}{3}$ regardless of state,” for example.

If the decision is repeated multiple times using new and independent outcomes of D , each time the revealed value of the decision will vary. For example, in one instance $d_t = 2$, in which case the realized profit ϕ_t is $\phi_t = 30 - 4.83(10) = -18.3$. The policy is optimal because if the event is performed multiple times, as it approaches infinity the expected profit, $\lim_{T \rightarrow \infty} (\frac{1}{T} \sum_{t=0}^T \phi_t)$, is optimized.

In more complex decision problems, the state variable at time t is dependent on the state variable in the previous time $t-1$. These are called *sequential decision problems*. In such problems, the sequential state dependency is modeled with a **state transition function** $S_{t+1} = S^M[S_t, a_t, W_{t+1}]$, where S_t is a state at step t , a_t is the action or control taken at step t , and W_{t+1} is newly received exogenous information.

The optimal policy problem is to select a policy π such that the expected value over all relevant time steps T is maximized, that is, Eq. (8.2) ([Powell, 2011](#)), where γ^t is a discount rate at step t and A_t^π is the optimal action under policy π as a function of the state S_t .

$$\max_{\pi} E^{\pi} \left[\sum_{t=0}^T \gamma^t \phi_t^{\pi} [S_t, A_t^{\pi}[S_t]] \right] \quad (8.2)$$

Bellman's (1957) efforts to simplify sequential problems using recursive functions can be applied to Eq. (8.2). In this case, the recursive form of Eq. (8.2) conditional on state S_t , called the Bellman Equation, is shown in Eq. (8.3) as an optimal policy.

$$V_t[S_t] = \max_{a_t \in A_t} (\phi_t[S_t, a_t] + \gamma E[V_{t+1}[S_{t+1}, a_t, W_{t+1}] | S_t]), \quad t < T \quad (8.3)$$

The optimal policy seeks an action a_t at time step t to maximize the value V_t , which depends on the sum of the immediate payoff ϕ_t and a discounted conditional expectation of V_{t+1} given S_t . The strategy to solving Eq. (8.3) depends on if the horizon T is finite or infinite. For example, an infinite horizon problem may involve finding a fixed point for V_t and using that to determine the optimal action. In a finite horizon problem, a typical strategy is to start from step T and proceed backwards to get the optimal decision at t . Let us demonstrate the fixed horizon problem with [Exercise 8.2](#).

Exercise 8.2

Consider a server allocation problem over two periods ($T=1$) with no discounting to minimize total user delay. There are three servers available to assign to one of two queues, where s_{ti} is the number of servers at queue i in period t . Users choose which queue to enter simultaneously as the server allocation decision in that period; once chosen, the user does not have the option to switch or balk. In the initial period, two users are equally likely to choose either queue to be served, where n_{ti} is the number of users choosing queue i in period t . Let w_{ti} be the average delay experienced at queue i in period t , and set equal to $w_{ti} = \frac{n_{ti}}{s_{ti}}, s_{ti} > 0$, and $w_{ti} = 10$ if $s_{ti} = 0$. In subsequent periods users choose a queue probabilistically based on the delays reported in the previous period: $\Pr[i | \{i, j\}]_{t+1} = \frac{1}{1 + \exp(-w_{ti} - w_{tj})}$. Using Eq. (8.3) determine the optimal decision at the first period.

In this problem, the decision of the operator in the first period signals to the users for the second period decision. In each period, there are three user states: $\{(0, 2), (1, 1), (2, 0)\}$. For each user state there are four decisions for server allocation: $\{(0, 3), (1, 2), (2, 1), (3, 0)\}$. If a backward recursion is sought, we need to first determine the delays for the 12 combinations of action and state:

$$1 : n_1 = (0, 2), s_1 = (0, 3) : w_1 = \left(10, \frac{2}{3}\right) \rightarrow \Pr[1]_2 = 0, \Pr[2]_2 = 1$$

$$2 : n_1 = (0, 2), s_1 = (1, 2) : w_1 = (0, 1) \rightarrow \Pr[1]_2 = 73\%, \Pr[2]_2 = 27\%$$

$$3 : n_1 = (0, 2), s_1 = (2, 1) : w_1 = (0, 2) \rightarrow \Pr[1]_2 = 88\%, \Pr[2]_2 = 12\%$$

- 4 : $n_1 = (0, 2), s_1 = (3, 0) : w_1 = (0, 10) \rightarrow \Pr[1]_2 = 1, \Pr[2]_2 = 0$
- 5 : $n_1 = (1, 1), s_1 = (0, 3) : w_1 = \left(10, \frac{1}{3}\right) \rightarrow \Pr[1]_2 = 0, \Pr[2]_2 = 1$
- 6 : $n_1 = (1, 1), s_1 = (1, 2) : w_1 = \left(1, \frac{1}{2}\right) \rightarrow \Pr[1]_2 = 38\%, \Pr[2]_2 = 62\%$
- 7 : $n_1 = (1, 1), s_1 = (2, 1) : w_1 = \left(\frac{1}{2}, 1\right) \rightarrow \Pr[1]_2 = 62\%, \Pr[2]_2 = 38\%$
- 8 : $n_1 = (1, 1), s_1 = (3, 0) : w_1 = \left(\frac{1}{3}, 10\right) \rightarrow \Pr[1]_2 = 1, \Pr[2]_2 = 0$
- 9 : $n_1 = (2, 0), s_1 = (0, 3) : w_1 = (10, 0) \rightarrow \Pr[1]_2 = 0, \Pr[2]_2 = 1$
- 10 : $n_1 = (2, 0), s_1 = (1, 2) : w_1 = (2, 0) \rightarrow \Pr[1]_2 = 12\%, \Pr[2]_2 = 88\%$
- 11 : $n_1 = (2, 0), s_1 = (2, 1) : w_1 = (1, 0) \rightarrow \Pr[1]_2 = 27\%, \Pr[2]_2 = 73\%$
- 12 : $n_1 = (2, 0), s_1 = (3, 0) : w_1 = \left(\frac{2}{3}, 10\right) \rightarrow \Pr[1]_2 = 1, \Pr[2]_2 = 0$

For each of these 12 branches, we can now determine the probabilities for each of the three user states and select the action that minimizes expected delay, where expected delay is $\phi_t = \sum_{s \in S} p_s (n_{t1} w_{t1} + n_{t2} w_{t2})$ and p_s is probability of a state occurring. This is presented in [Table 8.1](#).

The optimal policy for the operator is to allocate either to (1,2) or (2,1) in the initial period. This results in only branches {2, 3, 6, 7, 10, 11} being active for the second stage. Depending on the choices of the users, the operator would choose either (1,2) or (2,1) again.

The second period decisions based on (0,3) or (3,0) outcomes in period 1 have better performance than the other period 2 branches. However, the cost in the first period ϕ_1^* is too risky to be worth signaling to the passengers to keep those branches open. If the penalty cost of having no server was lower, then it is potentially better to seek either (0,3) or (3,0) as solutions instead.

The example illustrates a well-known issue in sequential decision problems: as the number of dimensions in the problem increases, the computational complexity increases exponentially. [Powell \(2011\)](#) calls the issue *the three curses of dimensionality*.

Definition 8.1 ([Powell, 2011](#)). *The three curses of dimensionality with sequential decision problems involve:*

- *State space: if the state variable S_t has I dimensions, and if S_{ti} can take on L possible values, then there may be L^I different states.*
- *Outcome space: The random variable W_t may have J dimensions. If W_{tj} can take on M outcomes, then the outcome space may take on up to M^J outcomes.*

- *Action space: The decision vector x_t might have K dimensions. If x_{tk} can take on N outcomes, the feasible space may involve N^K options.*

[Powell \(2011\)](#) describes four broad categories of policies with varying treatment of the curses of dimensionality. For the approximation categories, the types of methods can further break down into lookup table policies, parametric policies, and nonparametric policies.

- Myopic policies: no look-ahead
- Look-ahead policies: directly solving an approximation of the problem over a rolling horizon, for example, tree search, rollout heuristics
- Policy function approximations: using an approximation function to mimic the desired policy, for example, Q-learning uses lookup tables to assign rewards to state-action pairs
- Value function approximations: using an approximation function to estimate the value function to solve the Bellman equation based on it

For example, the Least Squares Monte Carlo simulation (LSM) from [Longstaff and Schwartz \(2001\)](#) discussed later is a value function approximation method that uses nonparametric kernel regression to estimate the expected value on an independently sampled path.

Sequential decision problems also work for stochastic processes. A stochastic process is a function of time and may exhibit nonstationary mean and variance. A classic example of a stochastic process is the random walk. An agent starts at $x=0$ at time $t=0$, and each unit of time there is a 50% probability of increasing x by one or decreasing it by one. The random variable X_t is the location of the agent after t units. Although the expected value remains 0 for all $t > 0$, the variance increases over time. Stochastic processes where the expected value does not change but the variance increases over time are called martingales. Martingales are used to explain the concept of “gambler’s ruin,” where strategies to increase bets to compensate for prior losses eventually lead to a gambler’s loss if they have a finite pool to bet from. The following discussion is drawn from [Øksendal \(1992\)](#).

One type of martingale is the Wiener process $W[t]$, which is also a limit of the random walk as the size of each time step approaches zero. It is a type of Lévy process, which is a stochastic process with independent stationary increments. In other words, the outcome of one increment in the Wiener process at time t is independent of the outcome at any other time $s \neq t$. A Wiener process exhibits the following properties: $W[t] - W[s]$, $t > s$, is Gaussian with mean of 0 and variance $t-s$.

Brownian motion is a type of Wiener process that includes deterministic trends defined by a linear drift rate. It can be expressed as a stochastic

Table 8.1 Summary of solution for [Exercise 8.2](#)

	1	2	3	4	5	6	7	8	9	10	11	12
a_2^*	(0,3)	(2,1)	(2,1)	(3,0)	(0,3)	(1,2)	(2,1)	(3,0)	(0,3)	(1,2)	(1,2)	(3,0)
V_2^*	1.33	1.65	1.77	1.33	1.33	1.70	1.70	1.33	1.33	1.77	1.65	1.33
(0,3)			(1,2)			(2,1)			(3,0)			
ϕ_1	10.50			2.25			2.25			10.50		
V_1	11.83			3.96			3.96			11.83		

differential equation (SDE) in Eq. (8.4a) and as a geometric Brownian motion (GBM) in Eq. (8.4b). The drift μ parameter determines the trend of the mean over time. The volatility σ parameter determines how much the process may fluctuate over time. The term dW is an increment in a Wiener process.

$$dX = \mu dt + \sigma dW \quad (8.4a)$$

$$dX = \mu X dt + \sigma X dW \quad (8.4b)$$

In GBM, $X[t] - X[s]$, $t > s$, is lognormally distributed. A GBM trajectory over time is illustrated in Fig. 8.2. Two realizations of the stochastic process are simulated from an initial value of $x[0] = 100$ and $\mu = 0.1$, $\sigma = 0.4$. The realizations are shown for $t = [0, 5]$ followed by a projection of the expected value and 95% confidence interval of the distribution for $t = (5, 6]$. The analytic solution to Eq. (8.4b) can be obtained using the Ito integral resulting in Eq. (8.5).

$$X[t] = X[0] \exp \left(\left(\mu - \frac{\sigma^2}{2} \right) t + \sigma W[t] \right) \quad (8.5)$$

The expected value of this process is $E[X[t]] = X[0] \exp(\mu t)$ and the variance is $Var[X[t]] = X[0]^2 \exp(2\mu t)(\exp(\sigma^2 t) - 1)$. The probability density function for X is defined in Eq. (8.6). The GBM parameters can be estimated using the maximum likelihood method.

$$f_X[x; \mu, \sigma, x_0, t] = \frac{1}{\sqrt{2\pi x \sigma \sqrt{t}}} \exp \left(-\frac{\left(\ln x - \ln x_0 - \left(\mu - \frac{\sigma^2}{2} \right) t \right)^2}{2\sigma^2 t} \right) \quad (8.6)$$

Trajectories are simulated by applying Eq. (8.7) recursively forward in discrete time steps of size h , where z_t is a random draw from a standard normal variate.

$$x[t+h] = x[t] \exp \left(\sigma \sqrt{h} z_t + \mu h \right) \quad (8.7)$$

The GBM is used as the underlying state variable for sequential decision problems involving states that span a range of continuous values over long periods of time, such as population or demand growth for a technology. In addition to the GBM, other stochastic processes include the Ornstein-Uhlenbeck (O-U) process, also called a mean-reverting process, and the Poisson jump process for modeling discrete jumps in state variables. The O-U

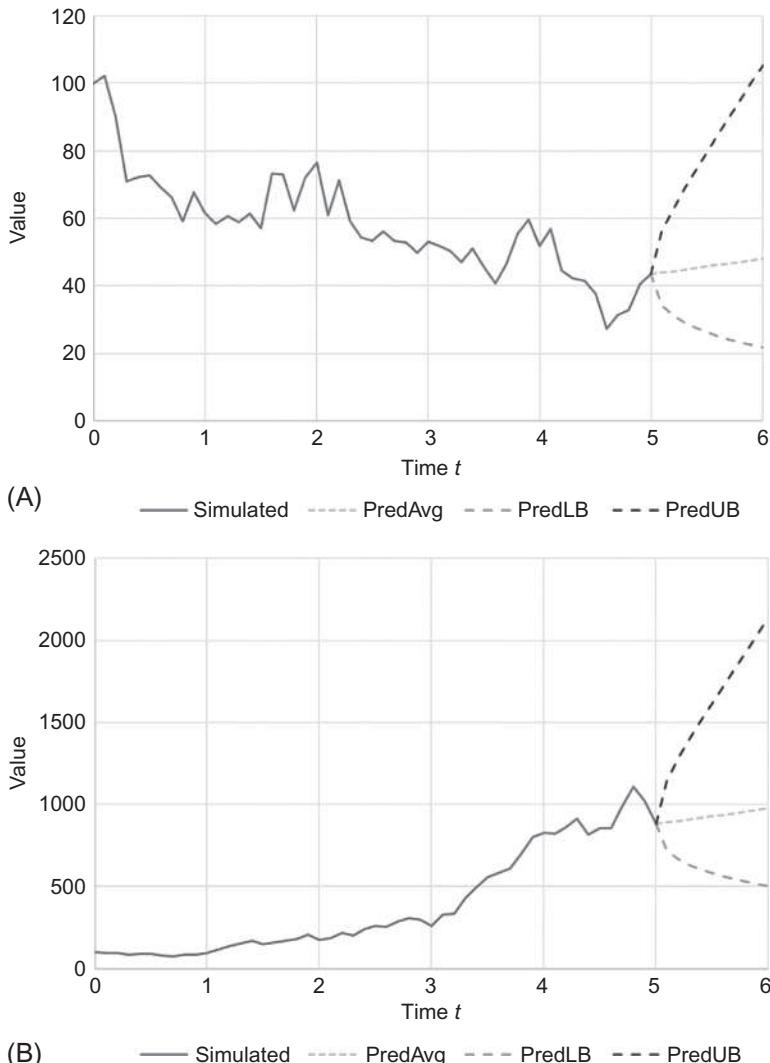


Fig. 8.2 Examples of stochastic processes: (A) first and (B) second instances of a simulation ($t=[0,5]$) and projection ($t=(5,6]$) of a geometric Brownian motion with $\mu=0.1$, $\sigma=0.4$.

process is a stationary process that is designed for variables that have fluctuations with some degree of inertia but do not tend to diverge from a fixed value. These include hourly or daily traffic patterns, weather, and other similar events. The SDE for the O-U process is shown in Eq. (8.8), where θ is a mean reversion rate, μ is the long-term mean, and σ is the volatility.

$$dx = \theta(\mu - x) dt + \sigma dW \quad (8.8)$$

The solution to Eq. (8.8) is given by Eq. (8.9) along with the expected value and variance in Eqs. (8.10a) and (8.10b) (see Tsekrekos, 2010).

$$x[t] = x[0] \exp \left(-\left(\theta + \frac{1}{2} \sigma^2 \right) t + \sigma W \right) + \frac{\theta \mu}{\theta + \frac{1}{2} \sigma^2} \left(1 - \exp \left(-\left(\theta + \frac{1}{2} \sigma^2 \right) t \right) \right) \quad (8.9)$$

$$E[x[t]] = \mu + (x[0] - \mu) \exp(-\theta t) \quad (8.10a)$$

$$\begin{aligned} Var[x[t]] &= \frac{\mu^2 \sigma^2}{2\theta - \sigma^2} \\ &\quad - \exp(-2\theta t) (x[0] - \mu)^2 + \frac{2\mu\sigma^2(x[0] - \mu)}{\theta - \sigma^2} \exp(-\theta t) \\ &\quad + \frac{2\theta^2(x[0] - \mu)^2 - \theta\sigma^2 x[0](3x[0] - 2\mu) + \sigma^4 x[0]^2}{(\theta - \sigma^2)(2\theta - \sigma^2)} \exp(-(2\theta - \sigma^2)t) \end{aligned} \quad (8.10b)$$

Stochastic processes are used to model sequential decision problems where the underlying state variables are dependent on prior values and represent continuous, real numbers. One such problem is the optimal stopping problem: as a decision-maker monitors a state variable over time, at what point should they decide to make an irreversible decision with asymmetric costs? A stop can mean investing in a high cost capital investment or technology, stopping a program with high start-up costs, switching an operation with high switching costs, and so on.

Before discussing in further detail in the next section, we present a classic example of the optimal stopping problem: the *secretary problem*. In this problem, an employer is looking to hire the best secretary out of n candidates but must interview each sequentially and decide after the interview whether to stop the search and hire that candidate (see Bruss, 1984). It is assumed that each candidate can be ranked unambiguously once interviewed relative to other candidates interviewed thus far. The objective is to design a policy that would maximize the expected ranking of the applicant from the group.

It is found that the best policy is to reject the first $\frac{n}{e}$ applicants, and then accept the first applicant afterward that is better than all prior candidates. This policy can find the top candidate 37% of the time regardless of the number of applicants. Bruss (1984) provides a proof of the optimal policy.

8.3 OPTIMAL TIMING UNDER UNCERTAINTY: REAL OPTIONS METHODS

For more complex optimal timing problems involving stochastic processes, one can use *real options* theory.

Definition 8.2 ([Trigeorgis, 1996](#)). *Real options* involve discretionary decisions or rights, with no obligation, to acquire or exchange an asset for a specified alternative price. The ability to value real options (e.g., to defer, expand, contract, abandon, switch use, or otherwise alter a capital investment) has brought a revolution to modern corporate resource allocation.

In project and investment appraisal, the classic approach of “net present value” (NPV) determines the expected benefits and costs of different alternatives, subject to discounting over time, to select the alternative with the best expected performance. This perspective is flawed because all alternatives are treated as commitments in the present time, even if they have not yet been invested in and may not have to be built until the future. A real options perspective differs from this view because the investment under uncertainty is viewed as an opportunity instead of a commitment. To illustrate, a project that is invested in right away might yield an expected profit over its lifetime of $-\$10M$ in present dollars. NPV analysis would reject this project. However, in an uncertain environment, there may be a 30% chance that sometime in the next 5 years the valuation of the project becomes a present value profit of $\$100M$. In that case, the best option may be to defer to the project to monitor the stochastic variable(s).

In the context of transportation network investments, [Chow et al. \(2011\)](#) (expanded from [Luehrman, 1998](#)) show how the change in perspective from NPV analysis to real options significantly expands the decision space, as shown in [Fig. 8.3](#). The decision space for traditional NPV is composed of only regions 1 (yes) and 6 (no). By allowing for opportunities from monitoring under uncertainty, the decision space expand beyond yes/no decisions to other regions (2—maybe now, 3—probably later, 4—maybe later, 5—probably never) for a much richer set of decisions for each project. The spectrum of decision outcomes also correspond in a network context to potential benefits of network synergies and growth as illustrated in [Fig. 8.3](#).

The added opportunity value is quantified as flexibility. [Exercise 8.3](#) illustrates the value of such flexibility.

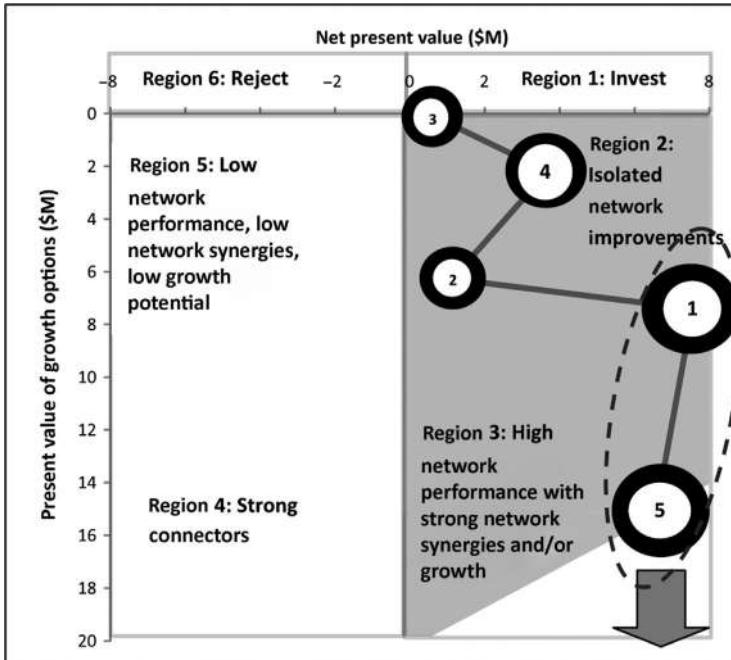


Fig. 8.3 Expanding the NPV decision space (top region) to four other regions as opportunities. (Source: [Chow et al., 2011](#).)

Exercise 8.3

An investment has a net profit P , which has a triangular distribution with the three vertices at $(-y, 0, y)$. If the investor can purchase an option contract that gives them the right to purchase the investment after the outcome of the underlying investment is known, what is the equilibrium price of that contract?

Under a commitment perspective, the expected profit of the investment is 0. Under the option setting, the investor only pays a fee x to purchase the right to buy the investment after the underlying uncertainty is revealed. If the investment falls in the loss region, the investor can choose not to purchase the investment and lose only x . If the investment is positive, the investor can gain as much as $y - x$. This trade-off is illustrated in Fig. 8.4.

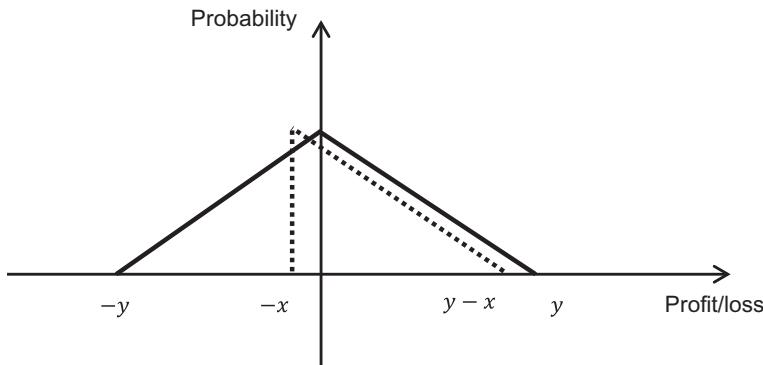


Fig. 8.4 Distributions of profit under commitment (solid) and option (dashed) investments.

As a triangular distribution, the maximum value of the PDF for the commitment at 0 is $1/y$.

The expected profit in this case is 50% a loss of $-x$ and a 50% gain of $(\frac{1}{6}y - x)$, where $\frac{1}{6}y$ is the conditional expectation of the right-hand side of the triangular distribution prior to including the fee. The total expected profit is therefore:

$$E[P] = \frac{1}{12}y - x$$

where $\frac{1}{12}y$ is called the option premium because that is the difference from the option value (which is also $\frac{1}{12}y$) and the NPV. If this option were to be traded, the equilibrium value of the contract is determined from the fixed point:

$$x^* = \frac{1}{12}y - x^* \rightarrow x^* = \frac{y}{24}$$

If $x^* < \frac{y}{24}$, it is worth it for the investor to purchase the option. If $x^* = \frac{y}{24}$, the investor is indifferent between the option contract and investing in the commitment.

This value would change over time if the distribution is time dependent and/or if revealing the uncertainty is a function of time (as it is in most cases). For example, in American and European financial options, the option contract has an expiration date. The closer it gets to the expiration date, the smaller the variance in the distribution. As variance decreases, the premium due to flexibility also decreases. If the option is “in the money” prior to expiration, it would be the value of the investment, whereas if it is “out of the money” it approaches zero.

Trigeorgis (1996) illustrates this relationship between NPV, option and premium in Eq. (8.11). The static NPV is the traditional value. The expanded NPV is the option value and measures the opportunity to invest

including the flexibility to set the decision to a later time. Option premiums form the difference, and along with the expanded NPVs, are always nonnegative.

$$\text{Expanded NPV} = \text{Static NPV} + \text{Option Premiums} \quad (8.11)$$

Real options theory is essentially about solving sequential decision problems applied in a continuous time setting with an emphasis on quantifying the value function for different types of decisions. Dixit and Pindyck (1994) illustrate an analytical solution for an infinite horizon optimal stopping problem for an investment in which the present value of the project if implemented, R , is a GBM.

First consider the deterministic case where the revenue evolves as noted in Eq. (8.12) with initial project value $R[0] = R_0$ and μ as the value growth rate.

$$R[t] = R_0 e^{\mu t} \quad (8.12)$$

The project costs an amount I to be invested in and there is a discount rate ρ , $\rho > \mu$. The value $V[R]$ of the investment opportunity for a current value of R is shown in Eq. (8.13).

$$V[R] = (R e^{\mu t} - I) e^{-\rho t} \quad (8.13)$$

If $\mu \leq 0$, then the optimal decision is to invest if $R - I > 0$. If $0 < \mu < \rho$, the deterministic optimum can be found by first-order condition for Eq. (8.13) as shown in Eq. (8.14).

$$\frac{dV}{dt} = -(\rho - \mu) R e^{-(\rho - \mu)t^*} + \rho I e^{-\rho t^*} = 0 \quad (8.14)$$

Solving for t^* , we get $t^* = \frac{1}{\mu} \ln \left(\frac{\rho I}{(\rho - \mu)R} \right)$. Substituting this into Eq. (8.12) we obtain the critical R^* in Eq. (8.15).

$$R^* = \frac{\rho I}{\rho - \mu} \quad (8.15)$$

The solution is clearly only dependent on the difference $\rho - \mu$. By further substituting the optimal time into Eq. (8.13), the value of the investment opportunity is shown in Eq. (8.16).

$$V[R] = \begin{cases} \left(\frac{\mu I}{\rho - \mu} \right) \left(\frac{(\rho - \mu)R}{\rho I} \right)^{\frac{\rho}{\mu}}, & R \leq R^* \\ R - I, & R > R^* \end{cases} \quad (8.16)$$

In the case where R is now a GBM, the investment opportunity is a real option with a value $V[R]$. The Bellman equation is derived for an increment in time dt by setting the total expected option value $\rho V dt$ equal to the expected rate of value appreciation, $E[dV]$, in Eq. (8.17).

$$\rho V dt = E[dV] \quad (8.17)$$

Using Ito's Lemma and substituting the GBM for dV leads to Eq. (8.18).

$$\frac{1}{2}\sigma^2 R^2 V''[R] + \mu R V'[R] - \rho V = 0 \quad (8.18)$$

Eq. (8.18) is solved by Dixit and Pindyck (1994) using the initial condition ($V[0]=0$), value matching boundary condition ($V[R^*]=R^*-I$), and “smooth-pasting” condition ($V'[R^*]=1$). The option value is shown in Eq. (8.19) with critical revenue in Eq. (8.20).

$$V[R] = AR^b \quad (8.19a)$$

$$A = \frac{(b-1)^{b-1}}{b^b I^{b-1}} \quad (8.19b)$$

$$b = \frac{1}{2} - \frac{\mu}{\sigma^2} + \sqrt{\left(\frac{\mu}{\sigma^2} - \frac{1}{2}\right)^2 + \frac{2\rho}{\sigma^2}} \quad (8.19c)$$

$$R^* = \frac{bI}{b-1} \quad (8.20)$$

Consider Exercise 8.4 to illustrate the model.

Exercise 8.4

For a candidate project in which its present value evolves as a GBM with $\mu=0.1$, $\sigma=0.4$ with discount rate $\rho=0.12$ and $I=\$100$, plot the option value as a function of the project value and identify the critical revenue. Compare this to a second project with value following GBM with $\mu=0.05$, $\sigma=0.2$.

Plugging in the parameters we obtain the following:

$$b = 1.106$$

$$A = 0.432$$

$$R^* = \$1042.44$$

The value of R^* happens to be over 10 times higher than I . This means that when monitoring the revenue, it is only worth investing the present value of \$100 when $R > R^* = \$1042.44$. The value of the investment opportunity at the critical revenue is $V^* = \$942.44$ for the optimal timing policy. Under NPV analysis, the critical project value is simply \$100. The significant gap between the investment cost and the critical project value under uncertainty is because of the combination of a relatively high volatility and low difference between the mean growth rate and discount rate. For example, when $\sigma \rightarrow 0$, $R^* \rightarrow \$600$ and $V^* \rightarrow \$500$. A value of $\rho=0.139$ yields the same critical project value and option value.

When $R = \$10$, the net present value is $NPV = 10 - 100 = -\$90$, which suggests not investing in the project. However, the presence of uncertainty makes this option still of value in monitoring, since $V[10] = \$5.52$. This means even if the current project value is only \$10, under the current settings the opportunity to invest in this project is worth \$5.52. The option value suggests a deferral premium of $\$5.52 - (-\$90) = \$95.52$.

Consider a second project for comparison in which its project value is also a GBM with halved parameter values $\mu = 0.05$, $\sigma = 0.2$. Its critical project value is $R_2^* = \$223.19$ ($V_2^* = \$123.19$). If the first project has a value of $R_1 = \$110$ ($V_1 = \$78.34$) while the second has value of $R_2 = \$150$ ($V_2 = \$59.97$), under NPV both would be worth investing immediately but the second project would be chosen over the first. Using real option valuation, the optimal decision is to defer both projects, but rank project 1 over project 2 as having over 30% more promise. If, after 1 year both project values then evolve to \$500, the decision is now to invest immediately in project 2 ($V_2 = \$531.15 > R_2^*$), which should result in 27% higher option value than to continue deferring project 1 ($V_1 = \$418.13$). The reasoning is that the likelihood for project 2 to reach \$500 from \$150 is quite low compared to project 1's rise from \$110 to \$500, but if it does happen it is now worth much more. The range of option values is shown for both projects in the plot in Fig. 8.5.

To further illustrate this point, consider project 1 with $R = \$500$. The value of the policy at \$418.13 can be verified with Monte Carlo simulation. If different sample paths were generated from \$500 such that the time when $R^* = \$1042.44$ is crossed (some may occur so far in the future that the present value is \$0), and then to discount the project back to time 0, the average over all those sample path discounted values should approach \$418.13 as the number of sample paths approaches infinity. This is left to the reader to verify.

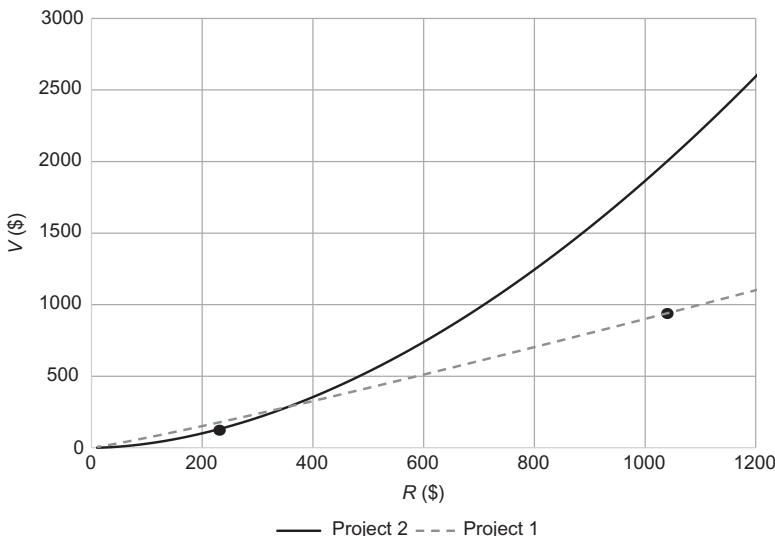


Fig. 8.5 Plot of project value R and the corresponding option value V for two projects.

Analytical solutions are more difficult to attain when the project value is a more complex function of an underlying asset. If the option value V of a project whose value $R[Q]$ is a function of an underlying stochastic process Q , the relationship between V and R needs to be derived. As pointed out in [Dixit and Pindyck \(1994\)](#), however, it is possible instead to relate R and V to Q separately, and then to tie them back together as boundary conditions under the optimal exercise threshold. We construct an artificial portfolio relating V to Q in the second-order differential equation in Eq. (8.21) with a solution $V[Q] = A_1 Q^{b_1} + A_2 Q^{b_2}$.

$$\frac{1}{2}\sigma^2 Q^2 F''[Q] + (r - \delta) Q V'[Q] - r V[Q] + Q = 0 \quad (8.21)$$

With the absorbing zero condition for GBMs we assume A_2 . The value-matching ($V[Q^*] = R[Q^*] - I$) and “smooth-pasting” conditions ($V'[Q^*] = R'[Q^*]$) then lead to the solution in Eq. (8.22) where b_1 needs to be determined.

$$V[Q] = A_1 Q^{b_1} \quad (8.22a)$$

$$A_1 = \frac{(b_1 - 1)^{b_1 - 1}}{(\delta b_1)^{b_1} I^{b_1 - 1}} \quad (8.22b)$$

$$Q^* = \frac{b_1 \delta I}{b_1 - 1} \quad (8.22c)$$

$$R^* = \frac{b_1 I}{b_1 - 1} \quad (8.22d)$$

For example, in network design the system performance can be treated as the project value, but that value can depend on a set of OD demand that evolves over time as GBM. This is especially problematic in the case for transportation projects that can exhibit complex nonlinear properties due to congestion effects. Examples of analytical models in real options for transportation include [Galera and Soliño \(2010\)](#) for valuation of highway concessions and [Li et al. \(2015\)](#) for transit technology selection and timing.

Despite these studies, in general many optimal timing decisions require numerical methods to solve, especially when considering finite horizons, complex project value functions, and multidimension stochastic processes. Three types of methods are described in [Trigeorgis \(1996\)](#): finite difference, binomial lattice, and Monte Carlo simulation.

Finite difference methods ([Brennan and Schwartz, 1977](#)) require establishing a differential equation to relate the option value to the stochastic

variables and then using discrete finite difference methods to numerically approximate the solution. While [Guo et al. \(2017\)](#) optimize switching options for real-time transport operations in an infinite horizon, they also must use finite difference to compute the value of the derivative of one of the terms in the system of equations.

Binomial lattice ([Cox et al., 1979](#)) divides the evolution of a stochastic process into discrete branches. By increasing the number of intervals up to the time horizon, the solution to the binomial tree converges to the true solution. This method is very effective for problems dealing with single- or low-dimensional stochastic processes. Several studies have applied binomial lattice method to transport problems: for example, [Garvin and Cheah \(2004\)](#) used it to evaluate Build-Operate-Transfer infrastructure projects and [Vodopivec and Miller-Hooks \(2017\)](#) used it to determine the optimal time to outsource on-demand mobility to nearby taxis under travel time uncertainty.

The third type is Monte Carlo simulation ([Boyle, 1977](#)). The original method involves simulation of multiple branching paths as look-ahead policies like with finite difference and binomial lattice method. [Longstaff and Schwartz \(2001\)](#) proposed a more computationally efficient simulation method called Least Squares Monte Carlo simulation (LSM), which generates independent sample paths and uses least squares regression to link their results together in valuations. It is a value function approximation policy based on nonparametric, kernel regression. The advantage of this approach is that it evaluates option values based on high-dimensional stochastic processes in a computationally efficient manner. [Gamba \(2002\)](#) proposed an extension of LSM to multiple interacting options under three stylized settings. Several studies have used LSM to obtain the option value: [Zhao et al. \(2004\)](#) for highway development, [Chow and Regan \(2011b\)](#) for timing network design investments, [Chow and Regan \(2011c\)](#) for timing multiple options within a network, [Zheng and Chen \(2016\)](#) for optimizing fleet replacement decisions under fuel price uncertainty, and [Chow \(2016b\)](#) for selecting routes to deploy a fleet of UAVs over time.

The LSM algorithm is a Monte Carlo simulation-based approximate dynamic programming method for determining the optimal policy for a timing problem. It simulates independent sample paths of the stochastic processes up to a finite horizon and then uses backward recursion to evaluate the Bellman equation in each discretized time step and sample path. The conditional expectation of the value function at each time step is approximated by constructing a basis function using the payoff at each sample path with positive option value as independent variables and then estimating the parameters

of the basis function using least squares. The estimated model is then applied to the time step and sample path payoff value to approximate the value function. Different basis functions can be used, including Laguerre, Hermite, Legendre, Chebyshev, Gegenbauer, and Jacobi polynomials. The Laguerre polynomial is shown in Eq. (8.23a) and the Hermite polynomial in Eq. (8.23b).

$$L_m[x] = e^{-\frac{x}{2}} \frac{e^x}{m!} \frac{d^m}{dx^m} [x^m e^{-x}] \quad (8.23a)$$

$$L_m[x] = \sum_{k=0}^m (-1)^m e^{\frac{x^2}{2}} \frac{d^m}{dx^m} e^{-\frac{x^2}{2}} \quad (8.23b)$$

Algorithm 8.1 is presented as follows.

Algorithm 8.1: (Longstaff and Schwartz, 2001). Least Squares Monte Carlo Simulation (LSM) Algorithm

Inputs: GBM Q with μ, σ ; initial values $Q[0], R[Q[0]]$; investment cost I ; finite horizon T and number of time step n ; payoff function $R[Q]$; discount rate ρ ; number of radial basis functions m for a chosen basis function L

1. Simulate P sample paths $\hat{Q}[\omega, t]$ and compute $R[\hat{Q}[\omega, t]]$ where $\omega \leq P$, $t \leq n$, and a time step has length $h = T/n$.
2. For $t = n$
 - a. Compute $R[\hat{Q}[\omega, n]]$ for each path $\omega \leq P$.
 - b. Let $\hat{V}[\omega, n] := \max(R[\hat{Q}[\omega, n]] - I, 0)$
 - c. Let $t := t - 1$.
3. For $0 < t < n$
 - a. For each $1 \leq \omega \leq P$
 - i. Construct a weighted sum of basis functions for each sample path where $\hat{V}[\omega, t+1] > 0$ as shown in Eq. (8.24), and estimate \hat{a}_j^t using least squares method.

$$e^{-\rho h} E[\hat{V}[t+1] | \omega] = \sum_{j=0}^m \hat{a}_j^t L_j[R[\hat{Q}[\omega, t]]] \quad (8.24)$$

- ii. If $R[\hat{Q}[\omega, t]] - R > \sum_{j=0}^m \hat{a}_j^t L_j[R[\hat{Q}[\omega, t]]]$
 1. Update stopping decision: $\hat{D}[\omega, t] := 1$, $\hat{D}[\omega, s] := 0 \forall s > t$
 2. $\hat{V}[\omega, t] := R[\hat{Q}[\omega, t]]$
 - iii. Else
 1. $\hat{D}[\omega, t] = 0$
 2. $\hat{V}[\omega, t] := \sum_{j=0}^m \hat{a}_j^t L_j[R[\hat{Q}[\omega, t]]]$
- b. Let $t := t - 1$.

4. For $t=0$

a. Option value is determined as in Eq. (8.25).

$$V[Q[0]] := \max \left(R[Q[0]] - I, \frac{1}{P} \sum_{\omega=1}^P \sum_{\tau=1}^n e^{-\rho\tau} \hat{V}[\omega, \tau] \hat{D}[\omega, \tau] \right) \quad (8.25)$$

b. Optimal decision is: $D[0] = 1$ (invest now) if $V[Q[0]] = R[Q[0]] - I$
else $D[0] = 0$ (defer).

Outputs: $V[Q[0]]$, $D[0]$

[Longstaff and Schwartz \(2001\)](#) partially proved that LSM converges asymptotically to the unbiased estimator of the true option value, although there is a tendency to have low bias for small values of m and high bias for small values of P ([García, 2003](#)). [Stentoft \(2004\)](#) further proved that the LSM approximation converges to the true value as $P \rightarrow \infty$ if $m = m[P]$ is increasing in P such that $m \rightarrow \infty$ and $\frac{m^3}{P} \rightarrow 0$. The LSM algorithm is illustrated in [Exercise 8.5](#) for a highway setting.

Exercise 8.5

You are considering building a two-lane toll road running parallel to an existing six-lane highway (three in each direction). Peak hour travel demand Q is assumed to be a GBM that varies over the years with $\mu = 0.05$, $\sigma = 0.20$, and $Q[0] = 6000$. The existing highway has a link cost of $c_e = 30 \left(1 + 0.15 \left(\frac{x_e}{6000} \right)^4 \right)$. The new toll road with \$5 toll would have a link cost of $c_t = 30 \left(1 + 0.15 \left(\frac{x_t}{2000} \right)^4 \right) + 15$.

Flow distribution is based on user equilibrium. The annual revenue is $14,600x_t$, resulting in a present value of $R = \frac{14,600x_t}{\rho - \mu}$, where $\rho = 0.12$. The investment cost (including present value of operating costs) is $I = \$100M$. For a horizon of 5 years, using $n = 10$, $P = 100$, and $m = 3$, use [Algorithm 8.1](#) to determine the Hermite polynomial regression model for estimating the continuation value at $t = 9$ and illustrate its use for two sample paths.

Due to the size of the R and V , they are in units of \$M.

For $m = 3$, the Hermite polynomials are: $\{1, x, x^2 - 1, x^3 - 3x\}$

From the simulation, only 12 of the 100 sample paths are in the money at $t = 9$. The attributes of these 12 “in-the-money” paths (where $R[\hat{Q}[\omega, 9]] - I > 0$) are summarized in [Table 8.2](#).

Table 8.2 Twelve “in-the-money” simulated path values for $t=9, 10$

Path	$\hat{Q}[\omega, 9]$ (\$M)	$x_i[\omega, 9]$	$R[\hat{Q}[\omega, 9]]$ (\$M)	$\hat{V}[\omega, 10]$ (\$M)
1	9630.625	1386.494	289.183	185.549
2	10,620.570	1981.785	413.344	0
3	11,912.808	2543.027	530.403	315.531
4	8695.595	583.978	121.801	257.843
5	9708.690	1441.963	300.752	37.075
6	8969.642	843.290	175.886	281.392
7	8832.981	715.821	149.300	76.154
8	9742.846	1465.686	305.700	291.421
9	9453.056	1253.592	261.463	100.104
10	13,474.514	3084.621	643.364	509.458
11	8683.227	571.962	119.295	246.187
12	11,636.501	2435.127	507.898	97.234

By discounting the simulated option values at $t=10$ by one $h=0.5$ and setting them as the dependent variables, we estimate the regression model for the weighted sum of Hermite polynomials:

$$\begin{aligned} e^{-\rho h} E[\hat{V}[10] | R[\hat{Q}[\omega, 9]]] = & 137.869 + 1.384 R[\hat{Q}[\omega, 9]] \\ & - (7.830 \times 10^{-4}) (R[\hat{Q}[\omega, 9]]^2 - 1) \\ & + (1.020 \times 10^{-5}) (R[\hat{Q}[\omega, 9]]^3 - 3R[\hat{Q}[\omega, 9]]) \end{aligned}$$

By using only in-the-money variables, it makes sure only the reliable paths are used to estimate the model in each time step. The estimated regression is used to evaluate stopping and continuation. Let us take path $\omega=1$ as an example. In that path, $R[\hat{Q}[1, 9]] - I = \$189.183$ M if invested in at $t=9$. If the decision-maker chooses to continue to defer, the expected present value of continuation is $e^{-\rho h} E[\hat{V}[10] | R[\hat{Q}[1, 9]]] = \129.027 M. The optimal decision in that case would be to stop: $D[1, 9]=1$, $\hat{V}[1, 9]=129.027$.

On the other hand, consider path $\omega=4$. $R[\hat{Q}[4, 9]] - I = \$21.801$ M, while $e^{-\rho h} E[\hat{V}[10] | R[\hat{Q}[4, 9]]] = \208.580 M. For that path, the optimal decision is to defer: $D[4, 9]=0$, $\hat{V}[4, 10]=208.580$. These are updated for each sample path and then the algorithm moves down to the next time step.

In the case when an investment involves a bundle of network component improvements, a deferral strategy in optimal timing carries an additional value to redesign the network (Chow and Regan, 2011b). This option has a premium value to redesign because a decision-maker who chooses to defer will have an option in the future to select a different design. An illustration is shown for the classic benchmark Sioux Falls network in Fig. 8.6. When the option to invest in an optimal design under base year state is extended to allow for investments over a 5-year horizon, the value of the option includes both the option to defer (without option to change design) plus the option to redesign.

Another application in a network design context is to consider sequential optimization of fleet deployment. The objective can be setup to have a profitable or selective component (e.g., Feillet et al., 2005). In that case, there is a term in the objective for minimizing travel disutility and another term for minimizing disutility of visiting a node or link. By not visiting a node or link, the risk of a subsequent failure is increased like in inventory problems. The risk can be estimated using LSM. Chow (2016b) implemented an algorithm with this feature to manage UAV fleet deployment to monitor a traffic network where prolonged lack of monitoring can lead to increased costs when an incident occurs.

8.4 OPERATING MODE SWITCHING UNDER UNCERTAINTY

In the prior section, the timing decisions are assumed to be practically irreversible in the sense that doing so would incur significant additional costs. Capital investment projects make good use cases for deferral strategy and sometimes may be combined with other strategies such as expansion, contraction, and abandonment.

Many other strategies, however, can involve switching from one mode to another and back again. Such strategies typically involve data-driven real-time operations. Examples in transport include peak period policies that are switched off when during off-peak periods, turning on and off ramp metering, surge pricing, and activating local bus stops based on demand rates.

Switching is also an optimal timing problem, but more complicated than the stopping problem because it is compounding multiple options (a random number of switches over a given time interval, where the timing of the first one impacts the timing of the subsequent ones) instead of considering only one. The use of switching options is best illustrated with Fig. 8.7 from Guo et al. (2017).

In the figure, a decision-maker chooses to operate either flexible or fixed route last mile transit as a function of ridership demand density. There exists

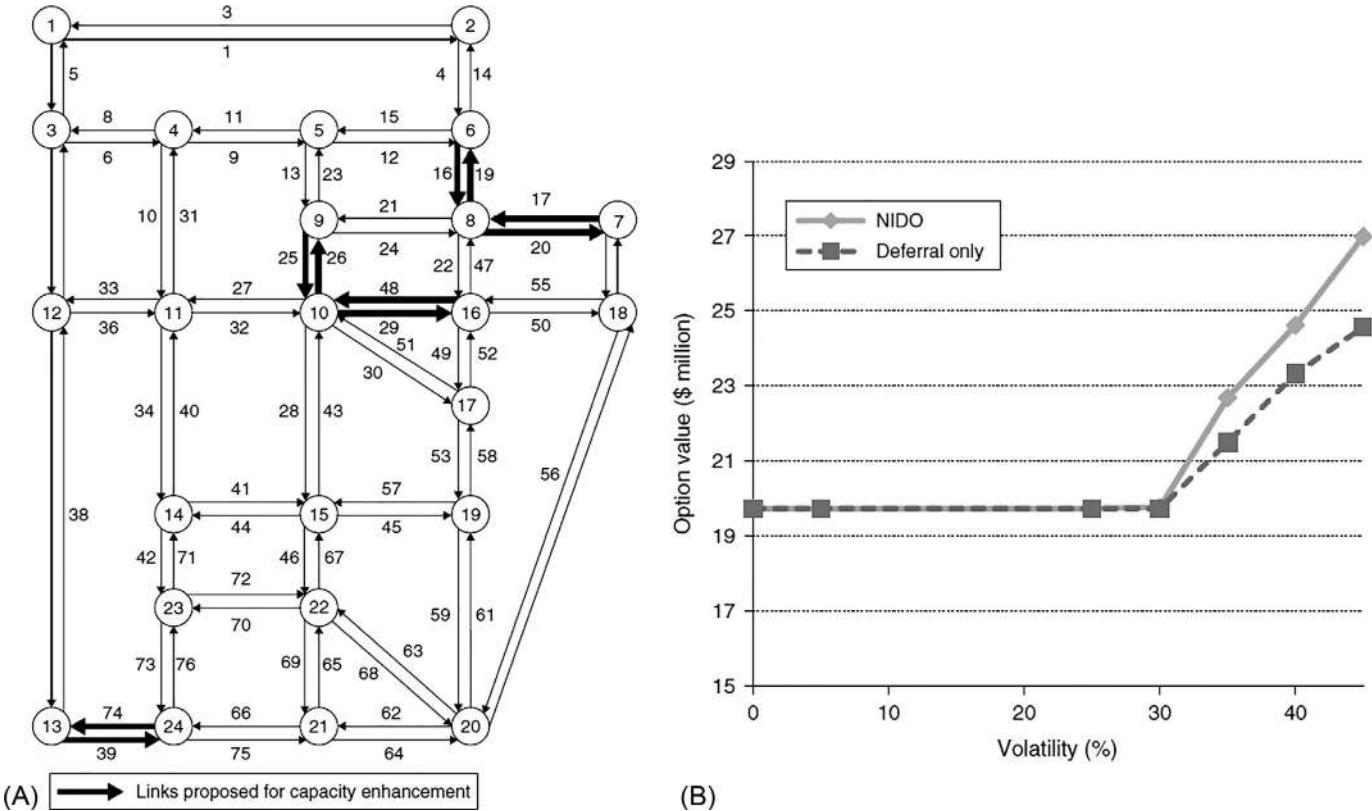


Fig. 8.6 (A) Sioux Falls network with candidate links and (B) comparison of deferral with redesign (Network Investment Deferral Option (NIDO)) to deferral only as function of σ . (Source: Chow and Regan, 2011b.)

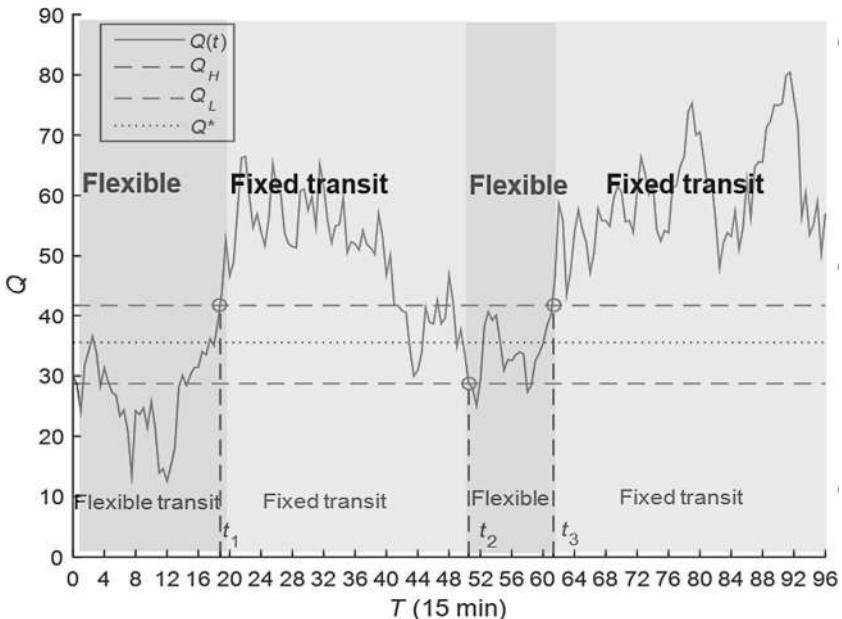


Fig. 8.7 Illustration of market switching. (Source: Guo et al., 2017.)

a demand density threshold over which fixed route service is more cost effective (Chang and Schonfeld, 1991). When there is no switching cost, there is a single threshold marked by the dotted horizontal line corresponding to Q^* . For example, even the first time the price exceeds the line for a moment at $T=4$, it makes sense to switch to fixed route service. However, when there is a switching cost under uncertainty, an additional buffer exists because the volatility of the stochastic process leads to a high risk of having to switch back the first time the line is passed. The buffer shown by the gap between Q_L and Q_H captures that risk. The effect of the uncertainty on this buffer is called hysteresis.

The resulting model provides a decision-maker with analytic thresholds to optimize decisions on when to switch: to go from flexible service to fixed route service when density rises above Q_H and to go from fixed route service to flexible service when density descends below Q_L . The switching option problem determines the values of these two thresholds and the resulting option value of having the flexibility to switch according to them. In the real options literature the problem is also called the *market entry and exit problem*.

In the case where an asset value evolves as a GBM, Dixit (1989) shows a closed-form solution for the market entry-exit problem. In this problem, a decision-maker chooses to switch from an “inactive” mode (call it Mode 0)

to an “active” mode (call it Mode 1) at a switching cost of k and a switching cost of l for vice versa. The parameters μ , σ , ρ are as defined earlier. While in Mode 1, a variable cost w is incurred to produce one unit of output. At $t=0$ the initial price is P_0 . Let $V_0[P]$ be the value function for operating in Mode 0, while $V_1[P]$ be for Mode 1. Using Ito’s lemma on the asset equilibrium condition, where the expected capital gain $E\left[\frac{dV_0[P]}{dt}\right]$ is set equal to the normal return $\rho V_0[P]$, the differential equation for Mode 0 is Eq. (8.26).

$$\frac{1}{2}\sigma^2 P^2 V_0''[P] + \mu P V_0'[P] - \rho V_0[P] = 0 \quad (8.26)$$

The right-hand side is zero because this is assumed to be the “inactive” mode with no operating profit. In the case of a two modes with operating profit, a relative difference can be used to set Mode 0 to be 0. For the Mode 1 case, with the operating profit, the differential equation in Eq. (8.27) includes the term on the right.

$$\frac{1}{2}\sigma^2 P^2 V_1''[P] + \mu P V_1'[P] - \rho V_1[P] = w - P \quad (8.27)$$

Both differential equations are linear and have the same homogeneous part. The solutions become Eq. (8.28), where the parameters β and α in Eq. (8.29) are dependent on $m \equiv \frac{2\mu}{\sigma^2}$ and $r \equiv \frac{2\rho}{\sigma^2}$.

$$V_0[P] = A_0 P^{-\alpha} + B_0 P^\beta \quad (8.28a)$$

$$V_1[P] = A_1 P^{-\alpha} + B_1 P^\beta + \left(\frac{P}{\rho - \mu} - \frac{w}{\rho} \right) \quad (8.28b)$$

$$\beta = \frac{(1-m) + ((1-m)^2 + 4r)^{\frac{1}{2}}}{2} \quad (8.29a)$$

$$-\alpha = \frac{(1-m) - ((1-m)^2 + 4r)^{\frac{1}{2}}}{2} \quad (8.29b)$$

The term $\frac{P}{\rho - \mu} - \frac{w}{\rho}$ is the value of keeping Mode 1 operating indefinitely, which is also equivalent to $E\left[\int_0^\infty (P_t - w)e^{-\rho t} dt\right]$. Using boundary conditions it is possible to set $A_0 = B_1 = 0$ and to assume that $A_1 \geq 0$ and $B_0 \geq 0$. Then the

value matching and smooth pasting conditions are used to obtain four equations in Eq. (8.30) that need to be solved simultaneously to obtain A_1 , B_0 , P_L , P_H , where P_H is the optimal price threshold to switch from Mode 0 to Mode 1 and P_L is the optimal threshold to switch from Mode 1 to Mode 0.

$$A_1 P_L^{-\alpha} + \frac{P_L}{\rho - \mu} - \frac{w}{\rho} = B_0 P_L^\beta - l \quad (8.30a)$$

$$A_1 P_H^{-\alpha} + \frac{P_H}{\rho - \mu} - \frac{w}{\rho} = B_0 P_H^\beta + k \quad (8.30b)$$

$$-A_1 \alpha P_L^{-\alpha-1} + \frac{1}{\rho - \mu} = B_0 \beta P_L^{\beta-1} \quad (8.30c)$$

$$-A_1 \alpha P_H^{-\alpha-1} + \frac{1}{\rho - \mu} = B_0 \beta P_H^{\beta-1} \quad (8.30d)$$

The primary trade-offs in this model is whether to operate in Mode 1 at a cost w when there is a profit that depends on a stochastic process, considering nonzero switching costs. To gain some intuition on the behavior of this model, consider the following transportation application in [Exercise 8.6](#).

Exercise 8.6

Consider a data-driven HOT or truck-toll lane on a highway corridor. Mode 0 is to treat all three lanes in one direction as general-purpose lanes. Mode 1 is to operate the leftmost lane into an HOT lane with a normalized operating cost equivalent to $w=\$1$ per unit time due to reduced general-purpose capacity. If the normalized revenue from traffic, Q , is the underlying asset with $\mu=0$, $\sigma=0.1$, discount rate $\rho=0.15$, and switching costs (in units of vehicles per hour) due to inconvenience of $l=k=\$2$, determine the thresholds and option values of the two modes for an initial value $Q_0=\$0.75$ and time step $h=0.1$. Simulate two trajectories from a base of $\$0.75$ and graphically compare the outcomes using the policy. Conduct sensitivity analysis for σ , ρ , w on the optimal thresholds.

For convenience, we use units of k veh h. The four equations are solved using `fsolve` in MATLAB. Based on the inputs, the results are $Q_L=\$0.585$, $Q_H=\$1.558$, $V_0=\$0.022$, $V_1=-\$1.444$. A comparison of two trajectories is shown in [Fig. 8.8](#).

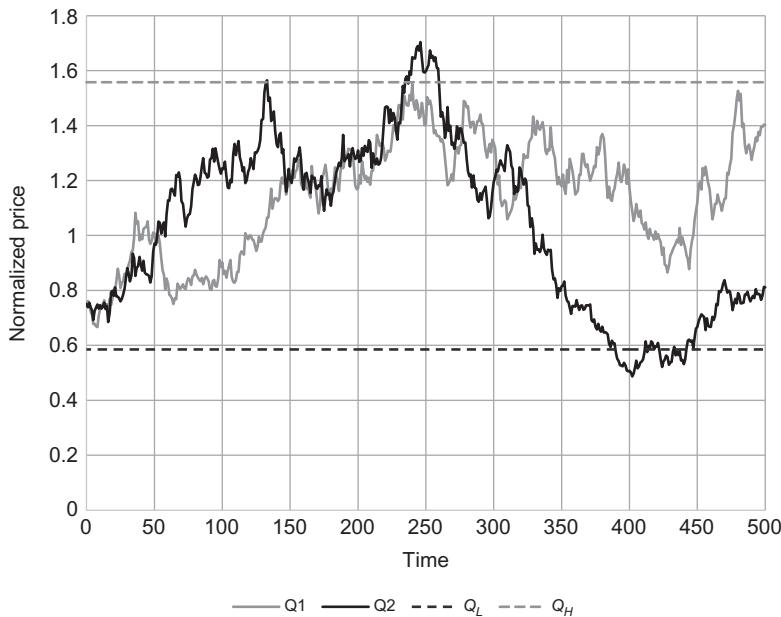


Fig. 8.8 Comparison of two trajectories (Q1, Q2) from an initial point $Q_0 = \$0.75$.

The figure shows how for Q2, the HOT lane is activated when it passes Q_H after 133 time steps and would stay on until it descends below Q_L after 390 time steps. For Q1, however, it never breaks above Q_H over this 500-step interval so in that outcome there would be no switching from all general-purpose lanes. A sensitivity test is conducted for several parameters and summarized in Table 8.3.

Table 8.3 Summary of sensitivity analysis for Exercise 8.6

	Q_L	Q_H	$V_0 [0.750]$	$V_1 [0.750]$
Base	\$0.585	\$1.558	\$0.022	-\$1.444
$\sigma = 0.2$	\$0.500	\$1.844	\$0.193	-\$1.118
$\rho = 0.1$	\$0.648	\$1.487	\$0.096	-\$1.627
$w = 0.8$	\$0.417	\$1.319	\$0.049	-\$0.304

By increasing volatility, the buffer between the lower and upper thresholds increases. When the discount rate decreases, it acts like a decrease in volatility. A reduction of operating cost reduces both thresholds and increases both option values.

The methodology can also be applied to an operational setting. In that case, it may make more sense to use an O-U process (Eq. (8.8)) to describe the underlying asset to capture time of day fluctuations. The drawback of using O-U process, however, is that the $E\left[\int_0^\infty (P_t - w)e^{-\rho t}dt\right]$ term cannot be analytically determined and needs to be numerically computed using finite difference (Guo et al., 2017). On the other hand, it makes it possible to construct thresholds for some interesting operations. Applications of O-U switching options include dry bulk-wet bulk shipping (Sødal et al., 2008) and transit fleet operating mode (Guo et al., 2017).

One of the challenges with using an O-U process is that the general solution to the stochastic processes shown in Eqs. (8.26) and (8.27) involves a confluent hypergeometric function, also called a Kummer function. For example, Eq. (8.31) represents the general solution to $V_0[Q]$ while Eq. (8.32) represents the Kummer function. The asymptotic expression is shown in Sødal et al. (2008) to be equal to Eq. (8.32). The parameter γ_0 is a root in Eq. (8.31b), w_0 is obtained from Eq. (8.31c), and x is obtained from Eq. (8.31d).

$$V_0[Q] = \left[A_0 H[-\gamma_0, w_0, x] + B_0 \left(\frac{2\mu\theta}{\sigma^2 Q} \right)^{1-w_0} H[1 - \gamma_0 - w_0, 2 - w_0, x] \right] Q^{\gamma_0} \quad (8.31a)$$

$$-\gamma^2 + \left(1 + \frac{2\theta}{\sigma^2} \right) \gamma + \frac{2\rho}{\sigma^2} = 0 \quad (8.31b)$$

$$w_0 = 2 - 2\gamma + \frac{2\theta}{\sigma^2} \quad (8.31c)$$

$$x = \frac{2\mu\theta}{\sigma^2 Q} \quad (8.31d)$$

$$H[\gamma, w, x] = 1 + \frac{\gamma}{w} x + \frac{\gamma(\gamma+1)x^2}{w(w+1)2!} + \frac{\gamma(\gamma+1)(\gamma+2)x^3}{w(w+1)(w+2)3!} + \dots \quad (8.32)$$

$$\lim_{x \rightarrow \infty} H[\gamma, w, x] = \frac{\Gamma[w]}{\Gamma[\gamma]} e^x x^{\gamma-w} \quad (8.33)$$

Without going into much more detail (readers are referred to Guo et al., 2017, for specifics), one can then determine optimal thresholds for switching

Table 8.4 Summary of fixed-flexible switching example for Fig. 8.7 in Guo et al. (2017)

	Variables	Values
Fixed transit inputs	Initial demand density $Q[0]$	32 trips/mi ² /h
	Discount rate ρ	7%
	0:1 switching cost k	\$10
	1:0 switching cost l	\$10
	Reversion rate θ	0.2
	Mean μ	40 trips/mi ² /h
	Volatility σ	7 trips/mi ² /h
	Fleet size F_c	5 veh
	Headway h_c	0.42 h
	Vehicle size S_c	75 seats/veh
Flexible transit inputs	Route spacing r	1.41 mi
	Total cost per time step C_{sc}	2881.1
	Fleet size F_f	58 veh
	Headway h_f	0.08 h
	Vehicle size S_f	7 seats/veh
Option valuation	Service zone area A	2.85 mi ²
	Total cost per time step C_{sf}	\$2881.1
	Incremental savings of fixed versus flexible Φ	\$1.9
	Expected cumulative savings $E_t \left[\int_t^{\infty} \Phi(Q) e^{-\rho(s-t)} ds \right]$	-\$39.4
	$V_0[Q[0]]$	\$23.5
	$V_1[Q[0]]$	\$17.4
	Q_L	28.7 trips/mi ² /h
	Q_H	41.8 trips/mi ² /h

between two operating modes. For example, Fig. 8.7 illustrates the switching option for a last mile service between fixed route transit and flexible service based on real-time demand density data. That figure corresponds to variables in Table 8.4.

The parameters for the fixed transit and flexible transit pertain to cost functions derived in Chang and Schonfeld (1991), who showed that under those circumstances there exists a single demand density threshold over which it is better to operate fixed route transit. By optimizing the costs relative to decision variables like fleet size and headway, the incremental costs C_{sc} and C_{sf} can be determined. These are used to determine the difference between costs acting as a savings function for fixed route transit as “Mode 1” (relative to no savings for flexible route as the “Mode 0”). The output thresholds show $Q_L = 28.7 \frac{\text{trips}}{\text{mi}^2 \text{h}}$ and $Q_H = 41.8 \frac{\text{trips}}{\text{mi}^2 \text{h}}$. Since $V_0 > V_1$ when

$Q[0] = 32 \frac{\text{trips}}{\text{mi}^2\text{h}}$, it is better to operate in flexible service mode. If $Q[t]$ eventually rises above $41.8 \frac{\text{trips}}{\text{mi}^2\text{h}}$, it would then make sense to pay the switching cost and switch to fixed transit operating mode. The expected cumulative savings support this decision: operating in fixed transit mode indefinitely would incur a negative cost savings of \$39.4 net present value.

8.5 SEQUENTIAL NETWORK DESIGN AND TIMING

Let us now revisit network design models discussed in [Chapter 7](#). In the context of smart cities, a range of different models and algorithms have been developed using data to improve network design and timing decisions under uncertainty with look-ahead policies and rolling horizons. Examples include dynamic vehicle routing problems (e.g., [Spivey and Powell, 2004](#); [Mitrović-Minić et al., 2004](#); [Thomas and White, 2004](#); [Ichoua et al., 2006](#)), dynamic pricing and routing ([Figliozzi et al., 2007](#); [Sayarshad and Chow, 2015](#)), adaptive discrete network design ([Chow and Regan, 2011c](#)), among others. The greatest opportunity of smart cities lies in such tools to solve sequential network design problems because they allow decision-makers to leverage the Big Data. However, these problems also prove to be the most challenging.

The primary challenge is that the curse of dimensionality is further exacerbated in network settings because of interdependencies between links. In real options terminology, network design is analogous to multioption problems where each network improvement is an option that influences the other candidate options. Clearly there are no closed-form solutions for such problems, and in general they are intractable. Although these problems cannot be solved optimally, [Chow and Regan \(2011c\)](#) proposed a policy that serves as a lower bound to the option value maximizing problem.

In the sequential network design and timing problem, at a current time $t=0$ with remaining budget I , the decision-maker needs to make a set of binary network design decisions Y_t for a set of projects J : $Y_{t,i}=1$ if project $i \in J$ is to be invested in at time t , and $Y_i=0$ if it is to be deferred. The problem assumes a finite time horizon T ; any project that is deferred at $t=T$ is rejected. The performance of a network at time t with design vector Y_t is defined as the expected net present value of the performance, $R[Y_t, Q_t]$, without any other expected changes. If a set of design decisions leads to an annual savings of $(\$1\text{M})e^{0.05t}$, for example, and the discount rate is $\rho=10\%$, then the NPV is $R[Y_t, Q_t] = \frac{\$1\text{M}}{0.10-0.05} = \20M . By combining timing with the network design, the decision-maker may end up seeing their

options all expire by the time horizon, or they may see only a subset that costs less than the budget I invested in.

For the sequential network design and timing problem for a set of network projects J at time t , the set of options can be equivalently represented as a sum of options ordered in a sequence η plus the option of changing the sequence in a future time $s > t$. $l_i[\eta]$ refers to the project index in J corresponding to the i^{th} project in sequence η . For example, if there are three projects such that $J = \{1, 2, 3\}$, and a sequence $\eta = (2, 3, 1)$, then $Y_{l_2[\eta]} = 1$ means that project 3 is invested in.

There exists an optimal sequence η^* of options from which the option value can be expressed in Eq. (8.34). For a given sequence η , the i^{th} project (denoted $i[\eta]$) is only allowed to be invested in if the $(i-1)^{\text{th}}$ project is also invested in. $V[S_t]$ is the expanded NPV, or option value, for a given state at time t . F^D is the premium associated with having the flexibility to defer the i^{th} project. F^L is the premium associated with the flexibility to invest in subsequent projects in the sequence. This premium captures the network interdependencies. F^{LS} is the premium associated with the flexibility to change the sequence of noncommitted projects in the future.

$$V[S_t] = \sum_{i \in J} NPV[i[\eta^*] | S_t] + \sum_{i \in J} (F^D[i[\eta^*] | S_t] + F^L[i[\eta^*] | S_t]) + F^{LS}[\eta^*, S_t] \quad (8.34)$$

Since the overall network design and timing value cannot be solved for, the Chow–Regan (CR) policy is proposed to obtain a lower bound as shown in [Definition 8.4](#).

Definition 8.4 *The CR policy obtains the optimal policy for a lower bound value, denoted $V_{CR}[S_t]$, expressed as Eq. (8.35).*

$$V_{CR}[S_t] = \sum_{i \in J} NPV[i[\eta^*] | S_t] + \sum_{i \in J} (F^D[i[\eta^*] | S_t] + F^L[i[\eta^*] | S_t]) \quad (8.35)$$

The CR policy can be solved by enumerating the sequences and using a multioption LSM algorithm proposed by [Gamba \(2002\)](#) to find the option value for each sequence. The idea of the approach is to constrain the original problem to require sequencing to obtain a solution which is then applied to the original problem. The algorithm from [Chow and Regan \(2011c\)](#) is shown in [Algorithm 8.2](#).

Algorithm 8.2: (Chow and Regan, 2011c). Sequence Enumeration and Multioption LSM to Obtain CR Policy

Inputs: network $G[N, A]$ with demand W ; network design vector Y with investment costs C ; network performance function $R[Y_t, Q_t]$; GBM Q with μ , σ ; initial values $Q[0]$, $R[Q[0]]$; investment cost I ; finite horizon T and number of time step n ; discount rate ρ ; number of radial basis functions m for a chosen basis function L

1. Simulate P sample paths $\hat{Q}[\omega, t]$ from a stochastic process (e.g. Eq. (8.7)).
2. Enumerate the set Θ of budget-feasible sequences; for a given sequence of candidate projects η , only keep the first i projects such that $\sum_{j=1}^i C_{l_j[\eta]} \leq I$.
3. For each sequence $\eta \in \Theta$,
 - a. Compute $\pi_{l_i[\eta]} = R[Y_{t, l_i[\eta]}, \hat{Q}[\omega, t]] - R[Y_{t, l_{i-1}[\eta]}, \hat{Q}[\omega, t]]$ for each i , where $\omega \leq P$, $t \leq n$, and a time step has length $h = T/n$
 - b. For $t = n$
 - i. For each $1 \leq \omega \leq P$,
 1. For i from $|\eta|$ to 1,
 - a. Let $\hat{V}_{l_i}[\omega, n] := \max(\pi_{l_i[\eta]} - \sum_{j=1}^i C_{l_j[\eta]} + \hat{V}_{l_{i+1}}[\omega, n], 0)$
 - ii. Let $t := t - 1$.
 - c. For t from n to 1
 - i. For i from $|\eta|$ to 1,
 1. Construct a weighted sum of basis functions for each sample path where $\hat{V}_{l_i}[\omega, t+1] > 0$ as shown in Eq. (8.36), and estimate \hat{d}_j^t using least squares method.
 2. For each $1 \leq \omega \leq P$
 - a. If $\pi_{l_i[\eta]} - \sum_{j=1}^i C_{l_j[\eta]} + \hat{V}_{l_{i+1}}[\omega, t] > \sum_{j=0}^m \hat{d}_j^t L_j[\pi_{l_i[\eta]}]$
 - i. Update stopping decision: $\hat{D}_{l_i}[\omega, t] := 1$
 - ii. $\hat{V}_{l_i}[\omega, t] := \pi_{l_i[\eta]} - \sum_{j=1}^i C_{l_j[\eta]} + \hat{V}_{l_{i+1}}[\omega, t]$
 - b. Else
 - i. $\hat{D}_{l_i}[\omega, t] := 0 \quad \forall j \geq i$
 - ii. $\hat{V}_{l_i}[\omega, t] := \sum_{j=0}^m \hat{d}_j^t L_j[\pi_{l_i[\eta]}]$
 - ii. Let $t := t - 1$.
 - d. For $t = 0$
 - i. Sequence-conditional option value is determined as in Eq. (8.37).

$$V_{l_i}[Q[0]] := \max \left(\pi_{l_i[\eta]} - \sum_{j=1}^i C_{l_j[\eta]} + \hat{V}_{l_{i+1}}[\omega, t], \frac{1}{P} \sum_{\omega=1}^P \sum_{\tau=1}^n e^{-\rho\tau} \hat{V}_{l_i}[\omega, \tau] \hat{D}_{l_i}[\omega, \tau] \right) \quad (8.37)$$

- ii.** Optimal decision is: $D_{l_i}[0] = 1$ (invest now) if $V_{l_i}[Q[0]] = \pi_{l_i[\eta]} - \sum_{j=1}^i C_{l_j[\eta]} + \hat{V}_{l_{i+1}}[\omega, t]$ else $D_{l_i}[0] = 0, \forall j \geq i$ (defer).
4. Select $\eta^* : \{\text{argmax}_\eta V_{l_i[\eta]}[Q[0]]\}$;
- a. let $V_j[Q[0]] := V_{l_i[\eta^*]}[Q[0]], \forall j \in J, j = l_i[\eta^*]$
 - b. let $D_j[Q[0]] := D_{l_i[\eta^*]}[Q[0]], \forall j \in J, j = l_i[\eta^*]$

Outputs: $V_j[Q[0]], D_j[0], \forall j \in J$ (note the sequence is not kept—only the final option values and decisions).

Proof of convergence to CR policy follows from [Gamba \(2002\)](#) and [Longstaff and Schwartz \(2001\)](#) and detailed in [Chow and Sayarshad \(2016\)](#). The algorithm is demonstrated in [Exercise 8.7](#).

Exercise 8.7

Consider a single OD demand from node O to node D shown in [Fig. 8.9](#), with three parallel links with different link performance functions, where y_i is a binary variable indicating whether a link is improved:

$$c_1 = 30 + \frac{x_1}{100 + 50y_1}$$

$$c_2 = 40 + \frac{x_2}{200 + 100y_2}$$

$$c_3 = 50 + \frac{x_3}{400 + 200y_3}$$

The graph in [Fig. 8.9](#) suggests there are different flows where one link is preferred over the others. The network agency has a budget to support up to two link investments. The network performance (since we are interested in relative comparison we can leave out the conversion from peak hour traffic to annual savings) is measured as:

$$R[Q] = \frac{(TTT_0[Q] - TTT_y[Q])}{\rho - \mu}$$

where TTT_0 is the total system travel time without any investments, TTT_y is the total system travel time with network improvements, and $\rho = 0.10$. Initial OD

demand is $Q[0] = 11,000$ and evolves as a GBM with $\mu = 0$, $\sigma = 0.2$. The agency has a 2-year time horizon with the option to decide only at the end of each year (total of three points in time where decisions can be made), on which link to invest, after which the budget expires. Use Algorithm 8.2 (with $P=20$ sample paths, $T=2$, $n=2$, and $m=3$ with Hermite polynomials) to suggest a decision to be made at time 0 and justify.

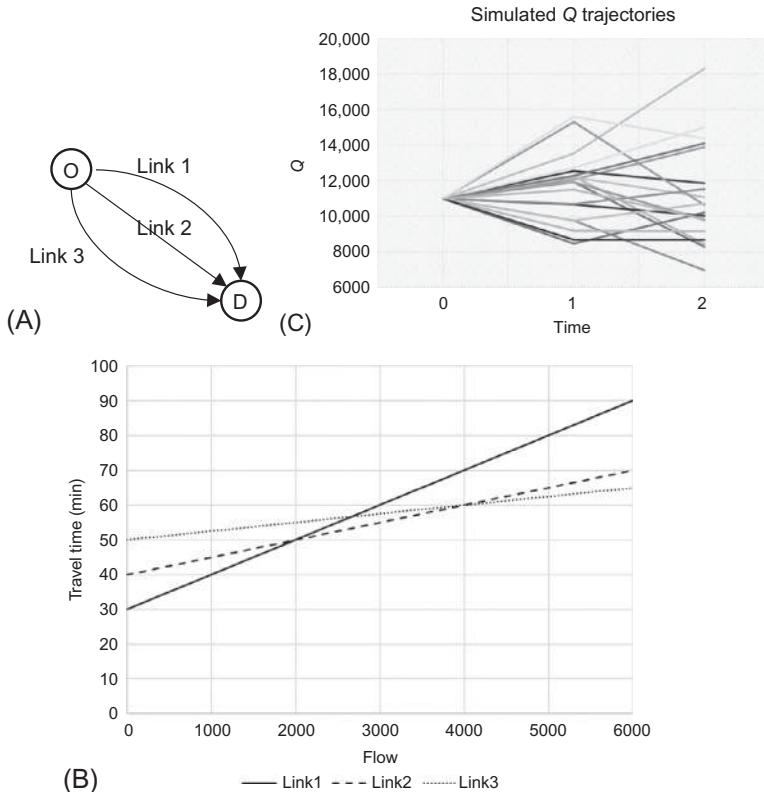


Fig. 8.9 (A) Single OD, three-link example with (B) a plot of the base link cost functions and (C) simulated trajectories of Q over a 2-year horizon.

After simulating 2 time steps with 20 paths we get the following sample paths for $\hat{Q}[p, t]$ along with the computed equilibrium $TTT_0[p, t]$ in Table 8.5.

With a budget of two projects, there are six possible project sequences: $(1, 2)$, $(2, 1)$, $(1, 3)$, $(3, 1)$, $(2, 3)$, $(3, 2)$. The steps for $(1, 2)$ are broken down here for illustrative purposes.

First calculate $R[Y_{0,l_1}, 11,000] = 220K$, $R[Y_{0,l_2}, 11,000] = 453K$. This means building only link 1 improvement would net a present value

savings of 220K, whereas adding link 2 would increase that amount to 453K. The marginal value of link 1 project is $\pi_{l_1}[0]=220$ K while for link 2 project it is $\pi_{l_2}[0]=233$ K. This is done for both time steps. At the final time step, the option value for project 2 is simply $\pi_{l_2}[p, 2]$ while for project 1 it is $\pi_{l_1}[p, 1] + V_{l_2}[p, 2]$, as presented in [Table 8.6](#).

Table 8.5 Simulation of 20 sample paths

ω	$Q[\omega, 1]$	$Q[\omega, 2]$	$TTT0[\omega, 1]$	$TTT0[\omega, 2]$
1	11,918.22	8310.446	730,727	466,696.2
2	13,523.6	18,277.64	860,170.8	1,286,684
3	12,070.06	13,881.24	742,654.8	890,010.7
4	12,534.71	11,867.43	779,564.5	726,752
5	11,916.52	9855.489	730,593.8	575,215.4
6	12,259.11	8460.147	757,597.4	476,912.4
7	9771.734	6978.685	569,157.9	378,630.4
8	8484.843	10,216.52	478,603.9	601,556.3
9	12,180.21	11,074.52	751,348.2	665,650.3
10	15,291.5	10,627.79	1,011,238	632,016.2
11	10,666.04	10,042.1	634,873.6	588,784.3
12	12,714.3	14,993.95	793,995.6	985,187.5
13	9207.324	9160.194	528,859.8	525,536
14	10,703.08	11,537.56	637,644.5	701,114.1
15	12,282.82	14,098.43	759,478.5	908,310.5
16	9766.49	10,722.35	568,779.3	639,088.3
17	11,914.06	9786.104	730,401.2	570,195.8
18	8687.566	8680.444	492,554.8	492,062.7
19	15,602.63	14,370.21	1,038,748	931,399.6
20	11,484.21	9866.02	696,996.5	575,978.5

For time step 1, we use kernel regression with Hermite polynomials: $\{1, x, x^2 - 1, x^3 - 3x\}$. Based on these independent variables where $x = \pi_{l_1}[p, 1]$ and the dependent variables are $e^{-\rho}E[V_{l_2}[p, 2] | \pi_{l_1}[p, 1]]$. For example, with project 2 we get:

$$\begin{aligned} e^{-\rho}E[V_{l_2}[p, 2] | \pi_{l_2}[p, 1]] &= 1801.957 \\ &\quad - 20.849(\pi_{l_2}[p, 1]) + 0.081(\pi_{l_2}[p, 1]^2 - 1) \\ &\quad - (9.5 \times 10^{-5})(\pi_{l_2}[p, 1]^3 - 3\pi_{l_2}[p, 1]) \end{aligned}$$

This continuation value, $e^{-\rho}E[V_{l_2}[p, 2] | \pi_{l_2}[p, 1]]$, is compared against the value of exercising. For project 2 the value of exercising in each sample path is $\pi_{l_2}[p, 1]$, whereas for project 1 it is equal to $\pi_{l_1}[p, 1] + V_{l_2}[p, 1]$. If $\pi_{l_1}[p, 1] + V_{l_2}[p, 1] > e^{-\rho}E[V_{l_2}[p, 2] | \pi_{l_2}[p, 1]]$, then the option is exercised at time and path. The option values, their continuation values, and the decision values are presented in [Table 8.7](#).

Table 8.6 Exercise values for each option in sequence

ω	$\pi_{I_1}[p, 1]$	$\pi_{I_2}[p, 1]$	$\pi_{I_1}[p, 2]$	$\pi_{I_2}[p, 2]$	$V_{I_1}[p, 2]$	$V_{I_2}[p, 2]$
1	248,786.9	269,552.3	144,921.9	140,924.9	285,846.8	140,924.9
2	302,975.1	339,916.6	492,236.6	595,711.6	1,087,948	595,711.6
3	253,701.9	275,861.3	315,715.6	356,693.4	672,409	356,693.4
4	269,015.4	295,617	247,152.7	267,458.1	514,610.8	267,458.1
5	248,732.1	269,482	186,367.2	191,010.8	377,378	191,010.8
6	259,882.7	283,817.3	148,738.6	145,450.1	294,188.8	145,450.1
7	184,003.9	188,103.7	112,846.6	103,762.8	216,609.3	103,762.8
8	149,372.4	146,203.4	196,707.1	203,793.9	400,501	203,793.9
9	257,294.7	280,483	222,276.4	235,813.8	458,090.3	235,813.8
10	368,328.7	426,759	208,788.3	218,853.9	427,642.2	218,853.9
11	209,928.3	220,281.5	191,680.8	197,567.2	389,248	197,567.2
12	275,044.3	303,434.2	356,912.6	411,456.6	768,369.2	411,456.6
13	168,426.7	169,087.3	167,153.4	167,544.5	334,698	167,544.5
14	211,034.8	221,668.2	236,658.1	254,053.8	490,711.9	254,053.8
15	260,662.6	284,823	323,571.6	367,077.6	690,649.2	367,077.6
16	183,856.4	187,922.4	211,611.8	222,391.7	434,003.5	222,391.7
17	248,652.9	269,380.5	184,408.4	188,600.9	373,009.4	188,600.9
18	154,618.5	152,459.2	154,432.9	152,237.2	306,670.1	152,237.2
19	380,446.2	443,056.9	333,528.8	380,280.2	713,809	380,280.2
20	234,980.1	251,917.8	186,665.3	191,377.9	378,043.1	191,377.9

Table 8.7 Exercise decisions

ω	$D_{I_1}[p, 1]$	$D_{I_2}[p, 1]$	$D_{I_1}[p, 2]$	$D_{I_2}[p, 2]$
1	1	1	0	0
2	0	0	1	1
3	1	1	0	0
4	1	1	0	0
5	1	1	0	0
6	1	1	0	0
7	1	1	0	0
8	0	0	1	1
9	1	1	0	0
10	1	1	0	0
11	1	1	0	0
12	1	0	0	1
13	1	1	0	0
14	1	1	0	0
15	1	1	0	0
16	1	1	0	0
17	1	1	0	0
18	0	0	1	1
19	1	1	0	0
20	1	1	0	0

At present time, the value of exercising project 2 is 233K while deferring would have an expected value of 252K. Therefore the option value of project 2 is 252K. Project 1's exercise value is then equal to the marginal value (220K) plus the option value of project 2, resulting in 472K. This is less than the value of deferring project 1 altogether (477K). Based on the volatility, it is better to defer project 1 and project 2 as it nets a value of 477K. The other sequences are evaluated similarly and all the exercise, deferral, option values and decisions are summarized in [Table 8.8](#).

The sequence with maximum project 1 value (which captures the options of subsequent projects) is (2,3) with a decision to defer both projects. The option to time and invest in any two links in this network is therefore at least as high as 498K. At a future year, all sequences that consider feasible combinations of all three links would be evaluated again as the current sequencing is not actually committed to. If it turned out to be (1,3) instead, then the CR policy decision would have been to invest in link 1 and defer the rest to evaluate between links 2 and 3 in the future.

Table 8.8 Summary of exercise, deferral, option values, and decisions for each sequence in whole

	Second project exercise	Second project deferral	Project 2 option	First project exercise	First project deferral	Project 1 option	Decision
(1,2)	232,941	252,055	252,055	472,055	476,525	476,525	Defer 1,2
(1,3)	185,263	222,901	222,901	442,901	434,036	442,901	Exercise 1, defer 3
(2,1)	177,941	185,101	185,101	460,101	481,604	481,604	Defer 2, 1
(2,3)	165,000	198,521	198,521	473,521	497,837	497,837	Defer 2, 3
(3,1)	160,819	164,557	164,557	409,002	450,396	450,396	Defer 3, 1
(3,2)	195,556	206,351	206,351	450,796	492,190	492,190	Defer 3, 2
Final			198,521			497,837	Defer 2, 3

Table 8.9 Summary of CR policy decisions as a function of σ

Volatility σ	Timing/design decisions	Lower bound option value (\$M)
0.05	Exercise all	19.61
0.25	Exercise 1, 5; defer 2, 3, 4	19.95
0.30	Exercise 1, 5; defer 2,3, 4	20.80
0.35	Exercise 1, 5; defer 2,3, 4	21.56
0.40	Defer all	24.71

The CR policy is applied to the Sioux Falls network in [Chow and Regan \(2011c\)](#) with a horizon of 5 years to choose among the five pairs of bidirectional links to invest in. In that example, there are 552 OD pairs (r, s) evolving as independent GBMs with $\mu_{rs} = 0$ and uniform $\sigma_{rs} \equiv \sigma$. A budget of \$5.5M is used where each project has its own cost. A discount rate of $\rho = 0.06$ is used. For [Algorithm 8.2](#), there were $P = 500$ sample paths, $n = T = 5$, and Hermite polynomials were used for the kernel regression with $m = 5$. The results are presented in [Table 8.9](#).

One major drawback of [Algorithm 8.2](#) is that it relies on sequence enumeration which limits valuation to only a small numbers of budget-feasible projects. For five projects, there are $5! = 120$ permutations, whereas for 10 projects there are $10! = 3,628,800$ permutations.

One way to overcome this drawback was proposed by [Chow and Sayarshad \(2016\)](#). The concept is that the option value is based on selecting the highest option value among all sequences Θ . If only a sample of options are independently selected at random, it is possible to approximate the maximum by fitting an extreme value distribution—a Weibull distribution—as shown in Eq. (8.38). The parameter φ is for location of the distribution and δ is for scaling.

$$F_W[y; \varphi, \delta] = 1 - \exp\left(-\left(\frac{y}{\varphi}\right)^\delta\right), \quad y > 0 \quad (8.38)$$

If the distribution can be fit to the samples, then the distribution of the maximum of the population is known to be Weibull distributed as well ([Gumbel, 1958](#)), as shown as $F_{MW}[y; \varphi, \delta]$ in Eq. (8.39). Then if a sample S of sequences are used to obtain option values $V_{s,s} \in S$, the distribution provides a *probabilistic measure* of where the CR policy option value lies.

$$F_{MW}[y; \varphi, \delta] = \left(1 - \exp\left(-\left(\frac{y}{\varphi}\right)^\delta\right)\right)^{|\Theta|}, \quad y > 0 \quad (8.39)$$

One can use maximum likelihood estimation (see [Balakrishnan and Kateri, 2008](#)) to estimate the parameters by solving first for $\hat{\delta}$ in Eq. (8.40a) and then for $\hat{\varphi}$ in Eq. (8.40b).

$$\frac{|S|}{\hat{\delta}} - \frac{|S|}{\sum_{i=1}^{|S|} \gamma_i^{\hat{\delta}}} \sum_{j=1}^{|S|} \gamma_j^{\hat{\delta}} \ln \gamma_j + \sum_{j=1}^{|S|} \ln \gamma_j = 0 \quad (8.40a)$$

$$\hat{\varphi} = \left(\frac{1}{|S|} \sum_{j=1}^{|S|} \gamma_j^{\hat{\delta}} \right)^{\frac{1}{\hat{\delta}}} \quad (8.40b)$$

Although with this approach one does not know the decisions under the CR policy, one at least can measure how close the sampled sequence option values are to the CR policy value. In turn, the cumulative distribution of the CR policy value can be cheaply evaluated for different sequential NDPs to easily compare methodologies as a “reference policy.” Compared to other measures such as competitive ratios (Karp, 1992), myopic policies, and perfect hindsight measures, the CR policy is sensitive to the rate at which information is revealed over time (information sensitivity) and can capture value from flexibility (network effect measurable). A myopic policy or perfect hindsight policy, on the other hand, cannot distinguish between one network instance in which information significantly helps anticipative look-ahead versus another instance in which it does not. The result is that comparing performances of algorithms against those insensitive reference policies cannot be extrapolated to other instances, whereas the CR policy does.

The CR policy Weibull distribution is illustrated in [Exercise 8.8](#) to show how it can be used to compare algorithms.

Exercise 8.8

A dynamic routing algorithm operated in City A earns an operator a present value of 1030. A competitor algorithm operated in City B earns that operator a present value of 3700. Randomly sampled route sequences constructed to serve demand data obtained in City A and City B are used to estimate parameters for Weibull distributions: $\hat{\delta}_A = 200$, $\hat{\varphi}_A = 1000$, $|\Theta_A| = 10000$, $\hat{\delta}_B = 30$, $\hat{\varphi}_B = 3500$, $|\Theta_B| = 300$. How well is Algorithm A to perform compared to Algorithm B if they were to be operated in the same city?

This is a very typical question asked in smart cities settings: which algorithm is more effective? Comparison based on performances in two different settings is difficult to be done. For example, it seems operator B is netting a higher profit than operator A, but is this because it is more effective or because of the structure of the demand in city B is more supportive of the algorithm? By using the CR policy as a third-party reference policy it is possible to rank the two algorithms’ performance

relative to a third-party reference: in terms of percentile on the CR-Weibull distribution.

We compute the percentiles that Algorithm A and Algorithm B perform relative to the CR policy Weibull distribution. Based on the demand data, sampled sequences, and determining the option values from timing those sequences, the parameters for the CR-Weibull distributions are shown for the two cities and plotted in Fig. 8.10. The figure suggests that City B's network naturally supports a higher value from the dynamic design and timing decisions. This can be due to the structure, information propagation, and demand patterns in City B.

$$F_{MW,A}[y; \varphi, \delta] = \left(1 - \exp\left(-\left(\frac{y}{1000}\right)^{200}\right)\right)^{10,000}$$

$$F_{MW,B}[y; \varphi, \delta] = \left(1 - \exp\left(-\left(\frac{y}{3500}\right)^{30}\right)\right)^{300}$$

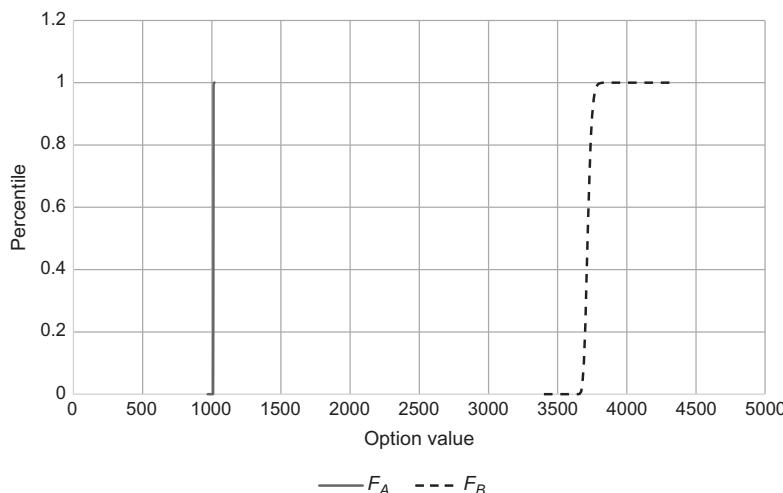


Fig. 8.10 Comparison of the CR-Weibull distributions for City A (F_A) and City B (F_B).

The CR-Weibull distribution provides a reference policy to compare the two algorithms. For example, Algorithm A operates on the 100 percentile of the CR-Weibull distribution (at $\frac{1030-1000}{1000} = 3\%$ above the 50 percentile). Algorithm A is able to exploit the premium value of the flexibility to adapt the design (sequence) to new information, which is missing in the CR policy. Meanwhile, Algorithm B operates on the 22.2 percentile of the CR policy. A head-to-head comparison in the same city is likely to see Algorithm A outperforming Algorithm B.

By using a reference policy we can draw this conclusion without having to setup an expensive field experiment comparing the two directly.

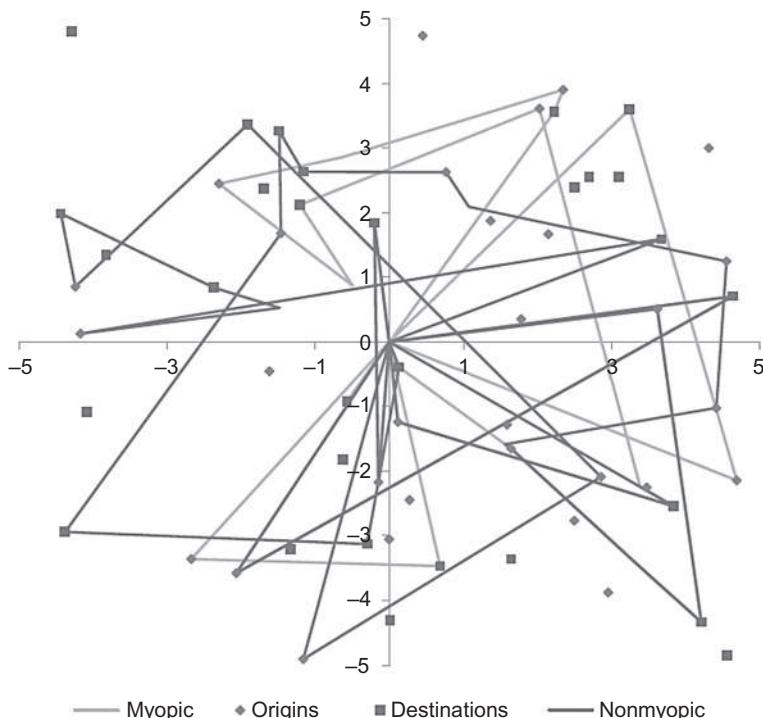


Fig. 8.11 Simulation of Hyytiä et al. (2012) dynamic dispatch policy. (Source: Chow and Sayarshad, 2016.)

The reference policy is further applied by Chow and Sayarshad (2016) to evaluate the Hyytiä et al. (2012) queueing-based policy along with a myopic policy. A random instance is simulated as illustrated in Fig. 8.11 where the policy is applied to route and dispatch a fleet of vehicles to pick passengers up. Three scenarios are simulated. The first one is the base scenario. The second one considers double the demand rate. The third considers half the vehicle capacity serving the base scenario demand. The option value performance is measured as cost (and temporarily offset to be positive to estimate the Weibull parameters). The results are summarized in Fig. 8.12.

Several interesting conclusions can be drawn. In the three cases tested, the nonmyopic policy from Hyytiä et al. (2012) performs about the same as the myopic policy. This may be because of the lack of adequate parameter tuning necessary to improvement performance, as shown in Ma et al. (2018). The values are at least better than the purely randomly generated sample sequence solutions, which should make sense. As demand is increased from Scenario A to Scenario B, we see the CR-Weibull distribution increasing in

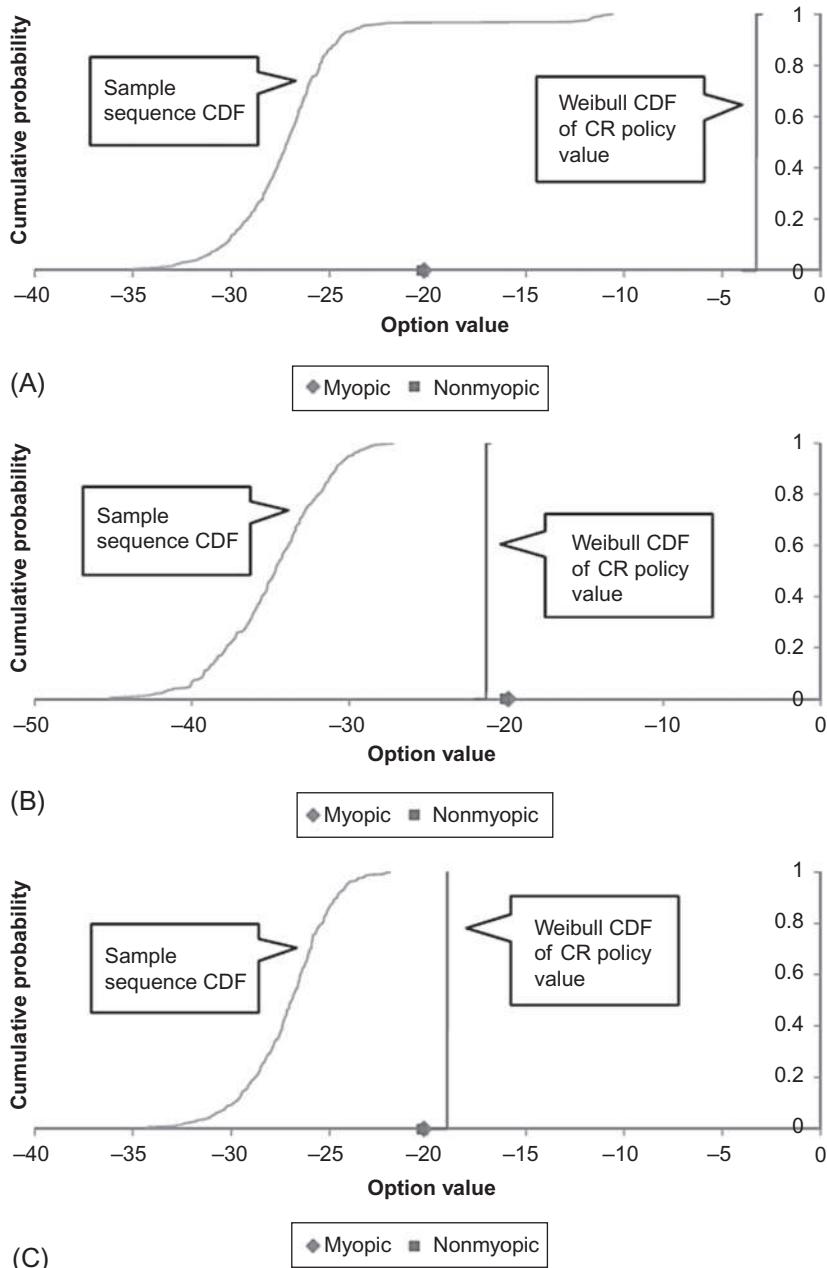


Fig. 8.12 Evaluation of myopic and nonmyopic policies relative to CR-Weibull distributions. (A) Scenario A, (B) Scenario B, and (C) Scenario C. (Source: [Chow and Sayarshad, 2016](#).)

cost which suggests there is an increase in network effects on vehicle dispatch and routing. This makes sense since the increase demand puts a greater strain on the available capacity. The demand doubling places less of a strain than halving the capacity, however, as that increases the CR-Weibull distribution of the cost further up. Although the performance of the Hyytiä policy appears to be roughly the same cost throughout all three scenarios, a different conclusion can be drawn when viewed relative to the reference policy. In scenario A, the Hyytiä policy significantly underperforms the CR policy. In scenario B, however, it improves over the CR policy. This suggests that the increased demand allows the Hyytiä policy to flourish more and gain option value from the flexibility to adjust routing plans.

Sequential NDP is a young and exciting field due to its many applications in urban mobility and potential enhancements from artificial intelligence and optimal learning. For example, the prior material in this chapter all assumes that data is freely made available to an operator. However, in the case of a setting with very limited information and highly autonomous systems (e.g., shared autonomous vehicles—see [Mahmassani, 2016](#)), vehicles act both as servers and sensors. In that case, navigating them over a network is not just a decision to optimize service to users but it is also a decision to position sensors to learn about the evolution of the network. There remains very little research in this area of combining network optimization models with optimal learning.

Traditional learning models that consider these trade-offs in resource allocation for exploitation and exploration include multiarmed bandit problems (e.g., [Li et al., 2010](#)). These models assume that the underlying distribution is unknown and requires resources to sample the options over multiple trials. Each sample also acts to serve the user or decision-maker as they earn a reward. Too much exploration can lead to a poor rate of rewards. These models generally use a measure of regret to compare multiarmed bandit algorithms. A drawback of the approaches is that they do not consider nonstationary stochastic processes very well and may be hampered by highly contextual trials.

Another area is the dynamic route navigation to optimally learn the network states. One example is [Ryzhov and Powell \(2011\)](#). These algorithms use Bayesian inferencing to rank and select routes to traverse over time. It will not be long before we see self-adapting bus routes that expand over a city to both serve users and to learn the ridership demand efficiently over time. By then perhaps truly smart cities will not be so far beyond our reach.

RESEARCH AND DESIGN CHALLENGES

The following are open-ended challenges that sometimes refer to real data sets. Readers are encouraged to try to tackle the problems, and post their designs or solutions on the following GitHub page, <https://github.com/BUILTNYU/Elsevierbook/tree/master/Chapter%208>.

- (8.1) Propose a dynamic pricing plan for a local city on-street parking that assumes the number of parking spaces is monitored every hour. Discuss how you would model the demand each hour and what data needs to be collected.
- (8.2) Fit a GBM to a stock of your choice. Forecast the probability of the stock exceeding 20% of its current price in the next 7 days, 30 days, and 1 year.
- (8.3) Fit an O-U model to the daily average weather temperature in your local city and another one for a city 50–100 miles nearby. Determine the correlation between the daily average weathers between the two cities. Simulate 500 sample paths of the 7-day forecast joint temperatures between the two cities using Cholesky decomposition (see [Appendix D](#)) and compare the simulations of the correlated variables to the independent moments defined in Eq. (8.10).
- (8.4) Use real option theory to compare option value of investing in a heavy rail system, a light rail system, or a bus system for a local city, assuming an underlying GBM for ridership demand. How does this comparison change if it was based on NPV instead?
- (8.5) For the Sioux Falls network example in [Chow and Regan \(2011b\)](#), determine the risk-neutral rate μ for an equivalent project with valuation $R = R_0 e^{\mu t}$ where R_0 is the current annual value of investment and R is the projected annual value in year t . If the option value of such a project was evaluated under an infinite horizon, how does that compare to the finite horizon option value in the study?
- (8.6) Verify the point made at the end of [Exercise 8.4](#) using Monte Carlo simulation.
- (8.7) Complete [Exercise 8.5](#) to determine the option value and optimal policy timing the toll road investment.
- (8.8) For the example in Challenge 8.4, consider having two of the technologies as options. Determine the threshold ridership demand values to switch between a lower capacity service to the higher capacity service and vice versa.

- (8.9) Take a look at the list of project sections for the California High Speed Rail:

<https://www.buildhsr.com/maps/>.

Use the CR policy to stage the sections based on intercity demand data that you can find. State any assumptions you make. Evaluate the quality of your policy using simulation.

- (8.10) As discussed in [Chapter 1](#), car sharing was not very successful in San Diego but was profitable in Seattle. Using travel demand data that you can obtain from the two cities, assume the same car-sharing mode choice model for each city specified toward travel costs, distance of trips, and access distance to nearest car-sharing location. Sample random sequences of zones where car-sharing stations would be placed and solve for the optimal timing based on each fixed sequence. Use these sequenced values to estimate the CDF of the CR policy for each city. If car-sharing works well in Seattle and not in San Diego, the hypothesis is that the CR policy value in San Diego is much worse than in Seattle. Can this be verified?

References

- Abdulaal, M., LeBlanc, L.J., 1979. Continuous equilibrium network design models. *Transp. Res. B Methodol.* 13 (1), 19–32.
- Adler, N., Pels, E., Nash, C., 2010. High-speed rail and air transport competition: game engineering as tool for cost-benefit analysis. *Transp. Res. B Methodol.* 44 (7), 812–833.
- Adomavicius, G., Tuzhilin, A., 2005. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.* 17 (6), 734–749.
- Agarwal, R., Ergun, Ö., 2008. Mechanism design for a multicommodity flow game in service network alliances. *Oper. Res. Lett.* 36 (5), 520–524.
- Ahuja, R.K., Orlin, J.B., 2001. Inverse optimization. *Oper. Res.* 49 (5), 771–783.
- Ahuja, R.K., Magnanti, T.L., Orlin, J.B., 1993. *Network Flows: Theory, Algorithms, and Applications*. Pearson, Upper Saddle River.
- Aïvodji, U.M., Gambs, S., Huguet, M.J., Killijian, M.O., 2016. Meeting points in ridesharing: a privacy-preserving approach. *Transp. Res. C* 72, 239–253.
- Akamatsu, T., 1996. Cyclic flows, Markov process and stochastic traffic assignment. *Transp. Res. B* 30 (5), 369–386.
- Albino, V., Berardi, U., Dangelico, R.M., 2015. Smart cities: definitions, dimensions, performance, and initiatives. *J. Urban Technol.* 22 (1), 3–21.
- Alexander, L., Jiang, S., Murga, M., González, M.C., 2015. Origin–destination trips by purpose and time of day inferred from mobile phone data. *Transp. Res. C* 58, 240–250.
- Allahviranloo, M., Recker, W., 2013. Daily activity pattern recognition by using support vector machines with multiple classes. *Transp. Res. B* 58, 16–43.
- Allahviranloo, M., Chow, J.Y.J., Recker, W.W., 2014. Selective vehicle routing problems under uncertainty without recourse. *Transp. Res. E* 62, 68–88.
- Ambler, S.W., 2017. Agile modeling: effective practices for modeling and documentation. <http://agilemodeling.com/>. (Accessed 10 March 2018).
- Amer, A., Chow, J.Y.J., 2017. A downtown on-street parking model with urban truck delivery behavior. *Transp. Res. A* 102, 51–67.
- Anderson, S.P., De Palma, A., 2004. The economics of pricing parking. *J. Urban Econ.* 55 (1), 1–20.
- Angel, S., Hyman, G.M., 1970. Urban velocity fields. *Environ. Plan. A* 2 (2), 211–224.
- Arentze, T.A., Timmermans, H.J.P., 2004. A multi-state supernetwork approach to modeling multiactivity multi-modal trip chains. *Int. J. Geogr. Inf. Sci.* 18 (7), 631–651.
- Arentze, T.A., Timmermans, H.J., 2009. A need-based model of multi-day, multi-person activity generation. *Transp. Res. B* 43 (2), 251–265.
- Arnott, R., 2006. Spatial competition between parking garages and downtown parking policy. *Transp. Policy* 13 (6), 458–469.
- Arnott, R., Inci, E., 2006. An integrated model of downtown parking and traffic congestion. *J. Urban Econ.* 60 (3), 418–442.
- Arnott, R., Rowse, J., 1999. Modeling parking. *J. Urban Econ.* 45 (1), 97–124.
- Arnott, R., Rowse, J., 2009. Downtown parking in auto city. *Reg. Sci. Urban Econ.* 39 (1), 1–14.
- Arnott, R., Rowse, J., 2013. Curbside parking time limits. *Transp. Res. A* 55, 89–110.
- Arnott, R., De Palma, A., Lindsey, R., 1988. Schedule delay and departure time decisions with heterogeneous commuters. *Transp. Res. Rec.* 1197, 56–67.
- Arnott, R., De Palma, A., Lindsey, R., 1990a. Economics of a bottleneck. *J. Urban Econ.* 27 (1), 111–130.

- Arnott, R., De Palma, A., Lindsey, R., 1990b. Departure time and route choice for the morning commute. *Transp. Res.* B 24 (3), 209–228.
- Arnott, R., De Palma, A., Lindsey, R., 1991. A temporal and spatial equilibrium analysis of commuter parking. *J. Public Econ.* 45 (3), 301–335.
- Arnott, R., De Palma, A., Lindsey, R., 1993. A structural model of peak-period congestion: a traffic bottleneck with elastic demand. *Am. Econ. Rev.* 83 (1), 161–179.
- Arrow, K.J., Harris, T., Marschak, J., 1951. Optimal inventory policy. *Econometrica* 19 (3), 250–272.
- Ashok, K., Ben-Akiva, M.E., 2002. Estimation and prediction of time-dependent origin-destination flows with a stochastic mapping to path flows and link flows. *Transp. Sci.* 36 (2), 184–198.
- Aswani, A., Shen, Z.J.M., Siddiq, A., 2015. Inverse optimization with noisy data. arXiv preprint arXiv: 1507.03266.
- Auld, J., Hope, M., Ley, H., Sokolov, V., Xu, B., Zhang, K., 2016. POLARIS: agent-based modeling framework development and implementation for integrated travel demand and network and operations simulations. *Transp. Res.* C 64, 101–116.
- Axhausen, K.W., Gärling, T., 1992. Activity-based approaches to travel analysis: conceptual frameworks, models and research problems. *Transp. Rev.* 12, 324–341.
- Bahill, A.T., Gissing, B., 1998. Re-evaluating systems engineering concepts using systems thinking. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* 28 (4), 516–527.
- Baker, B.M., Aye chew, M.A., 2003. A genetic algorithm for the vehicle routing problem. *Comput. Oper. Res.* 30 (5), 787–800.
- Balac, M., Ciari, F., Axhausen, K.W., 2017. Modeling the impact of parking price policy on free-floating carsharing: case study for Zurich, Switzerland. *Transp. Res. C Emerg. Technol.* 77, 207–225.
- Balakrishnan, N., Kateri, M., 2008. On the maximum likelihood estimation of parameters of Weibull distribution based on complete and censored data. *Stat. Probab. Lett.* 78 (17), 2971–2975.
- Balinski, M.L., Quandt, R.E., 1964. On an integer program for a delivery problem. *Oper. Res.* 12 (2), 300–304.
- Bansal, A., Chen, T., Zhong, S., 2011. Privacy preserving back-propagation neural network learning over arbitrarily partitioned data. *Neural Comput. Appl.* 20 (1), 143–150.
- Bard, J.F., 1991. Some properties of the bilevel programming problem. *J. Optim. Theory Appl.* 68 (2), 371–378.
- Bard, J.F., 2013. Practical Bilevel Optimization: Algorithms and Applications. vol. 30. Springer Science & Business Media, Boston.
- Bard, J.F., Moore, J.T., 1990. A branch and bound algorithm for the bilevel programming problem. *SIAM J. Sci. Stat. Comput.* 11 (2), 281–292.
- Bar-Gera, H., 2002. Origin-based algorithm for the traffic assignment problem. *Transp. Sci.* 36 (4), 398–417.
- Batty, M., Axhausen, K.W., Giannotti, F., Pozdnoukhov, A., Bazzani, A., Wachowicz, M., Ouzounis, G., Portugali, Y., 2012. Smart cities of the future. *Eur. Phys. J. Spec. Top.* 214 (1), 481–518.
- Becker, G.S., 1965. A theory of the allocation of time. *Econ. J.* 75 (299), 493–517.
- Beckmann, M., McGuire, C.B., Winsten, C.B., 1956. Studies in the Economics of Transportation. Yale University Press, New Haven, CT. also published as Rand-RM-1488-PR, Rand Corporation, Santa Monica, CA, May 12, 1955.
- Bell, M.G., 1995. Stochastic user equilibrium assignment in networks with queues. *Transp. Res.* B 29 (2), 125–137.
- Bellemands, T., Kochan, B., Janssens, D., Wets, G., Arentze, T., Timmermans, H.J.P., 2010. Implementation framework and development trajectory of FEATHERS activity-based simulation platform. *Transp. Res. Rec.* 2175, 111–119.
- Bellman, R., 1957. Dynamic Programming. Princeton University Press, New Jersey.

- Ben-Akiva, M.E., Lerman, S.R., 1985. Discrete Choice Analysis: Theory and Application to Travel Demand. MIT Press, Cambridge.
- Ben-Elia, E., Shiftan, Y., 2010. Which road do I take? A learning-based model of route-choice behavior with real-time information. *Transp. Res. A* 44 (4), 249–264.
- Berman, O., Odoni, A.R., 1982. Locating mobile servers on a network with Markovian properties. *Networks* 12 (1), 73–86.
- Berman, O., Larson, R.C., Chiu, S.S., 1985. Optimal server location on a network operating as an M/G/1 queue. *Oper. Res.* 33 (4), 746–771.
- Berman, O., Larson, R.C., Fouska, N., 1992. Optimal location of discretionary service facilities. *Transp. Sci.* 26 (3), 201–211.
- Bertsimas, D., Gupta, V., Paschalidis, I.C., 2012. Inverse optimization: a new perspective on the Black-Litterman model. *Oper. Res.* 60 (6), 1389–1403.
- Bertsimas, D., Gupta, V., Paschalidis, I.C., 2015. Data-driven estimation in equilibrium using inverse optimization. *Math. Program.* 153 (2), 595–633.
- Bettencourt, L.M., 2014. The uses of big data in cities. *Big Data* 2 (1), 12–22.
- Bettencourt, L.M., Lobo, J., Helbing, D., Kühnert, C., West, G.B., 2007. Growth, innovation, scaling, and the pace of life in cities. *Proc. Natl. Acad. Sci.* 104 (17), 7301–7306.
- Bhat, C.R., Koppelman, F.S., 1999. Activity-based modeling of travel demand. In: Hall, R.W. (Ed.), *The Handbook of Transportation Science*. Kluwer Academic Publishers, Norwell, MA, pp. 35–61.
- Bhat, C.R., Guo, J.Y., Srinivasan, S., Sivakumar, A., 2004. A comprehensive econometric microsimulator for daily activity-travel patterns. *Transp. Res. Rec.* 1894, 57–66.
- Bierlaire, M., Lurkin, V., 2017. Introduction to Disaggregate Demand Models (No. EPFL-CHAPTER-232996). Institute for Operations Research and the Management Sciences (INFORMS), Lausanne, pp. 48–67.
- Bird, C.G., 1976. On cost allocation for a spanning tree: a game theoretic approach. *Networks* 6 (4), 335–350.
- Birge, J.R., Hortaçsu, A., Pavlin, J.M., 2017. Inverse optimization for the recovery of market structure from market outcomes: an application to the miso electricity market. *Oper. Res.* 65 (4), 837–855.
- Bischoff, J., Maciejewski, M., 2016. Simulation of city-wide replacement of private cars with autonomous taxis in Berlin. *Procedia Comput. Sci.* 83, 237–244.
- Bishop, C.M., 2006. *Pattern Recognition and Machine Learning*. Springer, New York.
- Blanchard, B.S., Fabrycky, W.J., Fabrycky, W.J., 1998. *Systems Engineering and Analysis*, third ed. Prentice Hall, Upper Saddle River.
- Blum, A., Dwork, C., McSherry, F., Nissim, K., 2005. In: *Practical privacy: the SuLQ framework*. Proceedings of the Twenty-Fourth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, June. ACM, pp. 128–138.
- Boerkamps, J., van Binsbergen, A., Bovy, P., 2000. Modeling behavioral aspects of urban freight movement in supply chains. *Transp. Res. Rec.* 1725, 17–25.
- Boesch, P.M., Ciari, F., Axhausen, K.W., 2016. Autonomous vehicle fleet sizes required to serve different levels of demand. *Transp. Res. Rec.* 2542, 111–119.
- Borndörfer, R., Grötschel, M., Pfetsch, M.E., 2007. A column-generation approach to line planning in public transport. *Transp. Sci.* 41 (1), 123–132.
- Bowman, J., Ben-Akiva, M.E., 2001. Activity-based disaggregate travel demand model system with activity schedules. *Transp. Res. A* 35, 1–28.
- Boyce, D., Zhang, Y.F., 1997. Calibrating combined model of trip distribution, modal split, and traffic assignment. *Transp. Res. Rec.* 1607, 1–5.
- Boyce, D.E., Janson, B.N., Eash, R.W., 1981. The effect on equilibrium trip assignment of different link congestion functions. *Transp. Res. A* 15 (3), 223–232.
- Boyce, D.E., Mahmassani, H.S., Nagurney, A., 2005. A retrospective on Beckmann, McGuire and Winsten's studies in the economics of transportation. *Pap. Reg. Sci.* 84 (1), 85–103.

- Boyle, P.P., 1977. Options: a Monte Carlo approach. *J. Financ. Econ.* 4 (3), 323–338.
- Bradley, S., Hax, A., Magnanti, T., 1977. *Applied Mathematical Programming*. Addison-Wesley, Reading.
- Brennan, M.J., Schwartz, E.S., 1977. The valuation of American put options. *J. Financ.* 32 (2), 449–462.
- Brownlee, J., 2013. A tour of machine learning algorithms. <https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/>. (Accessed 12 November 2017).
- Brucker, P., Shaklevich, N.V., 2009. Inverse scheduling with maximum lateness objective. *J. Sched.* 12 (5), 475–488.
- Bruss, F.T., 1984. A unified approach to a class of best choice problems with an unknown number of options. *Ann. Probab.*, 882–889.
- Buchholz, N., 2015. Spatial Equilibrium, Search Frictions and Efficient Regulation in the Taxi Industry. Technical Report. University of Texas at Austin.
- Burkard, R.E., Pleschiutschnig, C., Zhang, J., 2004. Inverse median problems. *Discret. Optim.* 1 (1), 23–39.
- Burton, D., Toint, P.L., 1992. On an instance of the inverse shortest paths problem. *Math. Program.* 53 (1), 45–61.
- Cai, P., Wang, Y., Lu, G., Chen, P., Ding, C., Sun, J., 2016. A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting. *Transp. Res. C* 62, 21–34.
- Cairns, R.D., Liston-Heyes, C., 1996. Competition and regulation in the taxi industry. *J. Public Econ.* 59 (1), 1–15.
- Calthrop, E., Proost, S., 2006. Regulating on-street parking. *Reg. Sci. Urban Econ.* 36 (1), 29–48.
- Candler, W., Townsley, R., 1982. A linear two-level programming problem. *Comput. Oper. Res.* 9 (1), 59–76.
- Cantarella, G.E., Cascetta, E., 1995. Dynamic processes and equilibrium in transportation networks: towards a unifying theory. *Transp. Sci.* 29 (4), 305–329.
- Caragliu, A., Del Bo, C., Nijkamp, P., 2011. Smart cities in Europe. *J. Urban Technol.* 18 (2), 65–82.
- Cardinal, J., Demaine, E.D., Fiorini, S., Joret, G., Langerman, S., Newman, I., Weimann, O., 2011. The Stackelberg minimum spanning tree game. *Algorithmica* 59 (2), 129–144.
- Cascetta, E., 2009. *Transportation Systems Analysis: Models and Applications*. Springer Science & Business Media, Boston.
- Cascetta, E., Nuzzolo, A., Russo, F., Vitetta, A., 1996. In: A modified logit route choice model overcoming path overlapping problems. Specification and some calibration results for interurban networks. Proc. 13th ISTTT, Lyon, France.
- Ceder, A., Wilson, N.H., 1986. Bus network design. *Transp. Res. B Methodol.* 20 (4), 331–344.
- Cepeda, M., Cominetti, R., Florian, M., 2006. A frequency-based assignment model for congested transit networks with strict capacity constraints: characterization and computation of equilibria. *Transp. Res. B* 40 (6), 437–459.
- Chan, T.C., Lee, T., 2017. Trade-off preservation in inverse multi-objective convex optimization. arXiv preprint arXiv: 1706.06926.
- Chan, T.C., Craig, T., Lee, T., Sharpe, M.B., 2014. Generalized inverse multiobjective optimization with application to cancer therapy. *Oper. Res.* 62 (3), 680–695.
- Chang, S.K., Schonfeld, P.M., 1991. Optimization models for comparing conventional and subscription bus feeder services. *Transp. Sci.* 25 (4), 281–298.
- Chen, A., Yang, C., 2004. Stochastic transportation network design problem with spatial equity constraint. *Transp. Res.* 1882, 97–104.
- Chen, T., Zhong, S., 2009. Privacy-preserving backpropagation neural network learning. *IEEE Trans. Neural Netw.* 20 (10), 1554–1564.

- Chen, J., Shaw, S.L., Yu, H., Lu, F., Chai, Y., Jia, Q., 2011. Exploratory data analysis of activity diary data: a space–time GIS approach. *J. Transp. Geogr.* 19 (3), 394–404.
- Chen, R., Fung, B., Desai, B.C., Sossou, N.M., 2012. Differentially private transit data publication: a case study on the Montreal transportation system. *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 213–221.
- Chen, X., Zhu, Z., He, X., Zhang, L., 2015. Surrogate-based optimization for solving a mixed integer network design problem. *Transp. Res. Rec.* 2497, 124–134.
- Chin, A., Lai, A., Chow, J.Y.J., 2016. Nonadditive public transit fare pricing under congestion with policy lessons from a case study in Toronto, Ontario, Canada. *Transp. Res. Rec.* 2544, 28–37.
- Chow, J.Y.J., 2014. Activity-based travel scenario analysis with routing problem reoptimization. *Comput. Aided Civ. Infrastruct. Eng.* 29 (2), 91–106.
- Chow, J.Y.J., 2016a. Transportation research h-index rankings by institution, 2007–2016. LinkedIn Pulse. <https://www.linkedin.com/pulse/transportation-research-h-index-rankings-institution-2007-2016-chow>. (Accessed 14 January 2017).
- Chow, J.Y.J., 2016b. Dynamic UAV-based traffic monitoring under uncertainty as a stochastic arc-inventory routing policy. *Int. J. Transp. Sci. Technol.* 5 (3), 167–185.
- Chow, J.Y.J., Djavadian, S., 2015. Activity-based market equilibrium for capacitated multimodal transport systems. *Transp. Res. C* 59, 2–18.
- Chow, J.Y.J., Liu, H., 2012. Generalized profitable tour problems for online activity routing system. *Transp. Res. Rec.* 2284, 1–9.
- Chow, J.Y.J., Nurumbetova, A.E., 2015. A multi-day activity-based inventory routing model with space–time–needs constraints. *Transportmetrica A* 11 (3), 243–269.
- Chow, J.Y.J., Recker, W.W., 2012. Inverse optimization with endogenous arrival time constraints to calibrate the household activity pattern problem. *Transp. Res. B* 46 (3), 463–479.
- Chow, J.Y.J., Regan, A.C., 2011a. Resource location and relocation models with rolling horizon forecasting for wildland fire planning. *INFOR: Inf. Syst. Oper. Res.* 49 (1), 31–43.
- Chow, J.Y.J., Regan, A.C., 2011b. Real option pricing of network design investments. *Transp. Sci.* 45 (1), 50–63.
- Chow, J.Y.J., Regan, A.C., 2011c. Network-based real option models. *Transp. Res. B Methodol.* 45 (4), 682–695.
- Chow, J.Y.J., Regan, A.C., 2014. A surrogate-based multiobjective metaheuristic and network degradation simulation model for robust toll pricing. *Optim. Eng.* 15 (1), 137–165.
- Chow, J.Y.J., Sayarshad, H.R., 2014. Symbiotic network design strategies in the presence of coexisting transportation networks. *Transp. Res. B* 62, 13–34.
- Chow, J.Y.J., Sayarshad, H.R., 2016. Reference policies for non-myopic sequential network design and timing problems. *Netw. Spatial Econ.* 16 (4), 1183–1209.
- Chow, J.Y.J., Regan, A.C., Arkhipov, D.I., 2010a. Faster converging global heuristic for continuous network design using radial basis functions. *Transp. Res. Rec.* 2196, 102–110.
- Chow, J.Y.J., Yang, C.H., Regan, A.C., 2010b. State-of-the art of freight forecast modeling: lessons learned and the road ahead. *Transportation* 37 (6), 1011–1030.
- Chow, J.Y.J., Regan, A.C., Ranaiefar, F., Arkhipov, D.I., 2011. A network option portfolio management framework for adaptive transportation planning. *Transp. Res. A Policy Pract.* 45 (8), 765–778.
- Chow, J.Y.J., Jayakrishnan, R., Mahmassani, H.S., 2013. Is transport modeling education too multidisciplinary? A manifesto on the search for its evolving identity. In: Miller, E.J., Roorda, M.J. (Eds.), *Travel Behaviour Research: Current Foundations, Future Prospect*. Lulu Publishing, Toronto.

- Chow, J.Y.J., Ritchie, S.G., Jeong, K., 2014. Nonlinear inverse optimization for parameter estimation of commodity-vehicle-decoupled freight assignment. *Transp. Res.* E 67, 71–91.
- Chriqui, C., Robillard, P., 1975. Common bus lines. *Transp. Sci.* 9 (2), 115–121.
- Christofides, N., 1976. Worst-Case Analysis of a New Heuristic for the Travelling Salesman Problem (No. RR-388). Carnegie-Mellon Univ Pittsburgh PA Management Sciences Research Group.
- Church, R., ReVelle, C.S., 1974. The maximal covering location problem. *Pap. Reg. Sci.* 32 (1), 101–118.
- Clark, S., Watling, D., 2000. Probit-based sensitivity analysis for general traffic networks. *Transp. Res. Rec.* 1733, 88–95.
- Clarke, G., Wright, J.W., 1964. Scheduling of vehicles from a central depot to a number of delivery points. *Oper. Res.* 12 (4), 568–581.
- Çolak, S., Alexander, L.P., Alvim, B.G., Mehndiratta, S.R., González, M.C., 2015. Analyzing cell phone location data for urban travel: current methods, limitations, and opportunities. *Transp. Res. Rec.* 2526, 126–135.
- Comi, A., Delle Site, P., Filippi, F., Nuzzolo, A., 2012. Urban freight transport demand modelling: a state of the art. *Eur. Transp.* (51).
- Cominetti, R., Correa, J., 2001. Common-lines and passenger assignment in congested transit networks. *Transp. Sci.* 35 (3), 250–267.
- Cooper, R.G., Edgett, S.J., Kleinschmidt, E.J., 1998. Portfolio Management for New Products. Addison Wesley Longman, Inc., Reading, MA.
- Cordeau, J.F., Laporte, G., 2003. The dial-a-ride problem (DARP): variants, modeling issues and algorithms. *4OR* 1 (2), 89–101.
- Cordeau, J.F., Laporte, G., 2007. The dial-a-ride problem: models and algorithms. *Ann. Oper. Res.* 153 (1), 29–46.
- Cortés, C.E., Matamala, M., Contardo, C., 2010. The pickup and delivery problem with transfers: formulation and a branch-and-cut solution method. *Eur. J. Oper. Res.* 200 (3), 711–724.
- Cox, J.C., Ross, S.A., Rubinstein, M., 1979. Option pricing: a simplified approach. *J. Financ. Econ.* 7 (3), 229–263.
- Craianic, T.G., Ricciardi, N., Storchi, G., 2004. Advanced freight transportation systems for congested urban areas. *Transp. Res. C* 12 (2), 119–137.
- Croissant, Y., 2012. Estimation of multinomial logit models in R: the mlogit packages. R package version 0.2-2.
- Current, J., Marsh, M., 1993. Multiobjective transportation network design and routing problems: taxonomy and annotation. *Eur. J. Oper. Res.* 65 (1), 4–19.
- Current, J.R., Schilling, D.A., 1989. The covering salesman problem. *Transp. Sci.* 23 (3), 208–213.
- Current, J., ReVelle, C.S., Cohon, J.L., 1985. The maximum covering/shortest path problem: a multiobjective network design and routing formulation. *Eur. J. Oper. Res.* 21 (2), 189–199.
- CUSP, 2017. Center for Urban Science & Progress: Research. <http://cusp.nyu.edu/research/> (Accessed 14 January 2017).
- Dafermos, S., 1980. Traffic equilibrium and variational inequalities. *Transp. Sci.* 14 (1), 42–54.
- Dafermos, S., 1982. Relaxation algorithms for the general asymmetric traffic equilibrium problem. *Transp. Sci.* 16 (2), 231–240.
- Dafermos, S., Nagurney, A., 1984. Sensitivity analysis for the asymmetric network equilibrium problem. *Math. Program.* 28 (2), 174–184.
- Dafermos, S.C., Sparrow, F.T., 1969. The traffic assignment problem for a general network. *J. Res. Natl. Bur. Stand. B* 73 (2), 91–118.

- Daganzo, C.F., 1985. The uniqueness of a time-dependent equilibrium distribution of arrivals at a single bottleneck. *Transp. Sci.* 19 (1), 29–37.
- Daganzo, C.F., Sheffi, Y., 1977. On stochastic models of traffic assignment. *Transp. Sci.* 11 (3), 253–274.
- Daganzo, C.F., Bouthelier, F., Sheffi, Y., 1977. Multinomial probit and qualitative choice: a computationally efficient algorithm. *Transp. Sci.* 11 (4), 338–358.
- Dantzig, G.B., 1951. Application of the simplex method to a transportation problem. *Act. Anal. Prod. Alloc.* 13, 359–373.
- Dantzig, G.B., Ramser, J.H., 1959. The truck dispatching problem. *Manag. Sci.* 6 (1), 80–91.
- Dantzig, G.B., Thapa, M.N., 2003. *Linear Programming 2: Theory and Extensions*. Springer, New York.
- Daskin, M.S., 1983. A maximum expected covering location model: formulation, properties and heuristic solution. *Transp. Sci.* 17 (1), 48–70.
- Day, J., Nemhauser, G.L., Sokol, J.S., 2002. Management of railroad impedances for shortest path-based routing. *Electron. Notes Theor. Comput. Sci.* 66 (6), 53–65.
- De Borger, B., Fosgerau, M., 2012. Information provision by regulated public transport companies. *Transp. Res. B* 46 (4), 492–510.
- De Cea, J., Fernández, E., 1993. Transit assignment for congested public transport systems: an equilibrium model. *Transp. Sci.* 27 (2), 133–147.
- De Palma, A., Ginsburgh, V., Labbé, M., Thisse, J.F., 1989. Competitive location with random utilities. *Transp. Sci.* 23 (4), 244–252.
- Desaulniers, G., Hickman, M.D., 2007. Public transit. In: *Handbooks in Operations Research and Management Science*, vol. 14. Elsevier, North Holland, pp. 69–127.
- Desrosiers, J., Soumis, F., Desrochers, M., 1984. Routing with time windows by column generation. *Networks* 14 (4), 545–565.
- Dia, H., 2002. An agent-based approach to modelling driver route choice behaviour under the influence of real-time information. *Transp. Res. C* 10 (5), 331–349.
- Dial, R.B., 1971. A probabilistic multipath traffic assignment model which obviates path enumeration. *Transp. Res.* 5 (2), 83–111.
- Dijkstra, E.W., 1959. A note on two problems in connexion with graphs. *Numer. Math.* 1 (1), 269–271.
- Dixit, A., 1989. Entry and exit decisions under uncertainty. *J. Polit. Econ.* 97 (3), 620–638.
- Dixit, A.K., Pindyck, R.S., 1994. *Investment Under Uncertainty*. Princeton University Press, Princeton.
- Djavadian, S., Chow, J.Y.J., 2017a. An agent-based day-to-day adjustment process for modeling ‘Mobility as a Service’ with a two-sided flexible transport market. *Transp. Res. B* 104, 36–57.
- Djavadian, S., Chow, J.Y.J., 2017b. Agent-based day-to-day adjustment process to evaluate dynamic flexible transport service policies. *Transportmetrica B* 5 (3), 286–311.
- Dogrue, L., Joeckel, L., Vitak, J., 2017. The valuation of privacy premium features for smartphone apps: the influence of defaults and expert recommendations. *Comput. Hum. Behav.* 77, 230–239.
- Domencich, T.A., McFadden, D., 1975. *Urban Travel Demand—A Behavioral Analysis*. North-Holland Publishing Company Ltd., North Holland.
- Dong, R., Krichene, W., Bayen, A.M., Sastry, S.S., 2015. In: *Differential privacy of populations in routing games*. 2015 IEEE 54th Annual Conference on Decision and Control (CDC). IEEE.
- Donovan, B., Work, D.B., 2017. Empirically quantifying city-scale transportation system resilience to extreme events. *Transp. Res. C: Emer. Technol.* 79, 333–346.
- Douglas, A.E., 2010. *The Symbiotic Habit*. Princeton University Press, Princeton.
- Dror, M., Trudeau, P., 1989. Savings by split delivery routing. *Transp. Sci.* 23 (2), 141–145.

- Dror, M., Ball, M., Golden, B., 1985. A computational comparison of algorithms for the inventory routing problem. *Ann. Oper. Res.* 4 (1), 1–23.
- Dumas, Y., Desrosiers, J., Soumis, F., 1991. The pickup and delivery problem with time windows. *Eur. J. Oper. Res.* 54 (1), 7–22.
- Dwork, C., 2008. In: Differential privacy: a survey of results. International Conference on Theory and Applications of Models of Computation. Springer, Berlin, Heidelberg, pp. 1–19.
- Dwork, C., Nissim, K., 2004. In: Privacy-preserving datamining on vertically partitioned databases. CRYPTO, Augustvol. 3152. , pp. 528–544.
- Dwork, C., McSherry, F., Nissim, K., Smith, A., 2006. Calibrating noise to sensitivity in private data analysis. *TCC*. vol. 3876, pp. 265–284.
- Erlang, A.K., 1909. The theory of probabilities and telephone conversations. *Nyt Tidsskrift for Matematik* B 20 (6), 87–98.
- Esfahani, P.M., Shafeezadeh-Abadeh, S., Hanusanto, G.A., Kuhn, D., 2015. Data-driven inverse optimization with incomplete information. arXiv preprint arXiv: 1512.05489.
- Ettema, D., Borgers, A.W.J., Timmermans, H.J.P., 1993. Simulation model of activity scheduling behavior. *Transp. Res. Rec.* 1413, 1–11.
- Even, S., Itai, A., Shamir, A., 1975. In: On the complexity of time table and multi-commodity flow problems. 16th Annual Symposium on Foundations of Computer Science, pp. 184–193.
- Farahani, R.Z., Miandoabchi, E., Szeto, W.Y., Rashidi, H., 2013. A review of urban transportation network design problems. *Eur. J. Oper. Res.* 229 (2), 281–302.
- Feillet, D., Dejax, P., Gendreau, M., 2005. Traveling salesman problems with profits. *Transp. Sci.* 39 (2), 188–205.
- Feinberg, Y., 2000. Characterizing common priors in the form of posteriors. *J. Econ. Theory* 91 (2), 127–179.
- Fernandez, E., Marcotte, P., 1992. Operators-users equilibrium model in a partially regulated transit system. *Transp. Sci.* 26 (2), 93–105.
- Figliozzi, M.A., Mahmassani, H.S., Jaillet, P., 2007. Pricing in dynamic vehicle routing problems. *Transp. Sci.* 41 (3), 302–318.
- Fisher, M.L., Jaikumar, R., 1981. A generalized assignment heuristic for vehicle routing. *Networks* 11 (2), 109–124.
- Fisk, C., 1980. Some developments in equilibrium traffic assignment. *Transp. Res. B* 14 (3), 243–255.
- Florian, M., Gaudry, M., 1980. A conceptual framework for the supply side in transportation systems. *Transp. Res. B* 14 (1), 1–8.
- Florian, M., Nguyen, S., 1978. A combined trip distribution modal split and trip assignment model. *Transp. Res.* 12 (4), 241–246.
- Florian, M., Nguyen, S., Ferland, J., 1975. On the combined distribution-assignment of traffic. *Transp. Sci.* 9 (1), 43–53.
- Fosgerau, M., De Palma, A., 2013. The dynamics of urban traffic congestion and the price of parking. *J. Public Econ.* 105, 106–115.
- Fosgerau, M., Frejinger, E., Karlstrom, A., 2013. A link based network route choice model with unrestricted choice set. *Transp. Res. B* 56, 70–80.
- Frank, M., Wolfe, P., 1956. An algorithm for quadratic programming. *Nav. Res. Logist.* 3 (1–2), 95–110.
- Frejinger, E., Bierlaire, M., 2007. Capturing correlation with subnetworks in route choice models. *Transp. Res. B* 41 (3), 363–378.
- Friesz, T.L., Cho, H.J., Mehta, N.J., Tobin, R.L., Anandalingam, G., 1992. A simulated annealing approach to the network design problem with variational inequality constraints. *Transp. Sci.* 26 (1), 18–26.
- Friesz, T.L., Bernstein, D., Mehta, N.J., Tobin, R.L., Ganjalizadeh, S., 1994. Day-to-day dynamic network disequilibria and idealized traveler information systems. *Oper. Res.* 42 (6), 1120–1136.

- Friesz, T.L., Bernstein, D., Stough, R., 1996. Dynamic systems, variational inequalities and control theoretic models for predicting time-varying urban network flows. *Transp. Sci.* 30 (1), 14–31.
- Furuhashi, M., Daniel, K., Koenig, S., Ordóñez, F., Dessouky, M., Brunet, M.E., Cohen, L., Wang, X., 2015. Online cost-sharing mechanism design for demand-responsive transport systems. *IEEE Trans. Intell. Transp. Syst.* 16 (2), 692–707.
- Galera, A.L.L., Soliño, A.S., 2010. A real options approach for the valuation of highway concessions. *Transp. Sci.* 44 (3), 416–427.
- Gamba, A., 2002. In: An extension of least squares Monte Carlo simulation for multi-options problems. Proceedings of the Sixth Annual International Real Options Conference, Paphos, Cyprus, July, p. 41.
- Gan, L.P., Recker, W., 2008. A mathematical programming formulation of the household activity rescheduling problem. *Transp. Res. B* 42 (6), 571–606.
- Gan, L.P., Recker, W., 2013. Stochastic preplanned household activity pattern problem with uncertain activity participation (SHAPP). *Transp. Sci.* 47 (3), 439–454.
- Gandomi, A., Haider, M., 2015. Beyond the hype: big data concepts, methods, and analytics. *Int. J. Inf. Manag.* 35 (2), 137–144.
- Gao, S., 2012. Modeling strategic route choice and real-time information impacts in stochastic and time-dependent networks. *IEEE Trans. Intell. Transp. Syst.* 13 (3), 1298–1311.
- Gao, Z., Wu, J., Sun, H., 2005. Solution algorithm for the bi-level discrete network design problem. *Transp. Res. B Methodol.* 39 (6), 479–495.
- Gao, Z., Sun, H., Zhang, H., 2007. A globally convergent algorithm for transportation continuous network design problem. *Optim. Eng.* 8 (3), 241–257.
- Garcia, D., 2003. Convergence and biases of Monte Carlo estimates of American option prices using a parametric exercise rule. *J. Econ. Dyn. Control.* 27 (10), 1855–1879.
- Garey, M.R., Johnson, D.S., 1977. The rectilinear Steiner tree problem is NP-complete. *SIAM J. Appl. Math.* 32 (4), 826–834.
- Gärling, T., Kwan, M.P., Golledge, R.G., 1994. Computational-process modeling of household travel activity scheduling. *Transp. Res. B* 25, 355–364.
- Garrison, D., 2016. Car2Go ceases San Diego operations. The San Diego Union-Tribune 2016. December 31.
- Garvin, M.J., Cheah, C.Y., 2004. Valuation techniques for infrastructure investment decisions. *Constr. Manag. Econ.* 22 (4), 373–383.
- Geoffrion, A., Brüderl, R.M., 1978. Lagrangean relaxation applied to capacitated facility location problems. *AIEE Trans.* 10 (1), 40–47.
- Geweke, J., 1989. Bayesian inference in econometric models using Monte Carlo integration. *Econometrica*, 1317–1339.
- Ghosh, A., Craig, C.S., 1983. Formulating retail location strategy in a changing environment. *J. Mark.*, 56–68.
- Gibbons, R., 1992. Game Theory for Applied Economists. Princeton University Press, Princeton.
- Glazer, A., Niskanen, E., 1992. Parking fees and congestion. *Reg. Sci. Urban Econ.* 22 (1), 123–132.
- Golden, B.L., Wong, R.T., 1981. Capacitated arc routing problems. *Networks* 11 (3), 305–315.
- Golledge, R.G., Kwan, M.P., Gärling, T., 1994. Computational process modeling of household travel decisions using a geographical information system. *Pap. Reg. Sci.* 73 (2), 99–117.
- Golob, T.F., 2000. A simultaneous model of household activity participation and trip chain generation. *Transp. Res. B* 34 (5), 355–376.
- Gomory, R.E., 1958. Outline of an algorithm for integer solutions to linear programs. *Bull. Am. Math. Soc.* 64, 275–278.

- Gonzalez, M.C., Hidalgo, C.A., Barabasi, A.L., 2008. Understanding individual human mobility patterns. *Nature* 453 (7196), 779–782.
- Goodchild, M.F., 2013. Prospects for a space–time GIS: space–time integration in geography and GIScience. *Ann. Assoc. Am. Geogr.* 103 (5), 1072–1077.
- Granot, D., Granot, F., 1992. On some network flow games. *Math. Oper. Res.* 17 (4), 792–841.
- Grinstead, C.M., Snell, J.L., 2012. *Introduction to Probability*, second ed. American Mathematical Society, Providence.
- Guan, J.F., Yang, H., Wirasinghe, S.C., 2006. Simultaneous optimization of transit line configuration and passenger line assignment. *Transp. Res. B* 40 (10), 885–902.
- Guelat, J., Florian, M., Crainic, T.G., 1990. A multimode multiproduct network assignment model for strategic planning of freight flows. *Transp. Sci.* 24 (1), 25–39.
- Guilford, D., 2016. BMW's DriveNow is profitable now. *Automotive news*, 2016. October 3.
- Güler, Ç., Hamacher, H.W., 2010. Capacity inverse minimum cost flow problem. *J. Comb. Optim.* 19 (1), 43–59.
- Gumbel, E.J., 1958. *Statistics of Extremes*. Columbia University Press, republication by Dover, NY.
- Gümüş, Z.H., Floudas, C.A., 2005. Global optimization of mixed-integer bilevel programming problems. *Comput. Manag. Sci.* 2 (3), 181–212.
- Guo, Q.W., Chow, J.Y.J., Schonfeld, P., 2017. Stochastic dynamic switching in fixed and flexible transit services as market entry-exit real options. *Transp. Res. C* (in press). <https://doi.org/10.1016/j.trc.2017.08.008>.
- Hägerstrand, T., 1970. What about people in regional science? *Pap. Reg. Sci.* 24 (1), 7–24.
- Hajivassiliou, V.A., McFadden, D.L., 1998. The method of simulated scores for the estimation of LDV models. *Econometrica*, 863–896.
- Hakimi, S.L., 1964. Optimum locations of switching centers and the absolute centers and medians of a graph. *Oper. Res.* 12 (3), 450–459.
- Halat, H., Zockaei, A., Mahmassani, H.S., Xu, X., Verbas, O., 2016. Dynamic network equilibrium for daily activity-trip chains of heterogeneous travelers: application to large-scale networks. *Transportation* 43 (6), 1041–1059.
- Hamdouch, Y., Lawphongpanich, S., 2008. Schedule-based transit assignment model with travel strategies and capacity constraints. *Transp. Res. B* 42 (7), 663–684.
- Hansen, P., Jaumard, B., Savard, G., 1992. New branch-and-bound rules for linear bilevel programming. *SIAM J. Sci. Stat. Comput.* 13 (5), 1194–1217.
- Harker, P.T., 1988. Private market participation in urban mass transportation: application of computable equilibrium models of network competition. *Transp. Sci.* 22 (2), 96–111.
- Harker, P.T., Friesz, T.L., 1986. Prediction of intercity freight flows, I: theory. *Transp. Res. B* 20 (2), 139–153.
- Harsanyi, J.C., 1967. Games with incomplete information played by “Bayesian” players, I–III: Part I. The basic model. *Manag. Sci.* 14 (3), 159–182.
- Harvey, M.J., Liu, X., Chow, J.Y.J., 2016. A tablet-based surrogate system architecture for “in-situ” evaluation of cyber-physical transport technologies. *IEEE Intell. Transp. Syst. Mag.* 8 (4), 79–91.
- Hasselström, D., 1981. *Public Transportation Planning—A Math Program Approach* (Doctoral dissertation). University of Göteborg, Sweden.
- He, H., 2015. Electric taxi project in Hong Kong goes belly up: China's BYD brands 2-year campaign a ‘failure’. *South China Morning Post*. November 6, 2015.
- He, Y.B., Chow, J.Y.J., 2018. Optimal privacy control for transport network data sharing. Working paper.
- He, F., Shen, Z.J.M., 2015. Modeling taxi services with smartphone-based e-hailing applications. *Transp. Res. C* 58, 93–106.
- He, Y.B., Chow, J.Y.J., Nourinejad, M., 2017. In: A privacy design problem for sharing transport service tour data. *Proc. IEEE ITS Conference*, Yokohama, Japan.

- Hearn, D.W., 1982. The gap function of a convex program. *Oper. Res. Lett.* 1 (2), 67–71.
- Held, M., Karp, R.M., 1962. A dynamic programming approach to sequencing problems. *J. Soc. Ind. Appl. Math.* 10 (1), 196–210.
- Hensher, D.A., 2017. Future bus transport contracts under a mobility as a service (MaaS) regime in the digital age: are they likely to change? *Transp. Res. Part A Policy Pract.* 98, 86–96.
- Hendrickson, C., Kocur, G., 1981. Schedule delay and departure time decisions in a deterministic model. *Transp. Sci.* 15 (1), 62–77.
- Hensher, D.A., Greene, W.H., 2003. The mixed logit model: the state of practice. *Transportation* 30 (2), 133–176.
- Herrera, J.C., Work, D.B., Herring, R., Ban, X.J., Jacobson, Q., Bayen, A.M., 2010. Evaluation of traffic data obtained via GPS-enabled mobile phones: the Mobile Century field experiment. *Transp. Res. C* 18 (4), 568–583.
- Hillier, F.S., Lieberman, G.J., Nag, B., Basu, P., 2012. *Introduction to Operations Research*, 10th ed. Tata McGraw-Hill Education, New York.
- Hirschprung, R., Toch, E., Bolton, F., Maimon, O., 2016. A methodology for estimating the value of privacy in information disclosure systems. *Comput. Hum. Behav.* 61, 443–453.
- Hodgson, M.J., 1990. A flow-capturing location-allocation model. *Geogr. Anal.* 22 (3), 270–279.
- Holguín-Veras, J., Ozbay, K., Kornhauser, A., Brom, M., Iyer, S., Yushimito, W., Ukkusuri, S., Allen, B., Silas, M., 2011. Overall impacts of off-hour delivery programs in New York City Metropolitan Area. *Transp. Res. Rec.* 2238, 68–76.
- Holmgren, J., Davidsson, P., Persson, J.A., Ramstedt, L., 2012. TAPAS: a multi-agent-based model for simulation of transport chains. *Simul. Model. Pract. Theory* 23, 1–18.
- Hong, S.P., Kim, K.M., Byeon, G., Min, Y.H., 2017. A method to directly derive taste heterogeneity of travellers' route choice in public transport from observed routes. *Transp. Res. B* 95, 41–52.
- Hörl, S., Ruch, C., Becker, F., Frazzoli, E., Axhausen, K.W., 2017. Fleet control algorithms for automated mobility: a simulation assessment for Zurich. *Arbeitsberichte Verkehrs- und Raumplanung*, 1270.
- Horni, A., Nagel, K., Axhausen, K., 2016. Multi-Agent Transport Simulation MATSim. Ubiquity Press, London. <https://doi.org/10.5334/baw>.
- Horowitz, J.L., 1984. The stability of stochastic equilibrium in a two-link transportation network. *Transp. Res. B* 18 (1), 13–28.
- Hotelling, H., 1929. Stability in competition. *Econ. J.* 39 (153), 41–57.
- Huang, W., Li, S., Liu, X., Ban, Y., 2015. Predicting human mobility with activity changes. *Int. J. Geogr. Inf. Sci.* 29 (9), 1569–1587.
- Hunt, J.D., Stefan, K.J., 2007. Tour-based microsimulation of urban commercial movements. *Transp. Res. B* 41 (9), 981–1013.
- Huo, H., Cai, H., Zhang, Q., Liu, F., He, K., 2015. Life-cycle assessment of greenhouse gas and air emissions of electric vehicles: a comparison between China and the US. *Atmos. Environ.* 108, 107–116.
- Hyttiä, E., Penttinen, A., Sulonen, R., 2012. Non-myopic vehicle and route selection in dynamic DARP with travel time and workload objectives. *Comput. Oper. Res.* 39 (12), 3021–3030.
- Ichoua, S., Gendreau, M., Potvin, J.Y., 2006. Exploiting knowledge about future demands for real-time vehicle dispatching. *Transp. Sci.* 40 (2), 211–225.
- INCOSE, 2017. What is systems engineering? <http://www.incosse.org/AboutSE/WhatIsSE>. (Accessed 14 January 2017).
- Iteris, 2018. ARC-IT Version 8.1: The National ITS Architecture. United States Department of Transportation. <http://local.iteris.com/arc-it/index.html>. (Accessed 11 March 2018).
- Jackson, J.R., 1957. Networks of waiting lines. *Oper. Res.* 5 (4), 518–521.

- Jayakrishnan, R., Tsai, W.K., Prashker, J.N., Rajadyaksha, S., 1994. Faster path-based algorithm for traffic assignment. *Transp. Res.* Rec. 1443, 75–83.
- Jones, P.M., 1979. New approaches to understanding travel behaviour: the human activity approach. In: Hensher, D.A., Stopher, P.R. (Eds.), *Behavioral Travel Modeling*. Redwood Burn Ltd., London, pp. 55–80.
- Jung, J., Chow, J.Y.J., Jayakrishnan, R., Park, J.Y., 2014. Stochastic dynamic itinerary interception refueling location problem with queue delay for electric taxi charging stations. *Transp. Res.* C 40, 123–142.
- Kalai, E., Zemel, E., 1982. Generalized network problems yielding totally balanced games. *Oper. Res.* 30 (5), 998–1008.
- Kanafani, A., 1983. *Transportation Demand Analysis*. McGraw-Hill, New York.
- Kang, J.E., Recker, W., 2013. The location selection problem for the household activity pattern problem. *Transp. Res.* B 55, 75–97.
- Kang, J.E., Recker, W., 2014. Strategic hydrogen refueling station locations with scheduling and routing considerations of individual vehicles. *Transp. Sci.* 49 (4), 767–783.
- Kang, J.E., Chow, J.Y.J., Recker, W.W., 2013. On activity-based network design problems. *Transp. Res.* B 57, 398–418.
- Karp, R.M., 1992. On-line algorithms versus off-line algorithms: how much is it worth to know the future? *IFIP Congress* (1). vol. 12, pp. 416–429.
- Keane, M.P., 1990. *Four Essays in Empirical Macro and Labor Economics* (Doctoral dissertation). Brown University, Providence.
- Kelly, É., 2015. Helsinki's ambitious Uber for buses experiment has failed. What went wrong? *Science Business*. <http://sciencebusiness.net/news/77416/Helsinki%2099s-ambitious-Uber-for-buses-experiment-has-failed.-What-went-wrong>. (Accessed 15 January 2017).
- Kendall, D.G., 1951. Some problems in the theory of queues. *J. R. Stat. Soc. Ser. B Methodol.* 13 (2), 151–185.
- Khani, A., 2013. *Models and Solution Algorithms for Transit and Intermodal Passenger Assignment (Development of FAST-TriPs Model)* (Ph.D. dissertation). The University of Arizona.
- Kim, J.G., Kuby, M., 2012. The deviation-flow refueling location model for optimizing a network of refueling stations. *Int. J. Hydrol. Energy* 37 (6), 5406–5420.
- Kitamura, R., 1988. An evaluation of activity-based travel analysis. *Transportation* 15 (1–2), 9–34.
- Kolesar, P., Walker, W.E., 1974. An algorithm for the dynamic relocation of fire companies. *Oper. Res.* 22 (2), 249–274.
- Konduri, K.C., 2012. *Integrated Model of the Urban Continuum With Dynamic Time-Dependent Activity-Travel Microsimulation: Framework, Prototype, and Implementation* (Ph.D. dissertation). Arizona State University.
- Koppelman, F.S., 1975. *Travel Prediction With Models of Individual Choice Behavior* (Doctoral dissertation). Massachusetts Institute of Technology.
- Koppelman, F.S., Bhat, C., 2006. *A Self Instructing Course in Mode Choice Modeling: Multinomial and Nested Logit Models*. Technical report, Federal Transit Administration.
- Krok, A., 2016. Car2Stop: Car2Go shuts down services in San Diego. *CNET*. <https://www.cnet.com/roadshow/news/car2stop-car2go-shuts-down-services-in-san-diego/>. (Accessed 15 January 2017).
- Kumar, P., Nigam, S.P., Kumar, N., 2014. Vehicular traffic noise modeling using artificial neural network approach. *Transp. Res.* C 40, 111–122.
- Kurauchi, F., Bell, M.G., Schnöcker, J.D., 2003. Capacity constrained transit assignment with common lines. *J. Math. Model. Algorithms* 2 (4), 309–327.

- Kwan, M.P., 2000. Interactive geovisualization of activity-travel patterns using three-dimensional geographical information systems: a methodological exploration with a large data set. *Transp. Res.* C 8 (1), 185–203.
- Kwan, M.P., Lee, J., 2004. Geovisualization of human activity patterns using 3D GIS: a time-geographic approach. *Spat. Integr. Soc. Sci.* 27.
- Lagos, R., 2000. An alternative approach to search frictions. *J. Polit. Econ.* 108 (5), 851–873.
- Lam, W.H., Yin, Y., 2001. An activity-based time-dependent traffic assignment model. *Transp. Res.* B 35 (6), 549–574.
- Lam, W.H.K., Gao, Z.Y., Chan, K.S., Yang, H., 1999. A stochastic user equilibrium assignment model for congested transit networks. *Transp. Res.* B 33 (5), 351–368.
- Lampkin, W., Saalmans, P.D., 1967. The design of routes, service frequencies, and schedules for a municipal bus undertaking: a case study. *J. Oper. Res. Soc.* 18 (4), 375–397.
- Land, A.H., Doig, A.G., 1960. An automatic method of solving discrete programming problems. *Econometrica* 28 (3), 497–520.
- Langevin, A., Mbaraga, P., Campbell, J.F., 1996. Continuous approximation models in freight distribution: an overview. *Transp. Res.* B 30 (3), 163–188.
- Laporte, G., 1992a. The traveling salesman problem: an overview of exact and approximate algorithms. *Eur. J. Oper. Res.* 59 (2), 231–247.
- Laporte, G., 1992b. The vehicle routing problem: an overview of exact and approximate algorithms. *Eur. J. Oper. Res.* 59 (3), 345–358.
- Larson, R.C., 1974. A hypercube queuing model for facility location and redistricting in urban emergency services. *Comput. Oper. Res.* 1 (1), 67–95.
- Larson, R.C., Odoni, A.R., 1981. *Urban Operations Research*. Prentice-Hall, NJ.
- Le Ny, J., Pappas, G.J., 2014. Differentially private filtering. *IEEE Trans. Autom. Control* 59 (2), 341–354.
- LeBlanc, L.J., 1975. An algorithm for the discrete network design problem. *Transp. Sci.* 9 (3), 183–199.
- LeBlanc, L.J., Morlok, E.K., Pierskalla, W.P., 1975. An efficient approach to solving the road network equilibrium traffic assignment problem. *Transp. Res.* 9 (5), 309–318.
- Lee, M.S., McNally, M.G., 2003. On the structure of weekly activity/travel patterns. *Transp. Res.* A 37 (10), 823–839.
- Lee, G., You, S., Ritchie, S., Saphores, J.D., Sangkapichai, M., Jayakrishnan, R., 2009. Environmental impacts of a major freight corridor: a study of I-710 in California. *Transp. Res. Rec.* 2123, 119–128.
- Li, Z.C., Lam, W., Sumalee, A., 2008. Modeling impact of transit operator fleet size under various market regimes with uncertainty in network. *Transp. Res. Rec.* 2063, 18–27.
- Li, L., Chu, W., Langford, J., Schapire, R.E., 2010. In: A contextual-bandit approach to personalized news article recommendation. Proceedings of the 19th International Conference on World Wide Web. ACM, pp. 661–670.
- Li, C., Yang, H., Zhu, D., Meng, Q., 2012a. A global optimization method for continuous network design problems. *Transp. Res.* B Methodol. 46 (9), 1144–1158.
- Li, Z.C., Lam, W.H., Wong, S.C., 2012b. Optimization of number of operators and allocation of new lines in an oligopolistic transit market. *Netw. Spat. Econ.* 12 (1), 1–20.
- Li, L., Chen, X., Zhang, L., 2014. Multimodel ensemble for freeway traffic state estimations. *IEEE Trans. Intell. Transp. Syst.* 15 (3), 1323–1336.
- Li, Z.C., Guo, Q.W., Lam, W.H., Wong, S.C., 2015. Transit technology investment and selection under urban population volatility: a real option perspective. *Transp. Res.* B Methodol. 78, 318–340.
- Liao, F., Arentze, T.A., Timmermans, H.J.P., 2010. Supernetwork approach for multimodal and multi-activity travel planning. *Transp. Res. Rec.* 2175, 38–46.

- Liao, F., Arentze, T.A., Timmermans, H.J.P., 2013. Incorporating space-time constraints and activity travel time profiles in a multi-state supernetwork approach to individual activity-travel scheduling. *Transp. Res. B* 55, 41–58.
- Lin, S., 1965. Computer solutions of the traveling salesman problem. *Bell Syst. Tech. J.* 44 (10), 2245–2269.
- Lin, S., Kernighan, B.W., 1973. An effective heuristic algorithm for the traveling-salesman problem. *Oper. Res.* 21 (2), 498–516.
- Lin, D.Y., Eluru, N., Waller, S.T., Bhat, C.R., 2008. Integration of activity-based modeling and dynamic traffic assignment. *Transp. Res. Rec.* 2076, 52–61.
- Little, J.D., 1961. A proof for the queuing formula: $L = \lambda W$. *Oper. Res.* 9 (3), 383–387.
- Liu, H.X., He, X., He, B., 2009. Method of successive weighted averages (MSWA) and self-regulated averaging schemes for solving stochastic user equilibrium problem. *Netw. Spat. Econ.* 9 (4), 485–503.
- Liu, X., Liu, K., Guo, L., Li, X., Fang, Y., 2013. In: A game-theoretic approach for achieving k-anonymity in location based services. *INFOCOM, 2013 Proceedings IEEE*. IEEE, pp. 2985–2993.
- Liu, P., Liao, F., Huang, H.J., Timmermans, H., 2015a. Dynamic activity-travel assignment in multi-state supernetworks. *Transp. Res. B* 81 (P3), 656–671.
- Liu, X., Gong, L., Gong, Y., Liu, Y., 2015b. Revealing travel patterns and city structure with taxi trip data. *J. Transp. Geogr.* 43, 78–90.
- Liu, X., Yan, W.Y., Chow, J.Y.J., 2015c. Time-geographic relationships between vector fields of activity patterns and transport systems. *J. Transp. Geogr.* 42, 22–33.
- Liu, J., Kang, J.E., Zhou, X., Pendyala, R., 2017. Network-oriented household activity pattern problem for system optimization. *Transp. Res. Procedia* 23, 827–847.
- Liu, X., Chow, J.Y.J., Li, S., 2018. Online monitoring of local taxi travel momentum and congestion effects using projections of taxi GPS-based vector fields. *J. Geogr. Syst.* <https://doi.org/10.1007/s10109-018-0273-6>.
- Logi, F., Ritchie, S.G., 2002. A multi-agent architecture for cooperative inter-jurisdictional traffic congestion management. *Transp. Res. C* 10 (5–6), 507–527.
- Longstaff, F.A., Schwartz, E.S., 2001. Valuing American options by simulation: a simple least-squares approach. *Rev. Financ. Stud.* 14 (1), 113–147.
- Lorion, A., Harvey, M.J., Chow, J.Y.J., 2014. Redesign of curricula in transit systems planning to meet data-driven challenges. *J. Prof. Issues Eng. Educ. Pract.* 141 (3). 05014007.
- Luce, R.D., 1959. On the possible psychophysical laws. *Psychol. Rev.* 66 (2), 81–95.
- Luehrman, T.A., 1998. Strategy as a portfolio of real options. *Harv. Bus. Rev.* 76, 89–101.
- Luque-Baena, R.M., López-Rubio, E., Domínguez, E., Palomo, E.J., Jerez, J.M., 2015. A self-organizing map to improve vehicle detection in flow monitoring systems. *Soft. Comput.* 19 (9), 2499–2509.
- Lv, Y., Duan, Y., Kang, W., Li, Z., Wang, F.Y., 2015. Traffic flow prediction with big data: a deep learning approach. *IEEE Trans. Intell. Transp. Syst.* 16 (2), 865–873.
- Ma, X., Yu, H., Wang, Y., Wang, Y., 2015. Large-scale transportation network congestion evolution prediction using deep learning theory. *PLoS ONE* 10(3). e0119044.
- Ma, T.Y., Chow, J.Y.J., Xu, S.J., 2017a. Causal structure learning for travel mode choice using structural restrictions and model averaging algorithm. *Transportmetrica A* 13 (4), 299–325.
- Ma, Z., Urbanek, M., Pardo, M.A., Chow, J.Y.J., Lai, X., 2017b. Spatial welfare effects of shared taxi operating policies for first mile airport access. *Int. J. Transp. Sci. Technol.* 6 (4), 301–315.
- Ma, T.Y., Chow, J.Y.J., Rasulkhani, S., 2018. An integrated dynamic ridesharing dispatch and idle vehicle repositioning strategy on a bimodal transport network. *Proc. Transport Research Arena 2018*, Vienna, Austria.

- MaaS Finland, 2016. Mobility-as-a-Service (MaaS) launches first on-demand mobility service in Finland. Telematics Wire. <http://telematicswire.net/mobility-as-a-service-maas-launches-first-on-demand-mobility-service-in-finland/>. (Accessed 18 April 2017).
- Magnanti, T.L., Wong, R.T., 1984. Network design and transportation planning: models and algorithms. *Transp. Sci.* 18 (1), 1–55.
- Mahmassani, H.S., 1990. Dynamic models of commuter behavior: experimental investigation and application to the analysis of planned traffic disruptions. *Transp. Res. A* 24 (6), 465–484.
- Mahmassani, H.S., 2016. 50th anniversary invited article—autonomous vehicles and connected vehicle systems: flow and operations considerations. *Transp. Sci.* 50 (4), 1140–1162.
- Mahmassani, H.S., Chang, G.L., 1986. Experiments with departure time choice dynamics of urban commuters. *Transp. Res. B* 20 (4), 297–320.
- Mallozzi, L., 2007. Noncooperative facility location games. *Oper. Res. Lett.* 35 (2), 151–154.
- Manheim, M.L., 1979. Fundamentals of Transportation Systems Analysis. The MIT Press, Cambridge.
- Manheim, M.L., 1980. Understanding “supply” in transportation systems. *Transp. Res. A* 14 (2), 119–135.
- Manski, C.F., 1977. The structure of random utility models. *Theor. Decis.* 8 (3), 229–254.
- Mao, F., Ji, M., Liu, T., 2016. Mining spatiotemporal patterns of urban dwellers from taxi trajectory data. *Front. Earth Sci.* 10 (2), 205–221.
- Marcotte, P., 1985. A new algorithm for solving variational inequalities with application to the traffic assignment problem. *Math. Program.* 33 (3), 339–351.
- Marcotte, P., 1986. Network design problem with congestion effects: a case of bilevel programming. *Math. Program.* 34 (2), 142–162.
- Marcotte, P., Zhu, D.L., 1996. Exact and inexact penalty methods for the generalized bilevel programming problem. *Math. Program.* 74 (2), 141–157.
- Marianov, V., Revelle, C., 1994. The queuing probabilistic location set covering problem and some extensions. *Socio Econ. Plan. Sci.* 28 (3), 167–178.
- Marianov, V., ReVelle, C., 1996. The queueing maximal availability location problem: a model for the siting of emergency vehicles. *Eur. J. Oper. Res.* 93 (1), 110–120.
- Marianov, V., Serra, D., 2002. Location-allocation of multiple-server service centers with constrained queues or waiting times. *Ann. Oper. Res.* 111 (1), 35–50.
- Markakis, E., Saberi, A., 2005. On the core of the multicommodity flow game. *Decis. Support. Syst.* 39 (1), 3–10.
- McFadden, D., 1974. Conditional logit analysis of qualitative choice behavior. In: Zarembka, P. (Ed.), *Frontiers in Econometrics*. Academic Press, New York, pp. 105–142.
- McFadden, D., Train, K., 2000. Mixed MNL models for discrete response. *J. Appl. Econ.* 15 (5), 447–470.
- McNally, M.G., 2007. The four step model. In: Hensher, D.A., Button, K.J. (Eds.), *Handbook of Transport Modelling*, second ed., vol. 1. Elsevier, pp. 35–41.
- Megiddo, N., 1978. Cost allocation for Steiner trees. *Networks* 8 (1), 1–6.
- Megiddo, N., Zemel, E., Hakimi, S.L., 1983. The maximum coverage location problem. *SIAM J. Algebraic Discret. Methods* 4 (2), 253–261.
- Miller, H.J., 1991. Modelling accessibility using space-time prism concepts within geographical information systems. *Int. J. Geogr. Inf. Sci.* 5 (3), 287–301.
- Miller, C.C., 2014. Is owning overrated? The rental economy rises. *The New York Times*. August 29, 2014.

- Miller, H.J., Bridwell, S.A., 2009. A field-based theory for time geography. *Ann. Assoc. Am. Geogr.* 99 (1), 49–75.
- Miller, E.J., Roorda, M.J., 2003. A prototype model of 24-hour household activity scheduling for the Toronto area. *Transp. Res.* 1831, 114–121.
- Miller, C.E., Tucker, A.W., Zemlin, R.A., 1960. Integer programming formulation of traveling salesman problems. *JACM* 7 (4), 326–329.
- Miller, J., Nie, Y., Stathopoulos, A., 2017. Crowdsourced urban package delivery: modeling traveler willingness to work as crowdshippers. *Transp. Res.* 2610, 67–75.
- Miranda, F., Doraiswamy, H., Lage, M., Zhao, K., Gonçalves, B., Wilson, L., Hsieh, M., Silva, C.T., 2017. Urban pulse: capturing the rhythm of cities. *IEEE Trans. Vis. Comput. Graph.* 23 (1), 791–800.
- Mitrović-Minić, S., Krishnamurti, R., Laporte, G., 2004. Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows. *Transp. Res. B Methodol.* 38 (8), 669–685.
- Mohring, H., 1972. Optimization and scale economies in urban bus transportation. *Am. Econ. Rev.* 62 (4), 591–604.
- Möller, D.P., 2014. *Introduction to Transportation Analysis, Modeling and Simulation*. Springer, London.
- Moore, O., 2013. Toronto to take over struggling Bixi bike-share program. *The Globe and Mail* 2013. December 4.
- Mordukhovich, B.S., 2004. Equilibrium problems with equilibrium constraints via multiobjective optimization. *Optim. Methods Soft.* 19 (5), 479–492.
- Morganti, E., Dablanc, L., Fortin, F., 2014. Final deliveries for online shopping: the deployment of pickup point networks in urban and suburban areas. *Res. Transp. Bus. Manag.* 11, 23–31.
- Mosheiov, G., 1994. The travelling salesman problem with pick-up and delivery. *Eur. J. Oper. Res.* 79 (2), 299–310.
- Muñuzuri, J., Cortés, P., Grosso, R., Guadix, J., 2012. Selecting the location of minihubs for freight delivery in congested downtown areas. *J. Comput. Sci.* 3 (4), 228–237.
- Murchland, J.D., 1970. Braess's paradox of traffic flow. *Transp. Res.* 4 (4), 391–394.
- Murray, C.C., Chu, A.G., 2015. The flying sidekick traveling salesman problem: optimization of drone-assisted parcel delivery. *Transp. Res. C* 54, 86–109.
- Najmi, A., Rashidi, T.H., Abbasi, A., Waller, S.T., 2016. Reviewing the transport domain: an evolutionary bibliometrics and network analysis. *Scientometrics*, 1–23.
- Neumann, A., Nagel, K., 2013. Passenger agent and paratransit operator reaction to changes of service frequency of a fixed train line. *Procedia Comput. Sci.* 19, 803–808.
- Neutens, T., Van de Weghe, N., Witlox, F., De Maeyer, P., 2008. A three-dimensional network-based space-time prism. *J. Geogr. Syst.* 10 (1), 89–107.
- Newell, G.F., 1987. The morning commute for nonidentical travelers. *Transp. Sci.* 21 (2), 74–88.
- Nguyen, S., Dupuis, C., 1984. An efficient method for computing traffic equilibria in networks with asymmetric transportation costs. *Transp. Sci.* 18 (2), 185–202.
- Nguyen, S., Pallottino, S., 1988. Equilibrium traffic assignment for large scale transit networks. *Eur. J. Oper. Res.* 37 (2), 176–186.
- Nourinejad, M., Chow, J.Y., Roorda, M.J., 2016. Equilibrium scheduling of vehicle-to-grid technology using activity based modelling. *Transp. Res. C* 65, 79–96.
- Nuzzolo, A., Russo, F., Crisalli, U., 2001. A doubly dynamic schedule-based assignment model for transit networks. *Transp. Sci.* 35 (3), 268–285.
- Ohashi, H., Kim, T.S., Oum, T.H., Yu, C., 2005. Choice of air cargo transshipment airport: an application to air cargo traffic to/from Northeast Asia. *J. Air Transp. Manag.* 11 (3), 149–159.

- Øksendal, B., 1992. Stochastic Differential Equations: An Introduction With Applications. Springer Science & Business Media, Berlin.
- Ortuzar, J.D.D., Willumsen, L.G., 2002. Modelling Transport. vol. 3. Wiley & Sons, Chichester.
- Owen, S.H., Daskin, M.S., 1998. Strategic facility location: a review. *Eur. J. Oper. Res.* 111 (3), 423–447.
- Özdamar, L., Demir, O., 2012. A hierarchical clustering and routing procedure for large scale disaster relief logistics planning. *Transp. Res. E* 48 (3), 591–602.
- Park, R.E., Burgess, E.W., McKenzie, R.D., 1984. The City. University of Chicago Press, Chicago.
- Patriksson, M., 2004. Sensitivity analysis of traffic equilibria. *Transp. Sci.* 38 (3), 258–281.
- Pendyala, R.M., Yamamoto, T., Kitamura, R., 2002. On the formulation of time-space prisms to model constraints on personal activity-travel engagement. *Transportation* 29 (1), 73–94.
- Pendyala, R.M., Kitamura, R., Kikuchi, A., Yamamoto, T., Fujii, S., 2005. FAMOS: the Florida Activity Mobility Simulator. *Transp. Res. Rec.* 1921, 123–130.
- Perera, C., Zaslavsky, A., Christen, P., Georgakopoulos, D., 2014. Sensing as a service model for smart cities supported by internet of things. *Trans. Emerg. Telecommun. Technol.* 25 (1), 81–93.
- Perl, J., Daskin, M.S., 1985. A warehouse location-routing problem. *Transp. Res. B Methodol.* 19 (5), 381–396.
- Pigou, A.C., 1920. The Economics of Welfare. McMillan & Co, London.
- Pinjari, A.R., Bhat, C.R., 2011. Activity-based travel demand analysis. In: A Handbook of Transport Economics, vol. 10. Edward Elgar Publishing, Cheltenham, pp. 213–248.
- Plastria, F., 2001. Static competitive facility location: an overview of optimisation approaches. *Eur. J. Oper. Res.* 129 (3), 461–470.
- Poorzahedy, H., Abulghasemi, F., 2005. Application of ant system to network design problem. *Transportation* 32 (3), 251–273.
- Potters, J., Reijnierse, H., Biswas, A., 2006. The nucleolus of balanced simple flow networks. *Games Econ. Behav.* 54 (1), 205–225.
- Powell, W.B., 2011. Approximate Dynamic Programming: Solving the Curses of Dimensionality, second ed. vol. 703. John Wiley & Sons, Hoboken.
- Powell, W.B., Ryzhov, I.O., 2012. Optimal Learning. vol. 841. John Wiley & Sons, Hoboken.
- Powell, W.B., Sheffi, Y., 1982. The convergence of equilibrium algorithms with predetermined step sizes. *Transp. Sci.* 16 (1), 45–55.
- Prim, R.C., 1957. Shortest connection networks and some generalizations. *Bell Labs Tech.J.* 36 (6), 1389–1401.
- Puu, T., Beckmann, M., 1999. Continuous space modelling. In: Handbook of Transportation Science. Springer, US, pp. 269–310.
- Qian, Z.S., Xiao, F.E., Zhang, H.M., 2012. Managing morning commute traffic with parking. *Transp. Res. B* 46 (7), 894–916.
- Ramadurai, G., Ukkusuri, S., 2010. Dynamic user equilibrium model for combined activity-travel choices using activity-travel supernetwork representation. *Netw. Spat. Econ.* 10 (2), 273–292.
- Rashidi, T.H., Mohammadian, A., 2011. Household travel attributes transferability analysis: application of a hierarchical rule based approach. *Transportation* 38 (4), 697–714.
- Recker, W.W., 1995. The household activity pattern problem: general formulation and solution. *Transp. Res. B* 29 (1), 61–77.
- Recker, W.W., 2001. A bridge between travel demand modeling and activity-based travel analysis. *Transp. Res. B* 35 (5), 481–506.

- Recker, W.W., Parimi, A., 1999. Development of a microscopic activity-based framework for analyzing the potential impacts of transportation control measures on vehicle emissions. *Transp. Res.* D 4 (6), 357–378.
- Recker, W.W., McNally, M.G., Root, G.S., 1986a. A model of complex travel behavior: Part I—theoretical development. *Transp. Res.* A 20 (4), 307–318.
- Recker, W.W., McNally, M.G., Root, G.S., 1986b. A model of complex travel behavior: Part II. An operational model. *Transp. Res.* A 20 (4), 319–330.
- Regis, R.G., Shoemaker, C.A., 2007. A stochastic radial basis function method for the global optimization of expensive functions. *INFORMS J. Comput.* 19 (4), 497–509.
- Regue, R., Recker, W., 2014. Proactive vehicle routing with inferred demand to solve the bikesharing rebalancing problem. *Transp. Res. E-Log. Transp. Rev.* 72, 192–209.
- Reimer, J., 2012. From Altair to iPad: 35 years of personal computer market share. *Ars Technica*. [http://arstechnica.com/business/2012/08/from-altair-to-ipad-35-years-of-personal-computer-market-share/4/](http://arstechnica.com/business/2012/08/from-altair-to-ipad-35-years-of-personal-computer-market-share/). (Accessed 14 January 2017).
- ReVelle, C.S., Swain, R.W., 1970. Central facilities location. *Geogr. Anal.* 2 (1), 30–42.
- Rieser, M., 2010. Adding Transit to an Agent-Based Transportation Simulation: Concepts and Implementation (Ph.D. dissertation). TU, Berlin.
- Rochet, J.C., Tirole, J., 2003. Platform competition in two-sided markets. *J. Eur. Econ. Assoc.* 1 (4), 990–1029.
- Rochet, J.C., Tirole, J., 2006. Two-sided markets: a progress report. *RAND J. Econ.* 37 (3), 645–667.
- Rodriguez-Roman, D., 2018. A surrogate-assisted genetic algorithm for the selection and design of highway safety and travel time improvement projects. *Saf. Sci.* 103, 305–315.
- Roth, A.E., Sotomayor, M.A.O., 1990. Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis. Cambridge University Press, New York.
- Ruan, M., Lin, J.J., Kawamura, K., 2012. Modeling urban commercial vehicle daily tour chaining. *Transp. Res. E* 48 (6), 1169–1184.
- Russo, F., Comi, A., 2010. A modelling system to simulate goods movements at an urban scale. *Transportation* 37 (6), 987–1009.
- Ryzhov, I.O., Powell, W.B., 2011. Information collection on a graph. *Oper. Res.* 59 (1), 188–201.
- Saharidis, G.K., Ierapetritou, M.G., 2009. Resolution method for mixed integer bi-level linear problems based on decomposition technique. *J. Glob. Optim.* 44 (1), 29–51.
- Samuelson, P.A., 1952. Spatial price equilibrium and linear programming. *Am. Econ. Rev.* 42 (3), 283–303.
- Sankar, L., Kar, S., Tandon, R., Poor, H.V., 2011. In: Competitive privacy in the smart grid: an information-theoretic approach. 2011 IEEE International Conference on Smart Grid Communications (SmartGridComm). IEEE, pp. 220–225.
- Savage, S.J., Waldman, D.M., 2015. Privacy tradeoffs in smartphone applications. *Econ. Lett.* 137, 171–175.
- Sayarshad, H.R., Chow, J.Y.J., 2015. A scalable non-myopic dynamic dial-a-ride and pricing problem. *Transp. Res.* B 81, 539–554.
- Sayarshad, H.R., Chow, J.Y.J., 2016. Survey and empirical evaluation of nonhomogeneous arrival process models with taxi data. *J. Adv. Transp.* 50 (7), 1275–1294.
- Sayarshad, H.R., Chow, J.Y.J., 2017. Non-myopic relocation of idle mobility-on-demand vehicles as a dynamic location-allocation-queueing problem. *Transp. Res.* E 106, 60–77.
- Schllich, R., Axhausen, K.W., 2003. Habitual travel behaviour: evidence from a six-week travel diary. *Transportation* 30 (1), 13–36.
- Schöbel, A., 2012. Line planning in public transportation: models and methods. *OR Spectr.* 34 (3), 491–510.
- Schöbel, A., Scholl, S., 2006. In: Line planning with minimal traveling time. ATMOS 2005—5th Workshop on Algorithmic Methods and Models for Optimization of Railways. Internationales Begegnungs-und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl.

- Sheffi, Y., 1985. *Urban Transportation Networks*. Prentice Hall, Englewood Cliffs, NJ.
- Shen, J., Liu, X., Chen, M., 2017. Discovering spatial and temporal patterns from taxi-based Floating Car Data: a case study from Nanjing. *GIScience Remote Sens.* 54 (5), 1–22.
- Sheth, R., 2017. Concept design for Kutsuplus. <http://riddhi.me/projects/kutsuplus>. (Accessed 28 April 2017).
- Shoup, D.C., 2005. *The High Cost of Free Parking*. vol. 206. Planners Press, Chicago.
- Silman, L.A., Barzily, Z., Passy, U., 1974. Planning the route system for urban buses. *Comput. Oper. Res.* 1 (2), 201–211.
- Small, K.A., 1982. The scheduling of consumer activities: work trips. *Am. Econ. Rev.* 72 (3), 467–479.
- Small, K.A., Rosen, H.S., 1981. Applied welfare economics with discrete choice models. *Econometrica*, 105–130.
- Small, K.A., Verhoef, E.T., 2007. *The Economics of Urban Transportation*. Routledge, Abingdon.
- Smith, M.J., 1979. The existence, uniqueness and stability of traffic equilibria. *Transp. Res. B* 13 (4), 295–304.
- Smith, M.J., 1984a. Two alternative definitions of traffic equilibrium. *Transp. Res. B* 18 (1), 63–65.
- Smith, M.J., 1984b. The existence of a time-dependent equilibrium distribution of arrivals at a single bottleneck. *Transp. Sci.* 18 (4), 385–394.
- Smith, M.J., 1984c. The stability of a dynamic model of traffic assignment—an application of a method of Lyapunov. *Transp. Sci.* 18 (3), 245–252.
- Smith, M., Hazelton, M.L., Lo, H.K., Cantarella, G.E., Watling, D.P., 2014. The long term behaviour of day-to-day traffic assignment models. *Transportmetrica A* 10 (7), 647–660.
- Sødal, S., Koekebakker, S., Aadland, R., 2008. Market switching in shipping—a real option model applied to the valuation of combination carriers. *Rev. Financ. Econ.* 17 (3), 183–203.
- Solomon, M.M., Desrosiers, J., 1988. Survey paper—time window constrained routing and scheduling problems. *Transp. Sci.* 22 (1), 1–13.
- Spiess, H., Florian, M., 1989. Optimal strategies: a new assignment model for transit networks. *Transp. Res. B* 23 (2), 83–102.
- Spivey, M.Z., Powell, W.B., 2004. The dynamic assignment problem. *Transp. Sci.* 38 (4), 399–419.
- Srinivasan, K., Mahmoodani, H., 2000. Modeling inertia and compliance mechanisms in route choice behavior under real-time information. *Transp. Res. Rec.* 1725, 45–53.
- Srinivasan, K.K., Mahmoodani, H.S., 2005. A dynamic kernel logit model for the analysis of longitudinal discrete choice data: properties and computational assessment. *Transp. Sci.* 39 (2), 160–181.
- Srivastava, S.S., Kumar, S., Garg, R.C., Sen, P., 1969. Generalized traveling salesman problem through n sets of nodes. *CORS J.* 7, 97–101.
- Steenbrink, P.A., 1974. *Optimization of Transport Networks*. John Wiley & Sons.
- Steenhof, P., Woudsma, C., Sparling, E., 2006. Greenhouse gas emissions and the surface transport of freight in Canada. *Transp. Res. D* 11 (5), 369–376.
- Stentoft, L., 2004. Convergence of the least squares Monte Carlo approach to American option valuation. *Manag. Sci.* 50 (9), 1193–1203.
- Stopher, P.R., Meyburg, A.H., 1976. *Transportation Systems Evaluation*. Lexington Books, Lexington.
- Sulopuisto, O., 2016. Why Helsinki's innovative on-demand bus service failed. *Citiscope*. March 4.
- Sun, Z., Zan, B., Ban, X.J., Gruteser, M., 2013. Privacy protection method for fine-grained urban traffic modeling using mobile sensors. *Transp. Res. B* 56, 50–69.
- Sussman, J., 2000. *Introduction to Transportation Systems*. Artech House Publishers, Norwood.

- Suwansirikul, C., Friesz, T.L., Tobin, R.L., 1987. Equilibrium decomposed optimization: a heuristic for the continuous equilibrium network design problem. *Transp. Sci.* 21 (4), 254–263.
- Sweeney, L., 2002. k-Anonymity: a model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl. Based Syst.* 10 (05), 557–570.
- Talluri, K.T., Van Ryzin, G.J., 2004. *The Theory and Practice of Revenue Management*. Springer, Boston.
- Tang, J., Liu, F., Wang, Y., Wang, H., 2015. Uncovering urban human mobility from large scale taxi GPS data. *Physica A* 438, 140–153.
- Taniguchi, E., Shimamoto, H., 2004. Intelligent transportation system based dynamic vehicle routing and scheduling with variable travel times. *Transp. Res. C* 12 (3), 235–250.
- Tarantola, A., 2005. *Inverse Problem Theory and Methods for Model Parameter Estimation*. SIAM, Philadelphia.
- Tagsetiren, M.F., Smith, A.E., 2000. In: A genetic algorithm for the orienteering problem. Proc. 2000 Congress on Evolutionary Computation, IEEE. vol. 2, pp. 910–915.
- Tebaldi, C., West, M., 1998. Bayesian inference on network traffic using link count data. *J. Am. Stat. Assoc.* 93 (442), 557–573.
- Teitz, M.B., Bart, P., 1968. Heuristic methods for estimating the generalized vertex median of a weighted graph. *Oper. Res.* 16 (5), 955–961.
- Tenorio, L., 2001. Statistical regularization of inverse problems. *SIAM Rev.* 43 (2), 347–366.
- Thomas, B.W., White III, C.C., 2004. Anticipatory route selection. *Transp. Sci.* 38 (4), 473–487.
- Tobin, R.L., Friesz, T.L., 1988. Sensitivity analysis for equilibrium network flow. *Transp. Sci.* 22 (4), 242–250.
- Tomlin, J.A., 1971. A mathematical programming model for the combined distribution-assignment of traffic. *Transp. Sci.* 5 (2), 122–140.
- Tong, C.O., Wong, S.C., 1999. A stochastic transit assignment model using a dynamic schedule-based network. *Transp. Res. B* 33 (2), 107–121.
- Toth, P., Vigo, D. (Eds.), 2002. *The Vehicle Routing Problem*. Society for Industrial and Applied Mathematics, Philadelphia.
- Train, K.E., 2009. *Discrete Choice Methods With Simulation*. Cambridge University Press, New York.
- Transport Canada, 2015. Canadian ITS architecture. <https://www.tc.gc.ca/eng/innovation/its-architecture-download-1588.html>.
- Trigeorgis, L., 1996. *Real Options: Managerial Flexibility and Strategy in Resource Allocation*. MIT Press, Cambridge.
- Tsekrekos, A.E., 2010. The effect of mean reversion on entry and exit decisions under uncertainty. *J. Econ. Dyn. Control.* 34 (4), 725–742.
- Tsai, Y.C., Wang, S.L., Kao, H.Y., Hong, T.P., 2015. Edge types vs privacy in K-anonymization of shortest paths. *Appl. Soft Comput.* 31, 348–359.
- UN, 2015. The world population prospects: 2015 revision. <http://www.un.org/en/development/desa/publications/world-population-prospects-2015-revision.html> (Accessed 14 January 2017).
- USDOT, 2007. Systems engineering for intelligent transportation systems. Federal Highway Administration & Federal Transit Administration.
- USDOT, 2015. Beyond traffic: the smart city challenge. <https://www.its.dot.gov/factsheets/pdf/SmartCities.pdf>.
- USDOT, 2016. Smart city challenge lessons learned. <https://www.transportation.gov/sites/dot.gov/files/docs/Smart%20City%20Challenge%20Lessons%20Learned.pdf>.
- Van der Poort, E.S., Libura, M., Sierksma, G., van der Veen, J.A., 1999. Solving the k-best traveling salesman problem. *Comput. Oper. Res.* 26 (4), 409–425.

- Van Heerden, Q., Joubert, J.W., 2014. Generating intra and inter-provincial commercial vehicle activity chains. *Procedia Soc. Behav. Sci.* 125, 136–146.
- Van Ryzin, G., Mahajan, S., 1999. On the relationship between inventory costs and variety benefits in retail assortments. *Manag. Sci.* 45 (11), 1496–1509.
- Van Zuylen, H.J., Willumsen, L.G., 1980. The most likely trip matrix estimated from traffic counts. *Transp. Res. B* 14 (3), 281–293.
- Vardi, Y., 1996. Network tomography: estimating source-destination traffic intensities from link data. *J. Am. Stat. Assoc.* 91 (433), 365–377.
- Verbas, Ö., Mahmassani, H.S., Hyland, M.F., 2016. Gap-based transit assignment algorithm with vehicle capacity constraints: simulation-based implementation and large-scale application. *Transp. Res. B* 93, 1–16.
- Vickrey, W.S., 1954. The economizing of curb parking space. *Traffic Eng.* 25 (2), 62–67.
- Vickrey, W.S., 1969. Congestion theory and transport investment. *Am. Econ. Rev.* 59 (2), 251–260.
- Vilajosana, I., Llosa, J., Martinez, B., Domingo-Prieto, M., Angles, A., Vilajosana, X., 2013. Bootstrapping smart cities through a self-sustainable model based on big data flows. *IEEE Commun. Mag.* 51 (6), 128–134.
- Vlahogianni, E.I., Karlaftis, M.G., Golias, J.C., 2014. Short-term traffic forecasting: where we are and where we're going. *Transp. Res. C* 43, 3–19.
- Vlasic, B., Boudette, N.E., 2016. Self-driving Tesla was involved in fatal crash, U.S. says. *The New York Times*. June 30, 2016.
- Voccia, S.A., Melissa Campbell, A., Thomas, B.W., 2017. The same-day delivery problem for online purchases. *Transp. Sci.* (in press). <https://doi.org/10.1287/trsc.2016.0732>.
- Vodopivec, N., Miller-Hooks, E., 2017. An optimal stopping approach to managing travel-time uncertainty for time-sensitive customer pickup. *Transp. Res. B Methodol.* 102, 22–37.
- Vovsha, P., Bekhor, S., 1998. Link-nested logit model of route choice: overcoming route overlapping problem. *Transp. Res. Rec.* 1645, 133–142.
- Wahba, M., Shalaby, A., 2009. MILATRAS: a new modeling framework for the transit assignment problem. In: *Schedule-Based Modeling of Transportation Networks*. Springer, New York, pp. 1–24.
- Wang, L., 2009. Cutting plane algorithms for the inverse mixed integer linear programming problem. *Oper. Res. Lett.* 37 (2), 114–116.
- Wang, Q., Taylor, J.E., 2014. Quantifying human mobility perturbation and resilience in Hurricane Sandy. *PLoS ONE* 9(11). e112608.
- Wang, S., Meng, Q., Yang, H., 2013. Global optimization methods for the discrete network design problem. *Transp. Res. B Methodol.* 50, 42–60.
- Wardrop, J.G., 1952. In: Some theoretical aspects of road traffic research. ICE Proceedings: Engineering Divisions, vol. 1, no. 3. Thomas Telford, pp. 325–362.
- Washington, S.P., Karlaftis, M.G., Mannering, F., 2010. *Statistical and Econometric Methods for Transportation Data Analysis*. CRC Press, Boca Raton.
- Watling, D., Hazelton, M.L., 2003. The dynamics and equilibria of day-to-day assignment models. *Netw. Spat. Econ.* 3 (3), 349–370.
- WHO, 2010. Bulletin of the World Health Organization: urbanization and health. <http://www.who.int/bulletin/volumes/88/4/10-010410/en/>. (Accessed 14 January 2017).
- Wilson, A.G., 1967. A statistical theory of spatial distribution models. *Transp. Res.* 1, 253–269.
- Wilson, N.H.M., Sussman, J.M., Goodman, L.A., Hignett, B.T., 1969. Simulation of a computer aided routing system (CARS). Proceedings of the Third Conference on Applications of Simulation. Winter Simulation Conference, December, pp. 171–183.
- Wisetjindawat, W., Sano, K., Matsumoto, S., 2006. Commodity distribution model incorporating spatial interactions for urban freight movement. *Transp. Res. Rec.* 1966, 41–50.

- Wolsey, L.A., 1981. Integer programming duality: price functions and sensitivity analysis. *Math. Program.* 20 (1), 173–195.
- Wong, K.I., Wong, S.C., Bell, M.G., Yang, H., 2005. Modeling the bilateral micro-searching behavior for urban taxi services using the absorbing Markov chain approach. *J. Adv. Transp.* 39 (1), 81–104.
- Work, D.B., Bayen, A.M., 2008. In: Impacts of the mobile internet on transportation cyber-physical systems: traffic monitoring using smartphones. National Workshop for Research on High-Confidence Transportation Cyber-Physical Systems: Automotive, Aviation, & Rail, pp. 18–20.
- Wu, J.H., Florian, M., Marcotte, P., 1994. Transit equilibrium assignment: a model and solution algorithms. *Transp. Sci.* 28 (3), 193–203.
- Xu, H., Pang, J.S., Ordóñez, F., Dessouky, M., 2015. Complementarity models for traffic equilibrium with ridesharing. *Transp. Res. B* 81, 161–182.
- Xu, S.J., Nourinejad, M., Lai, X., Chow, J.Y.J., 2017. Network learning via multi-agent inverse transportation problems. *Transp. Sci.* (in press). <https://doi.org/10.1287/trsc.2017.0805>.
- Yang, H., 1997. Sensitivity analysis for the elastic-demand network equilibrium problem with applications. *Transp. Res. B* 31 (1), 55–70.
- Yang, H., Bell, M.G.H., 2007. In: Sensitivity analysis of network traffic equilibrium revisited: the corrected approach. 4th IMA International Conference on Mathematics in Transport Institute of Mathematics and its Applications.
- Yang, H., Huang, H.J., 1998. Principle of marginal-cost pricing: how does it work in a general road network? *Transp. Res. A* 32 (1), 45–54.
- Yang, H., Wong, S.C., 1998. A network model of urban taxi services. *Transp. Res. B* 32 (4), 235–246.
- Yang, H., Woo, K., 2000. Competition and equilibria of private toll roads in a traffic network. *Transp. Res. Rec.* 1733, 15–22.
- Yang, H., Yang, T., 2011. Equilibrium properties of taxi markets with search frictions. *Transp. Res. B* 45 (4), 696–713.
- Yang, F., Zhang, D., 2009. Day-to-day stationary link flow pattern. *Transp. Res. B* 43 (1), 119–126.
- Yang, H., Sasaki, T., Iida, Y., Asakura, Y., 1992. Estimation of origin-destination matrices from link traffic counts on congested networks. *Transp. Res. B* 26 (6), 417–434.
- Yang, H., Yagar, S., Iida, Y., Asakura, Y., 1994. An algorithm for the inflow control problem on urban freeway networks with user-optimal flows. *Transp. Res. B Methodol.* 28 (2), 123–139.
- Yang, H., Bell, M.G.H., 1998. Models and algorithms for road network design: a review and some new developments. *Transp. Rev.* 18 (3), 257–278.
- Yang, H., Wong, S.C., Wong, K.I., 2002. Demand-supply equilibrium of taxi services in a network under competition and regulation. *Transp. Res. B* 36 (9), 799–819.
- Yang, H., Liu, W., Wang, X., Zhang, X., 2013. On the morning commute problem with bottleneck congestion and parking space constraints. *Transp. Res. B* 58, 106–118.
- You, S.I., Chow, J.Y.J., Ritchie, S.G., 2016. Inverse vehicle routing for activity-based urban freight forecast modeling and city logistics. *Transportmetrica A* 12 (7), 650–673.
- Yu, R., Abdel-Aty, M., 2014. Analyzing crash injury severity for a mountainous freeway incorporating real-time traffic and weather data. *Saf. Sci.* 63, 50–56.
- Yue, Y., Zhuang, Y., Li, Q., Mao, Q., 2009. Mining time-dependent attractive areas and movement patterns from taxi trajectory data. 2009 17th International Conference on Geoinformatics. IEEE, pp. 1–6.
- Yue, Y., Lan, T., Yeh, A.G., Li, Q.Q., 2014. Zooming into individuals to understand the collective: a review of trajectory-based travel behaviour studies. *Travel Behav. Soc.* 1 (2), 69–78.

- Zanella, A., Bui, N., Castellani, A., Vangelista, L., Zorzi, M., 2014. Internet of things for smart cities. *IEEE Internet Things J.* 1 (1), 22–32.
- Zha, L., Yin, Y., Yang, H., 2016. Economic analysis of ride-sourcing markets. *Transp. Res. C* 71, 249–266.
- Zha, L., Yin, Y., Du, Y., 2017. Surge pricing and labor supply in the ride-sourcing market. *Transp. Res. Procedia* 23, 2–21.
- Zhang, J., Zhang, L., 2010. An augmented Lagrangian method for a class of inverse quadratic programming problems. *Appl. Math. Optim.* 61 (1), 57.
- Zhang, D., Nagurney, A., Wu, J., 2001. On the equivalence between stationary link flow patterns and traffic network equilibria. *Transp. Res. B* 35 (8), 731–748.
- Zhang, X., Yang, H., Huang, H.J., 2011a. Improving travel efficiency by parking permits distribution and trading. *Transp. Res. B* 45 (7), 1018–1034.
- Zhang, X., Zhang, H.M., Huang, H.J., Sun, L., Tang, T.Q., 2011b. Competitive, cooperative and Stackelberg congestion pricing for multiple regions in transportation networks. *Transportmetrica* 7 (4), 297–320.
- Zhao, T., Sundararajan, S.K., Tseng, C.L., 2004. Highway development decision-making under uncertainty: a real options approach. *J. Infrastruct. Syst.* 10 (1), 23–32.
- Zheng, S., Chen, S., 2016. Fleet replacement decisions under demand and fuel price uncertainties. *Transp. Res. D* (in press). <https://doi.org/10.1016/j.trd.2016.09.001>.
- Zheng, Y., Liu, Y., Yuan, J., Xie, X., 2011. In: *Urban computing with taxicabs*. Proc. 13th International Conference on Ubiquitous Computing, September, pp. 89–98.
- Zhou, Y., Fang, Z., Thill, J.C., Li, Q., Li, Y., 2015. Functionally critical locations in an urban transportation network: identification and space-time analysis using taxi trajectories. *Comput. Environ. Urban. Syst.* 52, 34–47.
- Zhou, J., Lam, W.H., Heydecker, B.G., 2005. The generalized Nash equilibrium model for oligopolistic transit market with elastic demand. *Transp. Res. B Methodol.* 39 (6), 519–544.
- Zhu, Y., Xie, K., Ozbay, K., Zuo, F., Yang, H., 2017. Data-driven spatial modeling for quantifying networkwide resilience in the aftermath of hurricanes Irene and Sandy. *Transp. Res. Rec.* 2604, 9–18.
- Zubieta, L., 1998. A network equilibrium model for oligopolistic competition in city bus services1. *Transp. Res. B Methodol.* 32 (6), 413–422.
- Zumel, N., 2015. A simpler explanation of differential privacy. <http://www.win-vector.com/blog/2015/10/a-simpler-explanation-of-differential-privacy/>. (Accessed 10 December 2017).

Appendices

The book is written for an audience that is assumed to have some basic knowledge in certain areas: systems engineering, queueing analysis, and discrete choice modeling. Those without such information can make use of these appendices drawn from various lecture notes. Some topic areas such as probability and GIS are assumed to be widely known from both traditional transportation disciplines and smart cities audiences that they are not reviewed here.

Appendix A: Rankings of Transportation Research at Universities around the World

Appendix B: Systems Engineering

Appendix C: Queueing Analysis

Appendix D: Discrete Choice Modeling

A RANKINGS OF TRANSPORTATION RESEARCH AT UNIVERSITIES AROUND THE WORLD

See [Table A.1](#).

This table compares the Scopus h-index of transportation research by academic institution, as searched on November 18, 2016. The methodology is described in [Chow \(2016a\)](#) and includes more than just urban transport systems research. The left side shows top rankings for all time while the right side shows top rankings from recent research. The range of h-indices suggests an institution with research within a 10-year period having an h-index in the range of high 20s–30s would be highly ranked worldwide. Readers interested in this type of bibliometric analysis for the transportation field should also look at [Najmi et al. \(2016\)](#).

Table A.1 Transportation research h-index rankings by institution for (left) all years up to 2016, and (right) for papers published in 2007–16
h-Index from publications over all years to 2016 **h-Index from publications in 2007–16**

	Institution	h-Index	Country		Institution	h-Index	Country
1	UC Berkeley	65	United States	1	UC Berkeley	39	United States
2	U. Montreal	61	Canada	2	TU Delft	37	Netherlands
3	UT Austin	57	United States	3	University of Sydney	33	Australia
4	MIT	56	United States	4	UT Austin	32	United States
5	UC Davis	52	United States	5	U. British Columbia	31	Canada
5	HKUST	52	China	5	National U. Singapore	31	Singapore
7	U. Sydney	48	Australia	7	HK Polytechnic	30	China
7	UMN Twin Cities	48	United States	7	Georgia Tech	30	United States
9	U. Leeds	47	United Kingdom	9	U. Leeds	29	United Kingdom
10	TU Delft	46	Netherlands	9	MIT	29	United States
11	Georgia Tech	45	United States	9	UMN Twin Cities	29	United States
12	UC Irvine	43	United States	9	U. Waterloo	29	Canada
12	U. College London	43	United Kingdom	9	U. Montreal	29	Canada
12	Ohio State	43	United States	14	Texas A&M College Station	28	United States
15	UMD-College Park	42	United States	14	Tsinghua U.	28	China
15	HK Polytechnic	42	China	14	UC Davis	28	United States
15	U. Washington Seattle	42	United States	14	HKUST	28	China
18	Texas A&M College Station	40	United States	18	Monash U.	27	Australia
18	Virginia Tech	40	United States	19	Virginia Tech	26	United States
18	National U. Singapore	40	Singapore	19	UMD-College Park	26	United States
18	U. Waterloo	40	Canada	19	NTU Athens	26	Greece

22	U. British Columbia	39	Canada	19	Arizona State U.	26	United States
22	Northwestern	39	United States	23	Purdue U.	25	United States
24	U. Michigan	38	United States	23	U. College London	25	United Kingdom
24	U. Toronto	38	Canada	23	U. Toronto	25	Canada
26	Purdue U.	37	United States	23	Queensland U. Tech.	25	Australia
26	Monash U.	37	Australia	23	U. Florida	25	United States
26	U. Hong Kong	37	China	23	U. Hong Kong	25	China
26	Imperial College London	37	United Kingdom	23	National Cheng Kung U.	25	Taiwan
30	NTU Athens	35	Greece	23	EPFL	25	Switzerland

B SYSTEMS ENGINEERING

Definitions of systems and systems engineering are provided by the International Council on Systems Engineering (INCOSE).

Definition B.1 *A system is a construct or collection of different elements that together produce results not obtainable by the elements alone.*

Definition B.2 *Systems engineering is a methodical, disciplined approach for the design, realization, technical management, operations, and retirement of a system.*

The systems engineering approach features several components linked in a process diagram shown in Bahill and Gissing (1998) in Fig. B.1. The key features are a focus on customer needs and requirements, a general breakdown of the system into components and integrated backup to the system level (the so-called V model (USDOT, 2007)), and having a continuous reevaluation or feedback loop in place.

The components contribute to an overall life cycle for the “systems engineering process.” Blanchard et al. (1998) identify six steps to this process:

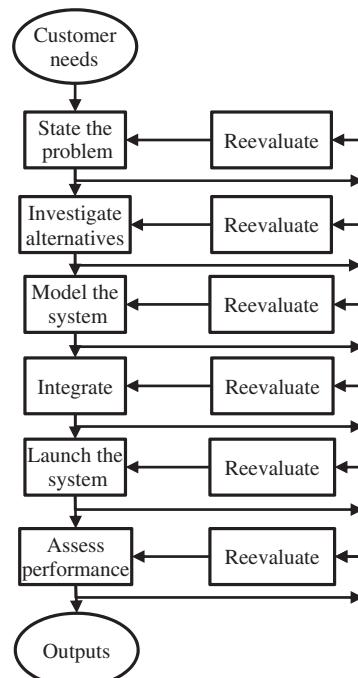


Fig. B.1 Systems engineering process.

- (1) Conceptual design: moving from client needs to identifying functional requirements and constraints
- (2) Preliminary design: specifying the functional flow of the system and designing the parameters
- (3) Detailed design and development: evaluating, prototyping, and testing the system against the client needs
- (4) Production and/or construction: producing the system and making it operational
- (5) Utilization and support: maintaining the system during its operation
- (6) Phaseout and disposal

Systems engineers are especially needed in the first three steps of the process, with the research and development focusing more on the first step. Any new system that is to be deployed, particularly intelligent transportation systems, should go through those steps with a systems engineer to ensure that a functional prototype of a system considers all the component interactions, addresses all the client needs as specified, and does so in a manner that is measurably effective (often against a benchmark system).

The conceptual design step includes a feasibility study to assess the needs, convert them into system operational requirements, and to identify a set of functions needed to address those requirements. It is breaking up the system into individual components, one for each function. Due to the potential complexity of systems, there needs to be a clear way to communicate the structure of a system and its components between different stakeholders throughout the process.

The Unified Modeling Language (UML) (see [Ambler, 2017](#)) has emerged as one of the more popular tools for communicating system designs. Currently UML is at version 2.x and consists of a series of 13 diagrams used to describe various aspects of a system. These are listed in [Table B.1](#) from [Ambler \(2017\)](#).

For identifying the needs and relating them to high-level functions, a use case diagram and a class diagram are effective tools. The use case diagram provides an overview of essentially the scope of the system and consists of four types of objects: “use cases,” “actors,” “relationships,” and “system boundary boxes.” Use cases are the reasons for a system to exist, which drive the derivation of the user needs. The U.S. National ITS Architecture calls these “service packages.” The relationships and boundary boxes identify high-level interactions that exist between specific use cases and actors. A use case diagram is typically drawn by first listing out all the actors (some stick figure icon), followed by ovals representing different system use cases

Table B.1 UML diagrams (Ambler, 2017)

Diagram	Description
Activity diagram	Depicts high-level business processes, including data flow, or to model the logic of complex logic within a system.
Class diagram	Shows a collection of static model elements such as classes and types, their contents, and their relationships.
Communication diagram	Shows instances of classes, their interrelationships, and the message flow between them. Communication diagrams typically focus on the structural organization of objects that send and receive messages.
Component diagram	Depicts the components that compose an application, system, or enterprise. The components, their interrelationships, interactions, and their public interfaces are depicted.
Composite structure diagram	Depicts the internal structure of a classifier (such as a class, component, or use case), including the interaction points of the classifier to other parts of the system.
Deployment diagram	Shows the execution architecture of systems. This includes nodes, either hardware or software execution environments, as well as the middleware connecting them.
Interaction overview diagram	A variant of an activity diagram which overviews the control flow within a system or business process.
Object diagram	Depicts objects and their relationships at a point in time, typically a special case of either a class diagram or a communication diagram.
Package diagram	Shows how model elements are organized into packages as well as the dependencies between packages.
Sequence diagram	Models the sequential logic, in effect the time ordering of messages between classifiers.
State machine diagram	Describes the states an object or interaction may be in, as well as the transitions between states.
Timing diagram	Depicts the change in state or condition of a classifier instance or role over time.
Use case diagram	Shows use cases, actors, and their interrelationships.

(these should all be actions involving how an actor, which includes the system itself, may use the system in some way). Boundary boxes may be used to classify distinct subsystems. Lines are drawn to connect actors to the use cases that impact them directly. Arrows can be used to connect use cases with each other with some association, such as having Use Case A <<extend>> to Use Case B (an arrow is then drawn from B to A).

As an example, consider two alternative systems for bus ticketing: a roadside ticketing kiosk and a ticketing kiosk on the bus. Use case diagrams for the two alternative fare payment systems are shown in Fig. B.2. This figure illustrates how there is no dependency on boarding wait time to the use case of fare payment for roadside ticketing and no tying up the bus driver.

The class diagram is designed to further break out the different classes in the system and to capture their relationships (e.g., inheritance, aggregation, association), operations, and attributes. There are two types of class diagrams: conceptual and design level diagrams. A conceptual diagram can be used at the conceptual and preliminary design stages and expand to a more detailed design class diagram in the detailed design stage. Whereas the use case diagram identifies the relationships needed, the class diagram helps to identify the key variables needed to relate the variables. It is especially useful when designing data-oriented systems. Classes are distinct objects of which multiple instances may exist. The classes of interest are defined by rectangles. Variables of interest within a class are identified within a connected rectangle. Aggregation is represented with a triangular icon separating the root to the branches. Other relationships are expressed using arrows with an action phrase describing the relationship. Specific numbers (e.g., many to one)

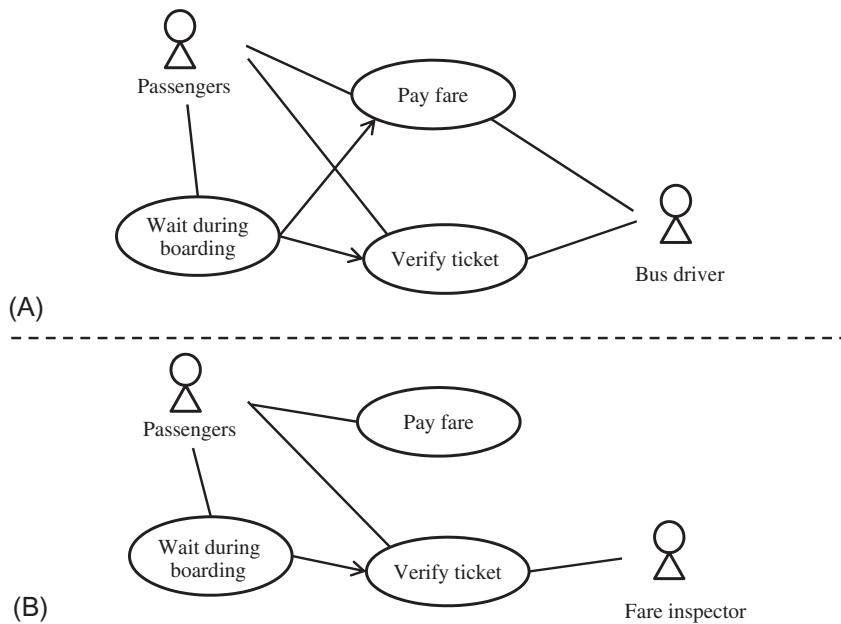


Fig. B.2 Use case diagrams for (A) on-bus fare ticketing and (B) roadside fare ticketing.

and types of queue disciplines are noted on the ends of the arrows corresponding to the class.

Following on the roadside ticketing example in Fig. B.2, a conceptual class diagram at this stage for the roadside ticketing may look something like Fig. B.3. Even for the same system, different class diagrams may exist. The design of the diagram itself depends on what aspects of the system most concerns the client. In the roadside ticketing example, the key trade-off is the change in bus dwell time and the automation of the fare payment for passengers. In this case, those elements are highlighted more (with the ticket queue and bus queue classes). If the focus was on how passengers would pay the system, that aspect of the system might be expanded further instead.

A class diagram provides a good entry into the preliminary design stage. In this stage, the emphasis is on carving the different functions and subsystems needed to serve the use cases for the different classes defined. During preliminary design, the functions are designed to ensure data flows between subsystems are properly captured and constraints are met. Trade-off analyzes are conducted between function parameters. At this stage it may be useful to use mathematical models to support the design and trade-off analysis. For

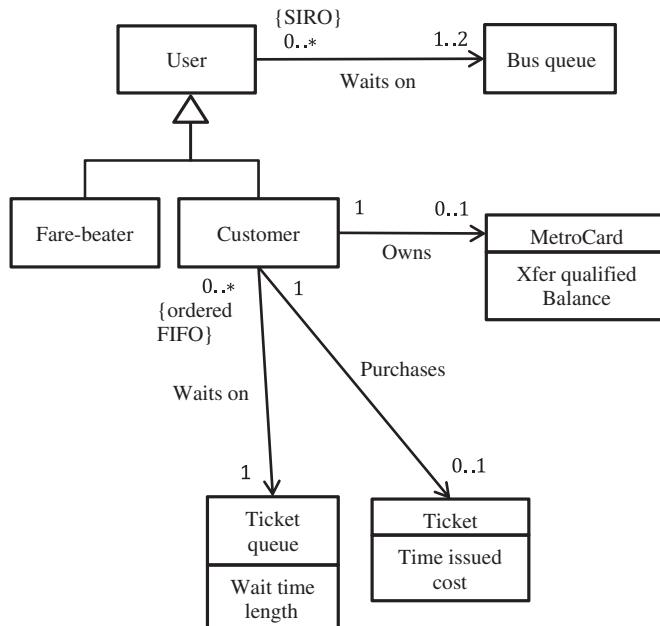


Fig. B.3 A conceptual diagram for the roadside bus ticketing system.

example, mathematical programming models may be employed to capture constraints and determine what would be suitable decision variables to achieve specific objectives.

Traditionally, “functional flow” diagrams are used to express the relationships between different functions and to capture the data flow requirements. Under the UML framework, other diagrams also can be used. One such diagram is the activity diagram. Each diagram is drawn to fulfill a use case and shows the logic between different functions to handle the use case. Functions are defined using rounded rectangles and two circles (one all black and one partially blackened) to define the start and end of a use case interaction. Triangles can be used to capture aggregation of components and diamonds are used to define decision nodes to split the logic path. Conditions for the decisions are captured in italicized brackets. Mutually exclusive activity sequences are captured using bars.

An example of an activity diagram is shown for the “Pay Fare” use case in Fig. B.4.

While the diagrams so far explain the functional relationships in a system, they do not describe the physical relationships. A deployment diagram, also known as a technical architecture diagram and a physical diagram, does that. A deployment diagram identifies the physical objects in the system (e.g., server, vehicles, various centers where data is stored), the functions that are stored within them, and the data that flows between them. An example from the ARC-IT V8.1 National ITS Architecture from the US DOT is shown in Fig. B.5. In this example of the transit fare collection management

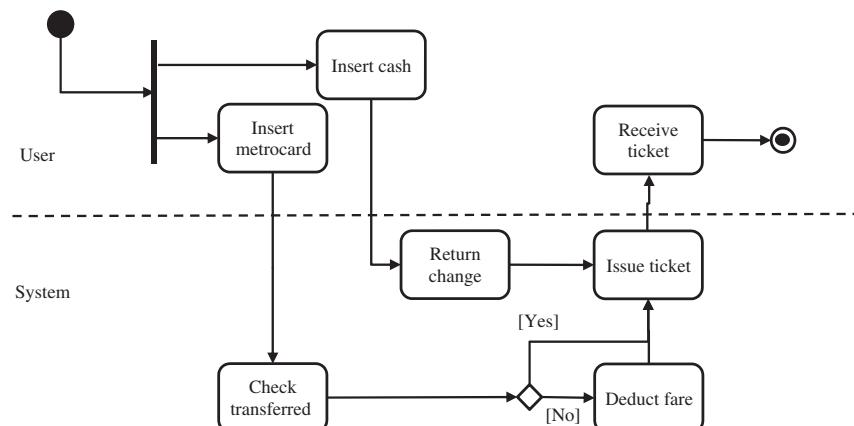


Fig. B.4 An activity diagram with swimlanes for Pay Fare use case in roadside ticketing.

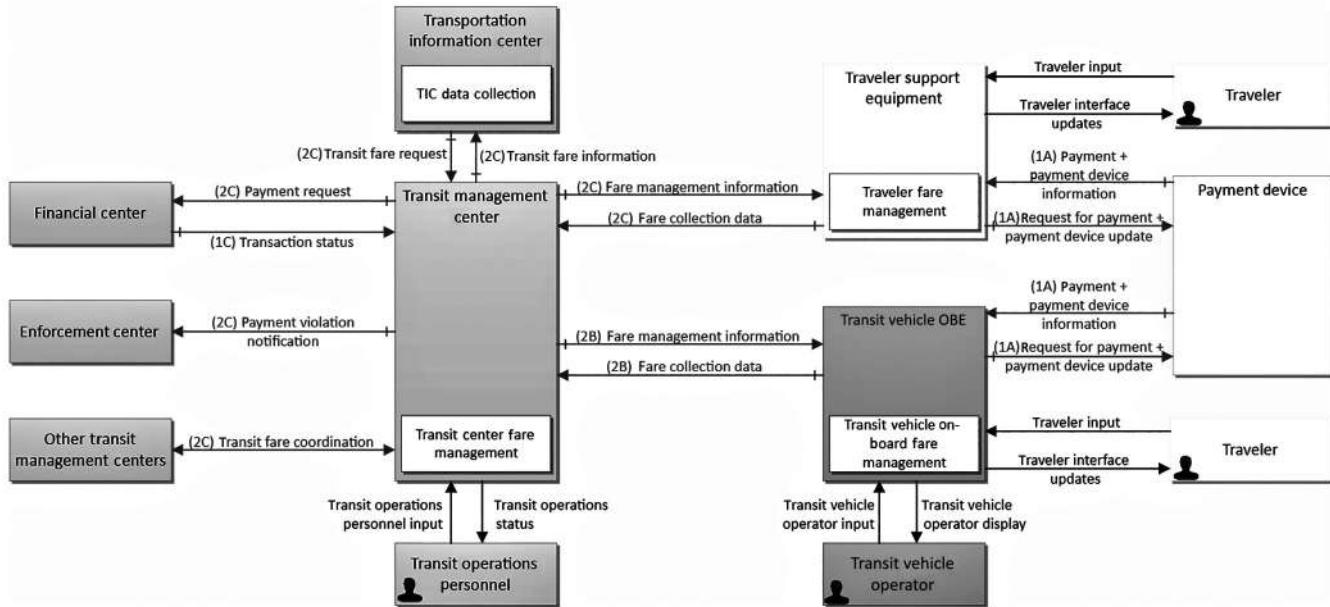


Fig. B.5 A deployment diagram of the PT04 Transit Fare Collection Management use case. (Source: Iteris, 2018.)

use case, physical objects like the transit management center, financial center, and transit vehicle onboard equipment (OBE) are listed as rectangles. Functions are listed as rectangles within these rectangles (e.g., TIC Data Collection). Data flows are captured by the arrows.

At the end of preliminary design, a systems engineer should have a clear set of diagrams that describe the use cases, how they are served by various functions and data flows, and the physical objects that run the functions. Trade-offs should have been conducted to understand the feasible design spaces and their sensitivities to different parameters. If decision support involves some optimization model, then the models should be clearly formulated.

In the detailed design stage, a systems engineer goes through the process of designing all the parameters of the system and testing the effectiveness of the system to different performance measures. Much of the testing at this stage may be based on a prototype or in a simulation or model environment. This is very much an iterative process where parameters are set for a given design and evaluated. After evaluating several design alternatives, a preferred system design alternative is chosen. Example performance measures in transportation are presented in [Table B.2](#), which is a selection taken from [Iteris \(2018\)](#).

Table B.2 Selection of ITS performance measures

Object category	Performance measure
Arterial management: delay	Control delay seconds per vehicle.
Arterial management: reliability	The buffer index (represents the extra time (buffer) travelers add to their average travel time when planning trips to arrive on-time 95% of the time).
Emergency/incident management: incident duration	Mean roadway clearance time per incident.
Emergency/incident management: traveler information	Time to alert motorists of an incident/emergency.
Emergency/incident management: use of technology	Number of regional roadway miles covered by ITS-related assets in use for incident detection.
Environment: clean air and climate change	Fine particulate (PM2.5) emissions—tons per day
Environment: clean air and climate change	Carbon dioxide emissions—tons per day
Freeway management: HOV lanes	Share of freeway network with HOV lanes.

Continued

Table B.2 Selection of ITS performance measures—cont'd

Object category	Performance measure
Freeway management: pricing and tolling	Percentage of drivers with ETC transponders.
Freeway management: transportation management centers	Percent of regional transportation system monitored by the TMC for real-time performance.
Freight management: border crossing	Average border crossing time for freight at international borders per year.
Freight management: detours and routing	Percent of detours of freeways and major arterials that can accommodate commercial vehicles.
Freight management: intermodal facilities	Average duration of delays per month at intermodal facilities.
Preservation: preserve existing infrastructure	Pavement condition index
Safety: vehicle crashes and fatalities	Total fatalities per X VMT.
Special event management: mode shift from SOV	Percent of special event attendees utilizing park and ride lots each year for selected events.
System efficiency: energy consumption	Excess fuel consumed (total or per capita).
System efficiency: vehicle miles traveled	Average VMT per capita per day, per week, or per year.
System options: bicycle and pedestrian accessibility and efficiency	Percent of roadways with bicycle and pedestrian facilities.
System options: modal options for individuals with disabilities	The percent of transit stops with ADA provisions.
System reliability: transit on-time performance	On-time performance of transit.
Transit operations and management: automated fare collection	Percent of total transfers performed with automated fare cards.
Transit operations and management: loading standards	Duration of standee time.
Transit operations and management: transit signal priority	Travel time delay on routes with queue jumping and automated vehicle location in use.
Travel demand management: commuter shuttle service	Percent of residents in region receiving marketing material on shuttle service opportunities.
Travel demand management: parking management	Capacity of park and ride lots.
Travel weather management: clearance time (weather-related debris)	Average time to clear selected surface transportation facilities of weather-related debris after weather impact.
Work zone management: extent of congestion	Length of average and maximum queues in work zones.

In addition to these performance measures, in transport system design involving algorithms other measures include computational efficiency, optimality of the design with respect to some desired objective, and data requirements as they relate to privacy.

Based on a set of alternative designs and performance measures, a carefully prepared test plan is needed to evaluate each system design's performance. For many systems, the testing must be first conducted in a simulation environment before being deployed in a field pilot. A simulation-based test plan should lay out the following:

- **Simulation rules:** The underlying simulation rules and how they reflect or ignore real-world conditions that would be resolved in a field pilot. A simulation that incorporates feedback from human participants has “human-in-the-loop” characteristics while one that incorporates feedback from physical components (e.g., scale models or control device) has “hardware-in-the-loop” features.
- **Common random elements:** The random elements within the simulation that are shared by all scenarios.
- **Performance measures and experimental design:** The measures of performance used to evaluate all scenarios plus the design of the experiment to evaluating each scenario (e.g., number of simulation runs, duration of cold start, or sampling methodology).
- The set of scenarios and how they correspond to specific system designs within the simulation environment; one should be a benchmark alternative in which data from the real world is used to calibrate.

As an example, in detailed design of a fare payment system, one may be interested in comparing the passenger wait time under two designs: one system where payment is made onboard a bus and a second where payment is made at a roadside kiosk. A test plan could then involve:

- Simulation rules: a specified period length where there is a specified passenger arrival rate, in which each passenger seeks to board the next bus arriving at a fixed headway and a random load of onboard passengers with a random number alighting. Enforcement is assumed to take place. This simulation design ignores network effects (delays would impact downstream headways and passenger arrivals).
- Common random elements: arrival rates, number of passengers onboard, percent of them alighting. Other random elements may be different population groups such as number of handicapped individuals or passengers who may be potential fare-beaters. Distributions should be specified, whether there are correlations or not, and justified for each element.

- Performance measures: dwell time for onboard passengers, time to board for passengers at the stop, number of fare-beaters missed. The test plan should define the experimental design: in this case, one might specify running 100 instances of the simulation and using the performance measures sampled from those 100 instances to construct a distribution.
- The two design alternatives are specified as the scenarios. Assuming the onboard fare payment is the benchmark alternative, data from the real world is used to calibrate this scenario and the common random elements. Then the new design alternatives are specified accordingly.

C QUEUEING ANALYSIS

C.1 Markov Chains

Transport systems are naturally queueing systems since they involve limited capacity service operated over time. Queueing analysis involves the study of the congestion effects within this limited capacity environment, typically under a steady-state setting.

Before covering queueing, an introduction needs to be made of the system setting. Queues form in a dynamic system where the state is characterized by number of people in the system or queue. While the future state is unknown, its distribution can be discerned from the current state. Under such systems, states transition from one to another with a certain probability. The transitions between states are governed by stochastic processes which is called a Markov process (Markov chain) under continuous (discrete) time. In a Markov chain, the probability of the next system state is dependent on the current system state, that is, $P[X_{t+1}=j | X_t=i]$. The material on Markov chains is generally referenced from [Grinstead and Snell \(2012\)](#) unless stated otherwise.

Definition C.1 *Markov chains* are stochastic processes where the probability of a state at $t+1$ only depends on the state at t .

A subset of Markov chains has transition probabilities that do not change over time, for each i and j , $P[X_{t+1}=j | X_t=i] = P[X_1=j | X_0=i], \forall t=1, 2, \dots$. These are *stationary* probabilities, and the subset is called a stationary Markov chain.

One example of a stationary transition matrix is for weather. Consider the matrix shown in [Table C.1](#). In this system there are three states, $\{Sunny, Cloudy, Rain\}$, and they change from 1 day t to the next. This transition matrix can be estimated from data to reflect the numbers within.

Table C.1 Example transition matrix

	Sunny [t+1]	Cloudy [t+1]	Rain [t+1]
Sunny [t]	0.6	0.3	0.1
Cloudy [t]	0.4	0.4	0.2
Rain [t]	0.2	0.4	0.4

In this weather system, if today it is sunny then the likelihood of it being sunny again the next day is 60%.

The transition probabilities can also be visualized as a *state transition diagram*, shown in Fig. C.1. Each oval represents a state while the arrows represent the transition probabilities. A state j that has a transition probability $p_{jj} = 1$ is called an *absorbing state*.

The transition probabilities, which are conditional probabilities, can be used to understand limiting characteristics of the system. First, the conditional probabilities may extend to n time steps. Let $P_{ij}^{[n]} = P[X_{t+n} = j | X_t = i]$. The probability distribution can be iteratively expanded as follows.

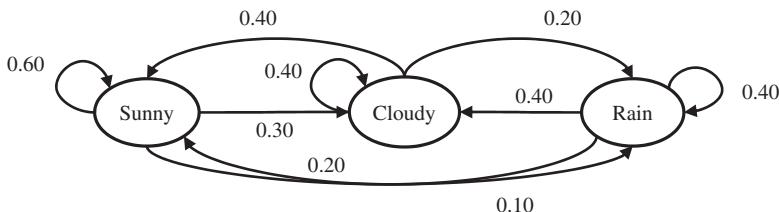
$$P_{ij}^{[n]} = \sum_{k=1}^M p_{ik} p_{kj}^{[n-1]} = \sum_{k=1}^M p_{ik} \sum_{k=1}^M p_{ik} p_{kj}^{[n-2]} = \dots$$

This is called the *Chapman-Kolmogorov* equations (see Hillier et al., 2012), which are stated formally in Eq. (C.1).

$$P_{ij}^{[n]} = \sum_{k=1}^M p_{ik}^{[m]} p_{kj}^{[n-m]}, \forall i = 1, \dots, M, j = 1, \dots, M, \text{any } m = 1, 2, \dots, n-1 \quad (\text{C.1})$$

If, for example, $n=2$ time steps and a Markov transition matrix of size $M=2$, then $P_{ij}^{[2]} = \sum_{k=1}^2 p_{ik} p_{kj}^{[1]} = p_{i1} p_{1j} + p_{i2} p_{2j}$. This is also equivalent to p_{ij}^2 . We can therefore state Eq. (C.2) to examine probability distributions at some future time step.

$$P^{[n]} = P^n \quad (\text{C.2})$$

**Fig. C.1** Example state transition diagram.

In the steady state, there are two types of outcomes. In the first type, the Markov chain has one or more absorbing states and the system eventually ends up in those states. In the second type, the Markov chain is *ergodic* if it is possible to eventually go from every state to every other state. Furthermore, a *regular* chain is an ergodic chain in which there exists n such that $p_{ij}^n > 0$ for all i, j . A regular Markov chain has a limiting matrix W representing the steady-state distribution in each state.

For the first type of limiting scenario, Markov chains can be used to answer questions like the probability that the process ends up in a certain absorbing state, the mean time to be absorbed, and the mean number of times the process ends up in a transient state. Answers to these questions can be obtained by first rearranging the transition probability matrix into the transient states followed by the absorbing states as shown in Eq. (C.3). The matrix Q is the set of transition probabilities for the transient states, R is for the absorbing states, and I is an identity matrix.

$$P = \begin{bmatrix} Q & R \\ 0 & I \end{bmatrix} \quad (\text{C.3})$$

Then $N = (I - Q)^{-1}$ is called the “fundamental matrix” of P . The n_{ij} of N gives the expected number of times that the process is in transient state j if started in transient state i . If we define c as a column vector of all 1’s, then t_i , where $t = Nc$, denotes the expected number of steps taken before the chain is absorbed if it starts in state i . If $B = NR$, then b_{ij} of B gives the probability of being absorbed by state j if the process starts at state i .

For ergodic and regular Markov chains, as $n \rightarrow \infty$ the matrix $P^n \rightarrow W$. We know that the limiting matrix is reached when all the row vectors are the same. In the example with the weather, it takes $n = 12$ for all the row vectors to be the same:

$$W = \begin{bmatrix} 0.4516 & 0.3548 & 0.1935 \\ 0.4516 & 0.3548 & 0.1935 \\ 0.4516 & 0.3548 & 0.1935 \end{bmatrix}$$

The $1 \times M$ row vector w can also be found by solving the following fixed point problem shown in Eq. (C.4).

$$\begin{aligned} w &= wP \\ \sum_{j=1}^M w_j &= 1 \end{aligned} \quad (\text{C.4})$$

This may be solved as follows:

$$\begin{bmatrix} P^T - I \\ 1_{1 \times M} \end{bmatrix} w^T = \begin{bmatrix} 0_{M \times 1} \\ 1 \end{bmatrix}$$

C.2 Queues

A length of a queue can be viewed as a type of continuous time Markov process, where there is a probability for each value from zero to infinity. Queueing analysis was first conducted by Erlang (1909) by using state balance equations to determine queue delays. The following terminology is used to characterize a queue.

- $a[t]$ =number of arrivals in system
- $c[t]$ =number of service completions
- $N[t] = a[t] - c[t]$ =number of users in system at time τ
- $t_a[i]$ =arrival time of user i
- $t_b[i]$ =time of service beginning for user i
- $t_c[i]$ =time of service end for user i
- $w_q[i] = t_b[i] - t_a[i]$ =wait time in queue for user i
- $w[i] = t_c[i] - t_a[i]$ =total time in system for user i
- $W(W_q)$ =average wait time of all users in the system (queue)
- $I[t] = \int_0^t N[\tau] d\tau$ is the total number of users served in $[0, \tau]$
- $L(L_q)$ =average number of people in the system (queue)
- λ = mean number of arrivals per unit time
- μ = mean number of service completions per unit time
- $\rho = \frac{\lambda}{\mu}$ = utilization ratio

Queues share some common characteristics. A queue discipline defines the rules used in clearing a queue. Examples include FIFO (first-in-first-out), LIFO (last-in-first-out), and SIRO (served in random order). A queue has user arrivals that follow a certain distribution. The service rate also follows a certain distribution and may be more than one server operating for a single queue. The queue may have a maximum capacity over which new users do not enter the system (they balk). The wait time is based on a combination of random arrivals, service times, and structure of the queue.

To make things easier for describe queues, Kendall (1951) came up with a notation in the format $A/B/m/k$, where A is a distribution of the user interarrival time (M if Markov with negative exponential arrivals, D if deterministic, and G if it is a general distribution). B is the service time

distribution. The m denotes the number of parallel servers and k is the queue capacity. An example is M/M/2/3, which means the arrival and service times are negative exponential distributions with two parallel servers and a maximum queue of three people allowed.

In queueing analysis, the objective is in general to estimate the steady-state properties of the queue (length and delay) under an alternative queue design. In the case of certain simple queues, more detailed information can be derived. [Little \(1961\)](#) proved one fundamental relationship in all queues in Eq. (C.5).

$$L = \lambda W \quad (\text{C.5})$$

The proof, in short, is to express the average number of users per time as $L = \frac{l[\tau]}{\tau}$, which can be further expanded to $\frac{l[\tau]}{\tau} = \frac{a[\tau]}{\tau} \frac{l[\tau]}{a[\tau]}$. The first part $\frac{a[\tau]}{\tau}$ is the arrival rate λ ; the second part $\frac{l[\tau]}{a[\tau]}$ is the average wait time W .

We can examine the simplest queue, M/M/1 as a first-come-first-serve (FCFS) discipline, using [Larson and Odoni \(1981\)](#) as reference. This type of queue is also called a birth-and-death process, a type of continuous time Markov process in which the state variable is a number ranging from zero to infinity. The state transition diagram is illustrated in Fig. C.2. It shows that at state n , the next state is either at $n+1$ or $n-1$.

In the case of M/M/1, the $\lambda_i = \lambda_j = \lambda$ and the service rates are similarly identical across all states, $\mu_i = \mu_j = \mu$. We want to obtain a steady-state probability of the system being in a specific state. First we acknowledge that the conditional probability is to increase or decrease by one unit is either $\lambda\Delta t$ or $\mu\Delta t$ due to the nature of Poisson processes (in an increment in time only one event can be processed, with a probability of occurrence equal to the rate). Then we can get an expression for there to be no transition occurring by the next time step.

$$P[N[t + \Delta t] = n | N[t] = n] = 1 - \lambda\Delta t - \mu\Delta t$$

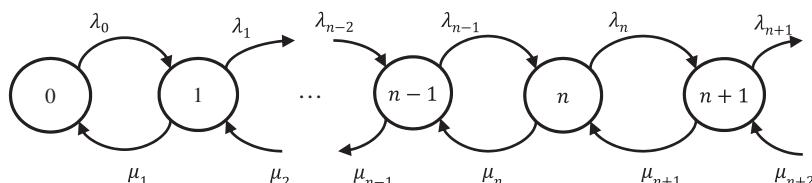


Fig. C.2 State transition diagram of a birth-and-death process, of which the M/M/1 model is one.

For small Δt and using $P_n(t) = P[N(t)=n]$ we can setup a balance equation at state n . At $t + \Delta t$, the state n can only occur if the prior state was n and it did not change, or if the prior state was $n + 1$ and one user was served, or if it was $n - 1$ and one user arrived. This is expressed as follows.

$$P_n(t + \Delta t) = P_{n+1}(t)\mu\Delta t + P_n(t)[1 - (\lambda + \mu)\Delta t] + P_{n-1}(t)\lambda\Delta t$$

Dividing by Δt and taking the limit as $\Delta t \rightarrow 0$, we get the following expression in Eq. (C.6).

$$\frac{dP_n(t)}{dt} = P_{n+1}(t)\mu - (\lambda + \mu)P_n(t) + P_{n-1}(t)\lambda \quad (\text{C.6})$$

We can use the boundary condition. When $n=0$:

$$\frac{dP_0(t)}{dt} = P_1(t)\mu - \lambda P_0(t)$$

Since we are evaluating the steady state, we can set $\frac{dP_0(t)}{dt} = 0$ and remove the time subscripts.

$$\mu P_1 = \lambda P_0$$

A further balance under steady-state equilibrium requires that any cross section between two states is stable:

$$\lambda P_n = \mu P_{n+1}, \text{ for } n = 0, 1, 2, 3 \dots$$

These are called *balance equations*. By substituting the terms recursively, it leads to Eq. (C.7).

$$P_n = \frac{\lambda^n}{\mu^n} P_0 = K_n P_0 \quad (\text{C.7})$$

We know that sum of probabilities has to be equal to 1:

$$\sum_{n=0}^{\infty} P_n = \left(1 + \sum_{n=1}^{\infty} K_n\right) P_0 = 1$$

$$P_0 = \frac{1}{1 + \left(\frac{\lambda}{\mu}\right) + \left(\frac{\lambda}{\mu}\right)^2 + \dots}$$

The denominator is just a geometric series where:

$$1 + ar + ar^2 + ar^3 + \dots = \frac{a}{1 - r}$$

Substituting that in, we get Eq. (C.8).

$$P_0 = \frac{1}{\left(\frac{1}{1 - \frac{\lambda}{\mu}} \right)} = 1 - \frac{\lambda}{\mu} = 1 - \rho \quad (\text{C.8a})$$

$$P_n = \rho^n (1 - \rho) \quad (\text{C.8b})$$

The other steady-state measures can be derived (see [Hillier et al., 2012](#)) and shown in Eq. (C.9). Note that essentially if we know L_q we can obtain the rest: $W_q = \frac{L_q}{\lambda}$, $W = W_q + \frac{1}{\mu}$, $L = L_q + \frac{\lambda}{\mu}$.

$$L = \sum_{n=0}^{\infty} n P_n = \frac{\rho}{1 - \rho} = \frac{\lambda}{\mu - \lambda} \quad (\text{C.9a})$$

$$W = \sum_{n=0}^{\infty} \frac{n+1}{\mu} P_n = \frac{1}{\mu(1 - \rho)} = \frac{1}{\mu - \lambda} \quad (\text{C.9b})$$

$$L_q = L - (1 - P_0) = \frac{\rho^2}{1 - \rho} = \frac{\lambda^2}{\mu(\mu - \lambda)} \quad (\text{C.9c})$$

$$W_q = \sum_{n=0}^{\infty} \frac{n}{\mu} P_n = \frac{\rho}{\mu(1 - \rho)} = \frac{\lambda}{\mu(\mu - \lambda)} \quad (\text{C.9d})$$

The probability density function for the wait time is shown in Eq. (C.10).

$$f_w[t] = (\mu - \lambda) e^{-(\mu - \lambda)t}, \text{ for } t \geq 0 \quad (\text{C.10})$$

To illustrate, if a fueling station has 10 vehicles arriving per hour and the average service time is 5 min, then the probability to be served in less than 2 min is:

$$F_W[0.033] = 1 - e^{-(12-10)(0.033)} = 6.4\%$$

The average queue length is $L_q = \frac{10^2}{12(2)} = 4.17$ vehicles.

For M/M/ m queues with more than one server, where $\rho = \frac{\lambda}{m\mu} < 1$, the birth-and-death process can still be used to evaluate the state probabilities and wait time as well. The state probabilities are shown in Eq. (C.11). The intuition with the fragmented terms is due to the probabilities being different when the number of servers is not all busy versus when they are.

$$P_0 = \left[\sum_{n=0}^{m-1} \frac{\left(\frac{\lambda}{\mu}\right)^n}{n!} + \frac{\left(\frac{\lambda}{\mu}\right)^m}{m!} \left(\frac{1}{1 - \frac{\lambda}{m\mu}} \right) \right]^{-1} \quad (\text{C.11a})$$

$$P_n = \begin{cases} \frac{\left(\frac{\lambda}{\mu}\right)^n}{n!} P_0, & 0 \leq n \leq m \\ \frac{\left(\frac{\lambda}{\mu}\right)^n}{m! m^{n-m}} P_0, & n \geq m \end{cases} \quad (\text{C.11b})$$

The average length of queue and CDF of the wait time are shown in Eq. (C.12).

$$L_q = \frac{P_0 \left(\frac{\lambda}{\mu}\right)^m \rho}{m! (1-\rho)^2} \quad (\text{C.12a})$$

$$F_W(t) = 1 - e^{-\mu t} \left[1 + \frac{P_0 \left(\frac{\lambda}{\mu}\right)^m}{m! (1-\rho)} \left(\frac{1 - e^{-\mu t \left(m-1 - \frac{\lambda}{\mu}\right)}}{m-1 - \frac{\lambda}{\mu}} \right) \right], \quad t \geq 0 \quad (\text{C.12b})$$

In the earlier example with the fuel station, if an extra pump is added, the average queue length drops from 4.17 vehicles down to

$$L_q = \frac{(0.4118) \left(\frac{10}{12}\right)^2 (0.4167)}{2(1-0.4167)^2} = 0.1751 \text{ vehicles.}$$

In the case where there is a finite capacity k where additional users arriving would balk, and assuming $m \leq k$, the state probabilities are shown in Eq. (C.13).

$$P_0 = \left[\sum_{n=0}^m \frac{\left(\frac{\lambda}{\mu}\right)^n}{n!} + \frac{\left(\frac{\lambda}{\mu}\right)^s}{m!} \sum_{n=m+1}^k \left(\frac{\lambda}{m\mu}\right)^{n-m} \right]^{-1} \quad (\text{C.13a})$$

$$P_n = \begin{cases} \frac{\left(\frac{\lambda}{\mu}\right)^n}{n!} P_0, & 0 \leq n \leq m \\ \frac{\left(\frac{\lambda}{\mu}\right)^n}{m! m^{n-m}} P_0, & m \leq n \leq k \end{cases} \quad (\text{C.13b})$$

The steady-state performance measures are shown in Eq. (C.14). Because of the capacity, the value of L is not obtained in the same way as earlier. The wait times W_q , W are obtained using an effective arrival rate $\bar{\lambda}$ shown in Eq. (C.14c).

$$L_q = \frac{P_0 \left(\frac{\lambda}{\mu} \right)^m \rho}{m! (1 - \rho)^2} [1 - \rho^{k-m} - (k-m)\rho^{k-m}(1-\rho)] \quad (\text{C.14a})$$

$$L = \sum_{n=0}^{m-1} n P_n + L_q + m \left(1 - \sum_{n=0}^{m-1} P_n \right) \quad (\text{C.14b})$$

$$\bar{\lambda} = \sum_{n=0}^{\infty} \lambda_n P_n = \lambda (1 - P_k) \quad (\text{C.14c})$$

For the fueling station example, a comparison can be made between an M/M/1 design and an M/M/1/1 design in Fig. C.3.

Many queues do not operate with exponentially distributed service times. For example, a single shuttle mobility-on-demand service may have exponential user arrivals but service time may not fit an exponential distribution. M/G/1 queues fall under this category. If the mean service time $1/\mu$ and its variance σ^2 can be observed, then $P_0 = 1 - \rho$. For the queue length there is a formula derived by Pollaczek-Khintchine (P-K) shown in Eq. (C.15).

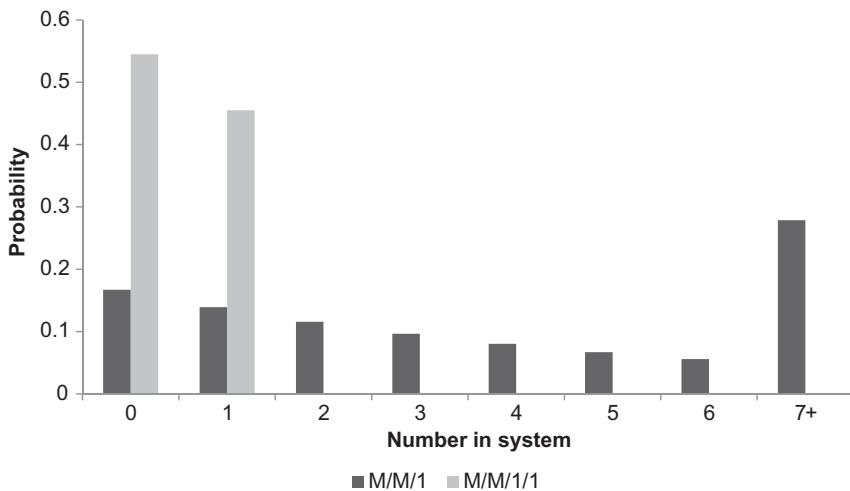


Fig. C.3 Comparison of the state probabilities for M/M/1 and M/M/1/1 queues.

$$L_q = \frac{\lambda^2 \sigma^2 + \rho^2}{2(1 - \rho)} \quad (\text{C.15})$$

The other performance measures can be obtained similarly to the M/M/ m queue.

C.3 Spatial Queues and Queue Networks

In an urban transport context, queues typically evolve over a space or a network. With spatial queues, [Larson and Odoni \(1981\)](#) describe a *hypercube* model to evaluate a set of m servers over a space. A simpler model can be deployed when it is just one server under some simplifying assumptions for a last mile application. Consider the rectangular region shown in [Fig. C.4](#).

In the single shuttle case, we can make some simplifying assumptions: user arrivals follow a Poisson process with λ and are uniformly distributed over the rectangular region, a vehicle starts from the depot to pickup a passenger and returns to the depot to complete the service. We assume dwell time Z at the location of the user pickup is independently distributed as $f_Z[z]$. Then this is simply an M/G/1 queue and its performance measures can be determined if the mean and variance of the service rate can be estimated.

Let us define the service time S as a function of the random distance to the location divided into D_x, D_y for the x - and y -coordinates. Then the service time is shown in Eq. [\(C.16\)](#).

$$S = 2 \left(\frac{D_x}{v_x} + \frac{D_y}{v_y} \right) + Z \quad (\text{C.16})$$

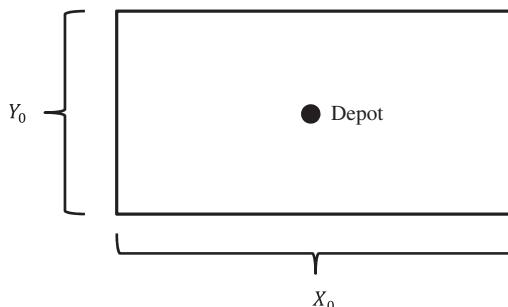


Fig. C.4 Region for a spatial queue analysis with a single shuttle based at a central depot.

D_x is a function of the uniformly distributed location X between $[0, X_0]$ and ν_x, ν_y are speeds. The cumulative probability of D_x can be defined as the following joint probability:

$$P(D_x \leq y) = \left(X - \frac{X_0}{2} \leq y \right) \cap \left(\frac{X_0}{2} - X \leq y \right)$$

The CDF is $F_{D_x}(y) = \frac{2y}{X_0}$, $0 \leq y \leq \frac{X_0}{2}$, and taking the derivative we get $f_{D_x}(y) = \frac{2}{X_0}$. Knowing the PDF we can derive the first and second moments:

$$E[D_x] = \int_0^{X_0/2} \frac{2y}{X_0} dy = \frac{X_0}{4}$$

$$V[D_x] = E[D_x^2] - (E[D_x])^2 = \frac{X_0^2}{12} - \frac{X_0^2}{16} = \frac{X_0^2}{48}$$

The y -axis distance moments are analogously constructed. This leads to the following values of the mean service time and variance in Eq. (C.17), where \bar{z} is the mean dwell time and σ_Z^2 is its variance.

$$\frac{1}{\mu} = E[S] = 2 \left(\frac{X_0}{4\nu_x} + \frac{Y_0}{4\nu_y} \right) + \bar{z} \quad (\text{C.17a})$$

$$\sigma_S^2 = 4 \left(\frac{\sigma_{D_x}^2}{\nu_x^2} + \frac{\sigma_{D_y}^2}{\nu_y^2} \right) + \sigma_Z^2 = \frac{1}{12} \left(\frac{X_0^2}{\nu_x^2} + \frac{Y_0^2}{\nu_y^2} \right) + \sigma_Z^2 \quad (\text{C.17b})$$

As an example, consider a fire dispatch problem where a single fire truck serves a 20-block by 20-block neighborhood. Service requires going to the site of the fire, clearing out the fire, and returning to depot before heading out for the next job. The speed of transport is 10 mph–200 blocks/h. Time at the site is normally distributed with mean of 20 min and standard deviation of 5 min. Fires occur at a rate of one per hour as a Poisson process.

The average service time is:

$$\frac{1}{\mu} = E[S] = 2 \left(\frac{20}{4(200)} + \frac{20}{4(200)} \right) + \left(\frac{1}{3} \right) = 0.433 \text{ h} \rightarrow \mu = 2.308/\text{h}$$

Variance of service time is:

$$\sigma_S^2 = \frac{1}{12} \left(\frac{20^2}{200^2} + \frac{20^2}{200^2} \right) + \left(\frac{1}{12} \right)^2 = 0.008611 \text{ h}^2$$

Now we can apply the P-K formula:

$$L_q = \frac{\lambda^2 \sigma^2 + \rho^2}{2(1-\rho)} = \frac{1^2(0.008611) + 0.4333^2}{2(1 - 0.4333)} = 0.1732$$

$$W_q = \frac{L_q}{\lambda} = 0.1732 \text{ h} = 10.39 \text{ min before response}$$

The L_q suggests that at any time there is 0.1732 of a fire that is unattended. The W_q suggests a response time on average of 10–11 min before the fire truck even starts driving toward the site of the fire.

The last topic in queueing analysis is considering multiple queues that feed each other in a network. For example, consider the network shown in Fig. C.5. In this network, there are three queues and inflows a_j into the network at those queues are j . For example, there is one user arriving per minute to queue no. 1. After completing a queue, passengers are distributed to other queues or to exit the network. When the queues within this network operate as M/M/m queues, the problem is called a *Jackson network*, named after Jackson (1957).

Under steady state, each queue in such a network behaves as an independent M/M/m queue with arrival rate shown in Eq. (C.18) if $m_j \mu_j > \lambda_j$. A set of balance equations results from the equation, which are used to solve for

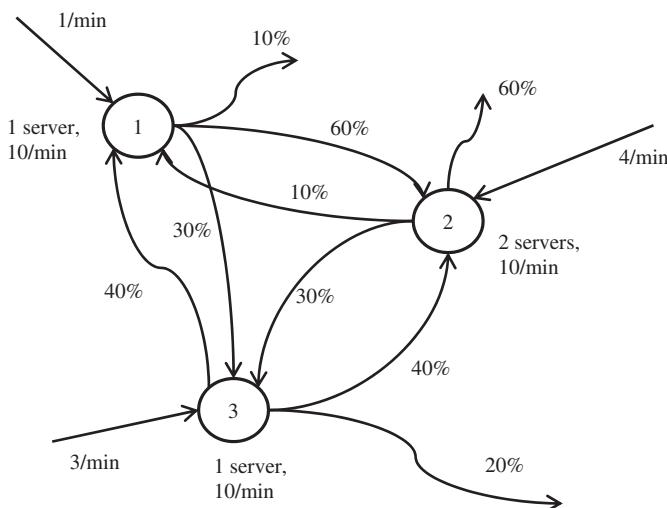


Fig. C.5 Example of a queue network.

the effective λ_j at each node. Once those are known, the standard performance measures are determined independently.

$$\lambda_j = a_j + \sum_{i=1}^m \lambda_i p_{ij} \quad (\text{C.18})$$

In addition to marginal probabilities of the individual queues, joint probabilities (e.g., probability of having two customers at node 1, one customer at node 2, and three or more customers at node 3) can be obtained as a product of independent probabilities. The expected number of users in the network is obtained by adding up all the average L_j 's. The expected wait time in the network is obtained using Little's formula in Eq. (C.19).

$$W = \frac{L}{\sum a_i} \quad (\text{C.19})$$

To illustrate, consider the example in Fig. C.5. The following balance equations are solved to determine the effective arrival rates to each node, which is $\lambda = (I - P^T)(a)$ in matrix form.

$$\lambda_1 = 1 + 0.1\lambda_2 + 0.4\lambda_3$$

$$\lambda_2 = 4 + 0.6\lambda_1 + 0.4\lambda_3$$

$$\lambda_3 = 3 + 0.3\lambda_1 + 0.3\lambda_2$$

Solving the balance equations we get $\lambda = [5, 10, 7.5]$. Now all the properties of the individual queues and the network can be derived. For example, the probability that all servers are idle at the same time is:

$$P(0, 0, 0) = \frac{1}{2} \left(\frac{1}{3}\right) \left(\frac{1}{4}\right) = \frac{1}{24}$$

The average number of users in the network requires first finding each of the number of users in the queues.

$$L_1 = \frac{\lambda_1}{\mu_1 - \lambda_1} = \frac{5}{10 - 5} = 1$$

$$L_2 = \frac{P_2(0) \left(\frac{\lambda_2}{\mu_2}\right)^s \rho_2}{s! (1 - \rho_2)^2} + \frac{\lambda_2}{\mu_2} = \frac{\left(\frac{1}{3}\right) \left(\frac{10}{20}\right)^2 \left(\frac{10}{20}\right)}{2 \left(1 - \frac{10}{20}\right)^2} + \frac{10}{10} = \frac{4}{3}$$

$$L_3 = \frac{\lambda_3}{\mu_1 - \lambda_1} = \frac{7.5}{10 - 7.5} = 3$$

Summing them up we get $L = \frac{16}{3} = 5.333$. Lastly, we can determine the expected wait time going through the network as:

$$W = \frac{L}{a_1 + a_2 + a_3} = \frac{16}{3(1+4+3)} = \frac{2}{3} \text{ min}$$

These are only wait times at those queues and do not include the deterministic travel times between queues.

D DISCRETE CHOICE MODELING

D.1 Consumer Theory

Discrete choice models are used to explain the distribution of behavior of consumers of alternative products or services. As a model of consumer behavior, we first discuss consumer theory. Five assumptions are made about how consumers treat goods (Kanafani, 1983).

- (1) Each consumer has a choice on quantity of different products consumed.
- (2) Every good possess characteristics that give satisfaction, that is, utility, to a consumer. Utility is therefore derived from the characteristics of goods, not the goods themselves.
- (3) Each consumer has consistent preference structure that is transitive. If one prefers A over B, and B over C, then they also prefer A over C.
- (4) Consumers are insatiable: more is always better than less.
- (5) Choice is limited by a budget constraint due to resource expenditures needed to consume. These expenditures may be money, time in a day, and so on.

Under these assumptions, consumers are presented with a bundle of m goods $X = \{x_1, \dots, x_m\}$ in which they compare to another bundle of goods Y . If bundle X is preferred over bundle Y , $X \succ Y$. Consumers are assumed to choose the bundle that is most preferred. The quantity of satisfaction for a bundle is measured with utility, a unitless measure of preference. If $X \succ Y$, then $U[X] > U[Y]$.

Mathematical operations can be applied to utility. It is a relative measure in the sense that if $U[X] > U[Y]$ then $U[X] + C > U[Y] + C$ and $\gamma U[X] > \gamma U[Y]$. The amount consumed is bound by the resource expenditure constraint. The utility that an individual can earn from a bundle can be represented by *isouility curves*. An example is shown in Fig. D.1. All points on one curve would earn the individual the same amount of utility. Under a fixed resource expenditure budget represented budget, the maximum utility is obtained from the isouility curve that just touches

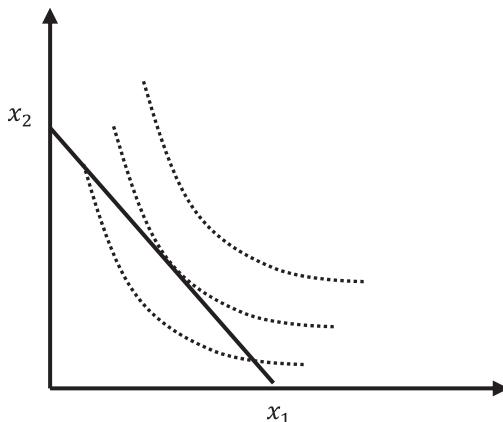


Fig. D.1 Illustration of isouutility curves and resource expenditure budget (straight line).

the budget. Isouutility curves are generally convex; if they are concave, it would mean the individual prefers consuming only one good over the others.

A common functional form to fit to an isouutility curve is the *Cobb-Douglas* function in Eq. (D.1).

$$U[x_1, \dots, x_m] = bx_1^{a_1} \dots x_m^{a_m} \quad (\text{D.1})$$

If an individual's utility for different products can be calibrated, then trade-offs can be analyzed using marginal utility in Eq. (D.2a) and marginal rate of substitution in Eq. (D.2b).

$$MU_i = \frac{\partial U}{\partial x_i} \quad (\text{D.2a})$$

$$MRS_{i,j} = \frac{MU_i}{MU_j} \quad (\text{D.2b})$$

Changes in consumption patterns can be explained as the sum of two effects. One is the *substitution effect* which captures the change in consumption mix under the same isouutility curve. The second is the *income effect* which captures the change in isouutility curve under the same consumption mix. The latter effect reflects a change in an individual's spending capacity.

An individual's demand function can then be derived from the isouutility curves. Consider the case of two products. The demand function for a single product X_1 can be determined by holding the second quantity X_2 fixed and varying the price. For a fixed budget B , there is an optimal quantity Q_i corresponding to each price P_i . When the prices and quantities of the product

are plotted the result is the individual's demand curve for product X_1 . This demand curve accounts for the individual's preferences for other goods under a fixed budget (Fig. D.2).

If all the individual demand curves are known, the market demand is simply the sum of all the individual functions. Classical demand analysis uses observed market quantities sold and prices along with other independent population variables to estimate market demand functions. However, this approach does not capture the underlying individual behaviors.

D.2 Random Utility Models: Multinomial Logit

An alternative approach is based on the *random utility model* (see Manski, 1977), which acknowledges that there are observable and unobservable portions of individual decision-making. By modeling the unobservable portion as a part of a population distribution for the choice, it is possible to construct a stochastic model of individual utility.

Consider a set C of J choices from which each member n of a population of N individuals selects an alternative $i \in C_n$ (where $C_n \subseteq C$ is a subset of

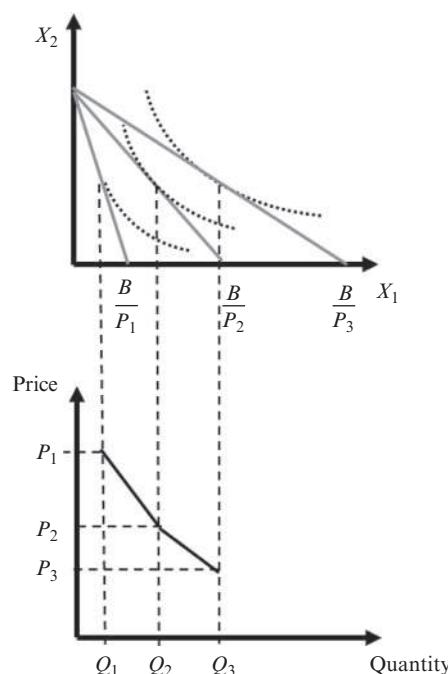


Fig. D.2 Derivation of individual demand function from utility curves.

choices available to n) that maximizes their utility U_{in} . The utility function is specified as shown in Eq. (D.3), where X_{in} is a vector of observable attributes related to alternative i , and V_{in} is a function (typically linear in parameters) of the observable attributes capturing the *representative utility*. The variate ε_{in} is a disturbance specific to individual n for alternative i and is assumed to be i.i.d. with respect to other individuals. The attributes X_{in} may be further categorized into alternative specific attributes (e.g., travel cost for a mode being chosen) and individual-specific attributes (e.g., individual's income).

$$U_{in} = V_{in}[X_{in}] + \varepsilon_{in} \quad (\text{D.3})$$

An individual n is modeled to choose i over j if $U_{in} > U_{jn}$. A utility function may look like the following:

$$U_{in} = \beta_{i0} + \beta_{i1}X_{in1} + \dots + \beta_{ik}X_{ink} + \varepsilon_{in}$$

Since utilities are relative and choice only matters regardless of the difference, only $J - 1$ alternatives have an alternative-specific constant. The probability of an individual choosing an alternative is then expressed in Eq. (D.4).

$$P_n[i | C_n] = \Pr \left[U_{in} \geq \max_j U_{jn}, \forall j \in C_n \right] \quad (\text{D.4})$$

For certain distributions of ε_{in} , it turns out that the expression in Eq. (D.4) is conveniently structured. McFadden (1974) showed that when ε_{in} follows a Gumbel distribution, a Type 1 extreme value distribution, the result leads to simplified expression. A Gumbel distribution has a PDF and CDF as shown in Eq. (D.5), where η is a location parameter and μ is a positive scale parameter. The mean is $\eta + \gamma/\mu$, where γ is the Euler constant (~ 0.577), and the variance is $\frac{\pi^2}{6\mu^2}$.

$$f[\varepsilon] = \mu e^{-\mu(\varepsilon-\eta)} \exp \left[-e^{-\mu(\varepsilon-\eta)} \right] \quad (\text{D.5a})$$

$$F[\varepsilon] = \exp \left[-e^{-\mu(\varepsilon-\eta)} \right], \mu > 0 \quad (\text{D.5b})$$

Gumbel variates exhibit several features. First, if ε is a Gumbel variate, and α and V are scalar constants, then an affine function $\alpha\varepsilon + V$ is Gumbel with parameters $(\alpha\eta + V, \frac{\mu}{\alpha})$. As an extreme value distribution, Gumbel distribution exhibits a feature like the Weibull distribution. If ε_{in} and ε_{jn} are independent Gumbel variates with parameters (η_i, μ) and (η_j, μ) , the distribution of $\max[\varepsilon_{in}, \varepsilon_{jn}]$ is also Gumbel distributed, with parameters

$\left(\frac{1}{\mu} \ln(e^{\mu \eta_i} + e^{\mu \eta_j}), \mu\right)$. The relationship is extendable to more than two variates: the max of J of Gumbel variates is also a Gumbel variate with parameters $\left(\frac{1}{\mu} \ln\left(\sum_{j \in J} e^{\mu \eta_j}\right), \mu\right)$. This means that the max expression in Eq. (D.4) is Gumbel distributed since the utilities are Gumbel variates as affine functions of ε . Finally, if $\varepsilon_n = \varepsilon_{jn} - \varepsilon_{in}$ for two independent Gumbel variates, then ε_n is logically distributed as shown in Eq. (D.6). This means in Eq. (D.4), the comparison of U_{in} and $\max_j U_{jn}$ can be represented as a logistic variate. Typically, the parameter β depends on μ , so we can simplify and set $\mu=1$ to estimate β relative to that value. With $\mu=1$, each Gumbel variate has a mean of $E[\varepsilon_{in}] = \gamma$ and standard deviation $std[\varepsilon_{in}] = \frac{\pi}{\sqrt{6}}$. The difference $\varepsilon_n = \varepsilon_{jn} - \varepsilon_{in}$ has mean of $E[\varepsilon_n] = 0$ and standard deviation $std[\varepsilon_n] = \frac{\pi}{\sqrt{3}}$.

$$F(\varepsilon_n) = \frac{1}{1 + e^{-\mu \varepsilon_n}} \quad (\text{D.6})$$

Domencich and McFadden (1975) demonstrate the derivation in more detail for linear-in-parameters utility functions, leading to Eq. (D.7) for the probability of an individual n choosing an alternative $i \in J_n$.

$$P_n[i] = \frac{e^{\beta x_{in}}}{\sum_{j \in C} e^{\beta x_{jn}}} = \frac{1}{1 + \sum_{j \in C, j \neq i} e^{\beta x_{jn} - \beta x_{in}}} \quad (\text{D.7})$$

Many practitioners use the term “logistic regression” synonymously with the MNL model. MNL model refers to a more generalized structure where each alternative’s utility function can be different. This was originally called a “conditional logit” by McFadden and is more generalized than logistic regression. Logistic regression only has one equation (U_n) so every alternative is required to have the same utility function. However, it is mechanically possible to use one utility function to incorporate all alternative-specific variables and to treat the nonalternative variables as zero. In this way, tools like the *mlogit* package (Croissant, 2012) in R can be used to estimate MNL models.

The MNL model is also convenient because of its similarity to the normal distribution but with fatter tails. To illustrate the difference, a binary choice example is modeled with three different distributions for ε_n : a uniform distribution, a normal distribution, and a logistic distribution. The result spanning over $[-10, 10]$ is shown in Fig. D.3. The logistic distribution has a distinct S-shape curve. Because of the shape, elasticities tend to be more

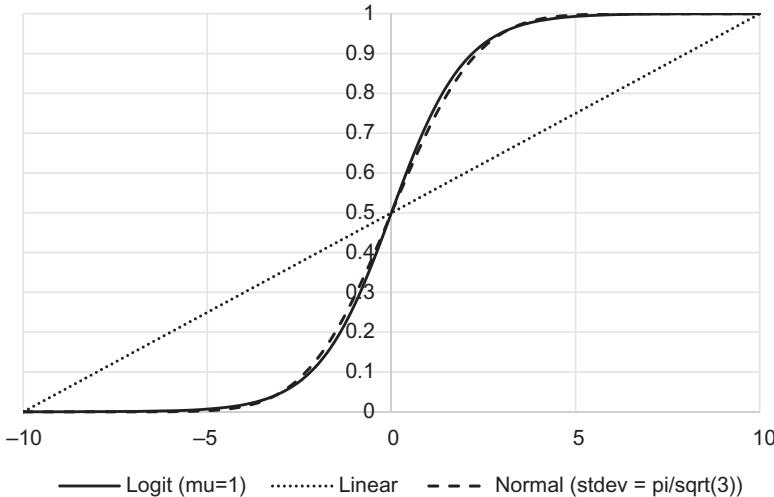


Fig. D.3 Comparison of CDFs of ε_n .

significant when the market share is closer to the 50 percentile. This can be illustrated with the derivation for the elasticity.

The elasticity of the probability of choosing an alternative with respect to an attribute x_{ink} is shown here for a binary logit example.

$$\begin{aligned}
 e_{P_n[i], x_{ink}} &= \frac{\partial P_n[i]}{\partial x_{ink}} \left(\frac{x_{ink}}{P_n[i]} \right), \text{ where } P_n[i] = \frac{1}{1 + \sum_{j \neq i} \exp(\beta_j x_{jn} - \beta_i x_{in})} \\
 &= \left[-\frac{\beta_{ik} \sum_{j \neq i} \exp(\beta_j x_{jn} - \beta_i x_{in})}{\left(1 + \sum_{j \neq i} \exp(\beta_j x_{jn} - \beta_i x_{in}) \right)^2} \right] \left(\frac{x_{ink}}{P_n[i]} \right) \\
 &= \beta_{ik} \left(\frac{x_{ink}}{P_n[i]} \right) \frac{\sum_{j \neq i} \exp(\beta_j x_{jn} - \beta_i x_{in}) + 1 - 1}{\left(1 + \sum_{j \neq i} \exp(\beta_j x_{jn} - \beta_i x_{in}) \right)^2} \\
 &= \beta_{ik} \left(\frac{x_{ink}}{P_n[i]} \right) (P_n[i] - P_n[i]^2)
 \end{aligned}$$

After simplifying, we get the following expression for direct elasticity with respect to an attribute in the alternative. This is shown to work for the general MNL case as well.

$$e_{P_n[i], x_{ink}} = \beta_{ik} x_{ink} (1 - P_n[i]) \quad (\text{D.8})$$

A cross elasticity of the probability of choosing alternative i with respect to change in an attribute x_{jnk} for alternative j can similarly be derived, leading to the expression in Eq. (D.9).

$$e_{P_n[i], x_{jnk}} = -P_n[j] \beta_{jk} x_{jnk} \quad (\text{D.9})$$

Marginal rates of substitution can also be computed in closed form especially with linear utility functions. For example, if a model is specified to have some measure of travel time (TVT_i) and cost ($Cost_i$), that is, $V_i = \dots + \beta_{TVT} TVT_i + \beta_{Cost} Cost_i + \dots$, then the value of time (VOT) in \$/h is as shown in Eq. (D.10).

$$VOT = \frac{\left(\frac{\partial V_i}{\partial Time_i} \right)}{\frac{\partial V_i}{\partial Cost_i}} = \frac{\beta_{TVT}}{\beta_{Cost}} \quad (\text{D.10})$$

While the MNL model has convenient closed form expressions for the probability and elasticities, it assumes that disturbances of each alternative for one individual are independent from each other. This leads to the fallacy of *Independence of Irrelevant Alternatives* (IIA). IIA can be illustrated with a class “Red Bus/Blue Bus” example.

Suppose there are two modes available: car and bus, each with equal representative utilities $V_{car} = V_{bus} = 1$. The market share is 50% each, and the ratio of mode choice is $\frac{P_n[car]}{P_n[bus]} = 1$. Due to IIA, this ratio remains the same regardless of what other alternatives are added. This means if we wanted to analyze a scenario where the buses are all painted red and we double the fleet with another set of blue buses that perform in the exact same way, then technically both sets of buses would have $V_{bluebus} = V_{redbus} = 1$. However, in that case we would end up with $P_n[car] = P_n[bluebus] = P_n[redbus] = 33.3\%$. This means that having essentially the same option in a different color reduces the mode share of “car” from 50% to 33.3% under the MNL model. Similar issues arise when modeling route choice when there are overlapping routes.

D.3 Random Utility Models: Probit, GEV, Mixed Logit

To address these issues, other types of random utility models have been proposed. A natural assumption is to use a normal distribution (called probit models). Probit models capture correlations between alternatives. For example, for the binary probit case, the error of the difference between two dimensions of a multivariate normal variate is distributed as in Eq. (D.11a), where σ_{12} is the correlation between the two dimensions. Then the probability of choosing alternative 1 over alternative 2 is shown in Eq. (D.11b).

$$\varepsilon_{21n} = \varepsilon_{2n} - \varepsilon_{1n} \sim N[0, \sigma_{22} + \sigma_{11} - 2\sigma_{12}] \quad (\text{D.11a})$$

$$P_n[1] = \Pr[\varepsilon_{21n} \leq V_{1n} - V_{2n}] = \Phi\left[\frac{V_{1n} - V_{2n}}{\sqrt{\sigma_{22} + \sigma_{11} - 2\sigma_{12}}}\right] \quad (\text{D.11b})$$

For more than two alternatives, however, the multinomial probit is troublesome to compute the probability $P_n[i]$ because we need to compute the $\Pr[\varepsilon_{j1n} \leq V_{1n} - \max V_{jn}]$. This requires solving multidimensional integrals where the outer integral is a function of the inner integrals when there are correlations. To overcome this challenge, approximation methods have been proposed like the Clark method (Daganzo et al., 1977). However, the method has been shown to be a very poor approximation in some cases. The more popular approach that has emerged in recent years with the support of increasing computational power is simulated integrals.

One approach attributed to [Geweke \(1989\)](#), [Hajivassiliou and McFadden \(1998\)](#), and [Keane \(1990\)](#) is the GHK simulator. To use the GHK simulator, first the Cholesky decomposition is conducted on the covariance matrix Ω_1 (utility differences from alternative 1) to obtain Cholesky factors L_1 , where $\Omega_1 = L_1 L_1^T$ and L_1 is a lower triangular matrix. Cholesky factorization in Matlab is done via the function `chol`. Then the utility functions can be converted into the following equivalent set:

$$\begin{aligned} \tilde{U}_{1in} &= \tilde{V}_{1in} + c_{11}\eta_1 \\ \tilde{U}_{2in} &= \tilde{V}_{2in} + c_{21}\eta_1 + c_{22}\eta_2 \\ \tilde{U}_{3in} &= \tilde{V}_{3in} + c_{31}\eta_1 + c_{32}\eta_2 + c_{33}\eta_3 \\ &\dots \end{aligned}$$

The advantage of this equivalent set is that the η_i disturbances are all independent standard normal variates now, and the utilities can be simulated

using a recursive approach. This approach can be used to simulate any multivariate normal distribution. Once this set of equivalent decomposed functions is obtained, the GHK simulator is run as shown in [Algorithm D.1](#).

Algorithm D.1 GHK Simulator (See [Train, 2009](#))

1. Calculate $\Pr\left[\eta_1 < -\frac{\tilde{V}_{1in}}{c_{11}}\right] = \Phi\left[-\frac{\tilde{V}_{1in}}{c_{11}}\right]$
2. Draw standard uniform μ'_1 , let $\eta'_1 = \Phi^{-1}\left[\mu'_1 \Phi\left[-\frac{\tilde{V}_{1in}}{c_{11}}\right]\right]$.
3. Calculate $\Phi\left[-\frac{\tilde{V}_{2in} + c_{21}\eta'_1}{c_{22}}\right]$.
4. Draw standard uniform μ'_2 , let $\eta'_2 = \Phi^{-1}\left[\mu'_2 \Phi\left[-\frac{\tilde{V}_{2in} + c_{21}\eta'_1}{c_{22}}\right]\right]$.
5. Calculate $\Phi\left[-\frac{\tilde{V}_{3in} + c_{31}\eta'_1 + c_{32}\eta'_2}{c_{33}}\right]$.
6. Repeat for all alternatives except i .
7. Simulated probability of run r is

$$\check{P}_{in}^r = \Phi\left[-\frac{\tilde{V}_{1in}}{c_{11}}\right] \times \Phi\left[-\frac{\tilde{V}_{2in} + c_{21}\eta'_1}{c_{22}}\right] \times \Phi\left[-\frac{\tilde{V}_{3in} + c_{31}\eta'_1 + c_{32}\eta'_2}{c_{33}}\right] \times \dots$$
8. After R simulated runs, the simulated probability is:

$$\check{P}_{in} = \frac{1}{R} \sum_r \check{P}_{in}^r$$

For those desiring a more closed form expression, certain dependency structures can be modeled using more generalized extreme value (GEV) distributions. This can be done by clustering alternatives into multiple dimensions. For example, the red bus/blue bus problem could be addressed by splitting the alternatives into two dimensions: the first is across type of mode (car, bus) and the second is by color of bus (red, blue). The choices are then joint 2-dimensional choices with the following choice set: {(car), (bus, red), (bus, blue)}. The utility functions can be modeled with correlations using a nest structure shown in [Fig. D.4](#).

The overall utility function structure of one of the lower branch alternatives is specified as shown:

$$U_{bus,n} = V_{bus} + \frac{1}{\mu_{color}} \ln \left(e^{\mu_{color} V_{(red|bus)}} + e^{\mu_{color} V_{(blue|bus)}} \right) + \varepsilon_{bus,n}$$

$$U_{red|bus,n} = V_{(red|bus)} + \varepsilon_{red bus,n}$$

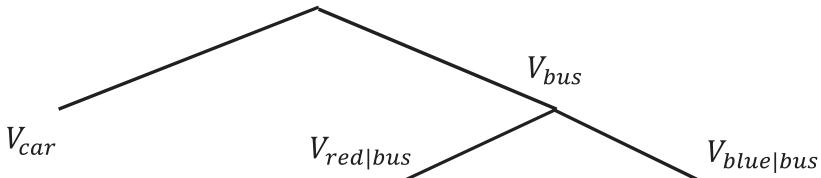


Fig. D.4 Nested logit example.

The variable V_{bus} is used to capture bus attributes that do not pertain to the color differences. The variable μ_{color} is a scaling factor to balance the significance of the differences at the lower level to the upper level. When $\mu_{color}=1$ it suggests the color differences are just as important as the other characteristics of the bus relative to car (and flattens the nest into an MNL). The term $\frac{1}{\mu_{color}} \ln(e^{\mu_{color} V_{(red|bus)}} + e^{\mu_{color} V_{(blue|bus)}})$ is the expected utility of the having the options red and blue to choose from (for Gumbel distribution the expectation of the maximum has this form).

To illustrate this model, consider the same example with $V_{car}=V_{bus}=1$, but assume $\mu_{color}=1.386$ and $V_{blue|bus}=V_{red|bus}=-0.5$. In this case, $\frac{1}{\mu_{color}} \ln(e^{\mu_{color} V_{(red|bus)}} + e^{\mu_{color} V_{(blue|bus)}}) = 0$ so the colors do not contribute to the choice of bus versus car. The percent of car remains 50%, bus is also 50%, and given bus, the distribution between red and blue is equally distributed. Then the overall probability of choosing a red bus is 25%.

In many cases the tastes of a population for different attributes, captured by β , are not homogeneous. MNL ignores these heterogeneous characteristics by accounting for differences in the disturbance term ϵ_{in} . However, if a population tends to exhibit more heterogeneous taste variation in one attribute than another, then the MNL model would fail to model that.

A mixed logit model is a flexible model structure where both the disturbance and the β parameters are random with probability density function $f[\beta|\theta]$ in terms of distribution parameters θ . The discrete analog of the mixed logit model is the latent class model. When the disturbance is Gumbel distributed it leads to the mixed multinomial logit model. Different distributions for β can be used, although the method discussed here assume a multivariate normal distribution across multiple β_{ink} variables with correlations. The probability of choosing an alternative is then determined with Eq. (D.12).

$$P_{in} = \int_{-\infty}^{\infty} \left(\frac{e^{\beta' x_{in}}}{\sum_j e^{\beta' x_{jn}}} \right) f[\beta | \theta] d\beta \quad (\text{D.12})$$

As with the multinomial probit model, computation of the integral can get complicated with multidimensional β , especially with correlations. A simulated probability approach is shown in Train (2009) (Algorithm D.2).

Algorithm D.2 Simulated Probability of Mixed Multinomial Logit Model With Normal Distributed Parameters (Train, 2009)

Given $\beta = \{\beta_1, \dots, \beta_K\}$ with $K \times 1$ mean vector b , $K \times K$ covariance matrix W :

1. Determine Cholesky matrix L corresponding to W
2. Simulate R samples
 - (a) For each sample r , generate $\eta_1^r, \dots, \eta_K^r$ as independent standard normal using Halton draws
 - (b) Assign $\beta_1^r = b_1 + L[1, 1]\eta_1^r, \beta_2^r = b_2 + L[2, 1]\eta_1^r + L[2, 2]\eta_2^r, \dots, \beta_K^r = b_K + L[K, 1]\eta_1^r + \dots + L[K, K]\eta_K^r$
 - (c) Let $P_{in}^r = \frac{e^{(\beta^r)' x_{in}}}{\sum_j e^{(\beta^r)' x_{jn}}}$
3. Simulated probability is: $\check{P}_n[i] = \frac{1}{R} \sum_r P_{in}^r$

Halton draws are used to be more statistically efficient in capturing the parameter space in less samples. Further studies have been done to show that mixed logit can approximate *any* random utility model (McFadden and Train, 2000). Srinivasan and Mahmassani (2005) verified this with approximating probit because the parameters can essentially be scaled up to remove the effect of the Gumbel disturbance in the mixed MNL model.

D.4 Estimation

The MNL model can be estimated using maximum likelihood estimation. Given N observations, the likelihood function is shown in Eq. (D.13a), where y_{in} is a binary observation equal to one if n is observed to choose i , and zero otherwise. The log-likelihood in Eq. (D.13b) is a more tractable form to maximize.

$$L = \prod_{n=1}^N \prod_{i \in C_n} P_n[i]^{y_{in}} \quad (\text{D.13a})$$

$$LL = \sum_{n=1}^N \sum_{i \in C_n} \gamma_{in} \ln P_n[i] \quad (\text{D.13b})$$

Optimal parameters are obtained by using first-order conditions shown in Eq. (D.14). This is a system of equations that can be solved iteratively using Newton-Raphson method (see Ben-Akiva and Lerman, 1985).

$$\begin{aligned} \frac{\partial LL}{\partial \beta_k} &= \sum_{n=1}^N \sum_{i \in C_n} \gamma_{in} \left(x_{ink} - \frac{\sum_{j \in C_n} e^{\beta' x_{jn}} x_{jnk}}{\sum_{j \in C_n} e^{\beta' x_{jn}}} \right) = \sum_{n=1}^N \sum_{i \in C_n} (\gamma_{in} - P_n[i]) x_{ink} \\ &= 0, \quad \forall k \end{aligned} \quad (\text{D.14})$$

The Hessian can be determined as Eq. (D.15a), where each element is obtained using Eq. (D.15b).

$$H = \begin{pmatrix} \frac{\partial^2 LL}{\partial \hat{\beta}_1 \partial \hat{\beta}_1} & \dots & \frac{\partial^2 LL}{\partial \hat{\beta}_1 \partial \hat{\beta}_k} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 LL}{\partial \hat{\beta}_k \partial \hat{\beta}_1} & \dots & \frac{\partial^2 LL}{\partial \hat{\beta}_k \partial \hat{\beta}_k} \end{pmatrix} \quad (\text{D.15a})$$

$$\frac{\partial^2 LL}{\partial \hat{\beta}_k \partial \hat{\beta}_l} = - \sum_{n=1}^N \sum_{i \in C} P_n[i] \left(x_{ink} - \sum_{j \in C_n} x_{jnk} P_n[j] \right) \left(x_{ink} - \sum_{j \in C_n} x_{jnl} P_n[j] \right) \quad (\text{D.15b})$$

The Hessian is used to determine the standard errors of the estimated β_k 's. For example, the covariance of the β_k 's is shown in Eq. (D.16a), which is used to determine the standard errors in Eq. (D.16b).

$$\text{Cov}[\hat{\beta}] = -H^{-1} \quad (\text{D.16a})$$

$$\text{s.e.} \hat{\beta}_k = \sqrt{\text{Cov}[\hat{\beta}]_{kk}} \quad (\text{D.16b})$$

Two statistical tests can be conducted on the estimated parameters to evaluate their statistical significance. The first is a t -test to determine the P -value of statistical significance relative to a null hypothesis of $\hat{\beta}_k = 0$ in Eq. (D.17a). A P -value of .05 or lower is desired, although in combination with some other variables it might be beneficial to leave a parameter in even with slightly higher P -values. The other test is a likelihood ratio test with a Chi Square test using the test statistic shown in Eq. (D.17b). $LL[\hat{\beta}]$ is the

log-likelihood of the model with the estimated parameter(s). $LL[\hat{\beta}^H]$ is the log-likelihood of the model with the parameters set to zero.

$$\text{Test statistic : } \frac{\hat{\beta}_k}{\text{s.e.}\hat{\beta}_k} \quad (\text{D.17a})$$

$$\text{Test statistic : } 2(LL[\hat{\beta}] - LL[\hat{\beta}^H]) \quad (\text{D.17b})$$

The fitness of a model based on the estimated parameters is measured using “McFadden’s R^2 ,” or the ρ^2 . There are two forms of ρ^2 in Eq. (D.18).

$$\text{Version 1 : } \rho^2 = 1 - \frac{LL[\beta]}{LL[0]} \quad (\text{D.18a})$$

$$\text{Version 2 : } \rho^2 = 1 - \frac{LL[\beta]}{LL[c]} \quad (\text{D.18b})$$

In R, for example, the second version is computed, whereas in BIO-GEME it is the first version. To account for large numbers of parameters (using version 1), one can compute the adjusted ρ^2 as shown in Eq. (D.19).

$$\bar{\rho}^2 = 1 - \frac{LL[\beta] - K}{LL[0]} \quad (\text{D.19})$$

D.5 Aggregation, Segmentation, and Revenue Management

For the MNL, the expected value of the maximum utility can be used for economic interpretation. It is also called a “measure of accessibility” as it provides a scalar quantity for the expected worth of a set of alternatives. By itself, the measure is rather meaningless. However, [Small and Rosen \(1981\)](#) showed how to compute a *compensating variation* from a change in the welfare measure for an individual in Eq. (D.20).

$$\Delta CV = -\frac{1}{\lambda} \left[\ln \sum_{i \in C_n^2} e^{\mu V_{in}^2} - \ln \sum_{i \in C_n^1} e^{\mu V_{in}^1} \right] \quad (\text{D.20})$$

The λ is a marginal utility of with respect to a measure of cost in the context of the choice. For example, if using discrete choice to model car ownership, then λ should be with respect to income. In the case of choices made at a daily level for commuting, λ may be more appropriately linked to travel costs expended that day. The negative sign is meant to represent the willingness to pay for the change to the set of choices available to the user.

If there is an improvement in expected utility of the set of choices, then the individual is willing to *pay* (negative) ΔCV for the change.

With a way of measuring welfare improvements for individuals, there needs to be a way to aggregate these measures up to the market or population. For example, a model is developed to predict the preference for alternative i given input variables x_n : $P(i|x_n)$. Even then, if we estimated this model from 20,000 New Yorkers, how do we apply it to the 12M New Yorkers when we do not know the attributes x_n of every person in the population? If we know the density function of the choice probability across the population as $p(x)$, then the market share can be computed as in Eq. (D.21).

$$W(i) = \int_x P[i|x]p[x]dx = E[P[i|x]] \quad (\text{D.21})$$

However, since $p(x)$ is generally unknown, there needs to be a way to estimate the aggregation of preferences to the market or population level. [Koppelman \(1975\)](#) classifies five ways to aggregate across individuals.

1. Average individual
2. Classification—same as 1 but with subgroups
3. Statistical differentials—approximate distribution of attributes in population
4. Explicit integration
5. Sample enumeration—use a sample to represent entire population and use sample $\hat{W}[i]$ to estimate population $W(i)$

The most popular method is sample enumeration. If N_S is the size of sample S for population T , then the predicted share in Eq. (D.22a) would be used to estimate the population share. With G mutually exclusive and collective exhaustive segments, each with population N_g , $g=1, \dots, G$, the share can be computed with N_{sg} samples from each group in Eq. (D.22b).

$$\hat{W}[i] = \frac{1}{N_S} \sum_{n=1}^{N_S} P[i|x_n] \quad (\text{D.22a})$$

$$\hat{W}[i] = \sum_{g=1}^G \left(\frac{N_g}{N_T} \right) \left(\frac{1}{N_{sg}} \right) \sum_{n=1}^{N_{sg}} P[i|x_n] \quad (\text{D.22b})$$

One of the advantages of disaggregate demand analysis is the ability to target strategies for specific individuals or segments. But which segments are most effective to identify? If a sample is to be divided into three income segments, two gender segments, and three home location segments, that

grows to 18 distinct segments. A sample of 2000 observations with 18 segments would have on average only 100 or fewer observations per segment. [Koppelman and Bhat \(2006\)](#) show how to decide whether to segment a population using a likelihood ratio test. The test involves dividing the sample into the hypothesis G segments, estimating the model for each segment as well as for the pooled data, and determining the $LL[\beta_g]$ for the segments and $LL[\beta]$ for the pooled data. Then the likelihood ratio test is shown in Eq. (D.23), where $n = \sum_{g=1}^G K_g - K$ is the d.o.f., and K is number of coefficients in the model.

$$2 \left(\sum_{g=1}^G LL[\beta_g] - LL(\beta) \right) \geq \chi_n^2 \quad (\text{D.23})$$

Discrete choice models reflect the behavioral response of users of a system. Beyond the use of these models to forecast the behavior, one can also integrate the models into system design decisions. For example, if the model includes a price variable p_{in} for each product i for segment n of S segments, the pricing can be set to maximize revenue as shown in Eq. (D.24a) or to maximize consumer surplus as in Eq. (D.24b). This is considered *revenue management*, a topic that is covered comprehensively in [Talluri and Van Ryzin \(2004\)](#).

$$\max_{p_{in}} R = \sum_{i \in C} \sum_{n=1}^S N_n p_{in} P_n[i | p_{in}] \quad (\text{D.24a})$$

$$\max_{p_{in}} CS = \sum_n \frac{N_n}{\alpha_n} \ln \sum_j e^{V_{jn}[p_{in}]} \quad (\text{D.24b})$$

For example, consider a market divided into two segments (700 in segment 1 and 300 in segment 2) drawn from [Bierlaire and Lurkin \(2017\)](#). Two products are offered, where demand from segment 1 is shown below:

$$V_{1n_1} = -0.65p_1 + 0.5$$

$$V_{2n_1} = -0.65p_2$$

$$V_{0n_1} = 0$$

Demand for segment 2 is shown below.

$$V_{1n_2} = -0.1p_1 + 0.1$$

$$V_{2n_2} = -0.1p_2$$

$$V_{0n_2} = 0$$

If the objective is to maximize revenue by setting the prices for the two products, the problem is highly nonconvex. This can be illustrated by holding the price of product 2 to $p_2 = \$2$ and plotting the revenue for different values of p_1 for the two segments and in total as shown in Fig. D.5. The plot shows how the total revenue can have multiple local maxima.

Using a local optimization, one solution is $p_1^* = \$14.66$, $p_2^* = \$14.70$, with revenues of \$1.23 for product 1 from segment 1, \$0.73 for product 2 from segment 1, \$755.65 for product 1 from segment 2, and \$682.69 for product 2 from segment 2. The total revenue is \$1440.30.

The decision variables can extend to include capacities, operating policies, among others. A more general problem is the *assortment problem*, which determines the products offered and their quantities for consumption. Typically, the problem may be budget constrained: there are only so much capacity available to allocate quantities of different products, so a seller can choose to sell only one product or to differentiate over many different products with small quantities each. In the assortment problem, the presence of an alternative is a decision variable. This can be modeled with variable $u_{in} = 1$ if product i is made available to segment n and zero otherwise. Then,

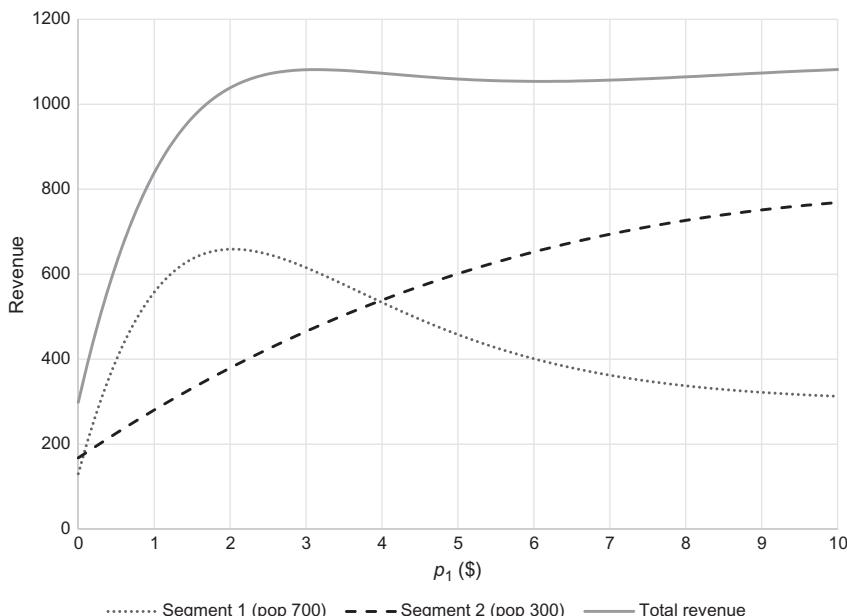


Fig. D.5 Illustration of nonconvexity of revenue maximization on two segments with $p_2 = \$2$.

under a MNL model, the probability of choosing product i is shown in Eq. (D.25) (see [Van Ryzin and Mahajan, 1999](#)).

$$P_n[i|C_n] = \frac{u_{in}e^{V_{in}}}{\sum_{j \in C} u_{jn}e^{V_{jn}}} \quad (\text{D.25})$$

An example assortment problem is shown below to maximize revenue (there are many variations):

$$\max_{q_i, u_{in}} R = \sum_{i \in C} \sum_{n=1}^S p_i q_i$$

Subject to

$$\sum_n \frac{u_{in} N_n e^{V_{in}}}{\sum_{j \in C} u_{jn} e^{V_{jn}}} \geq q_i, \forall i \in C$$

$$\sum_i q_i \leq B$$

$$u_{in} \in \{0, 1\}, q_i \geq 0$$

where N_n is the population of segment n , B is the capacity, q_i is the number of product i available for sale, and p_i and the price of each product.

Consider an example with two segments (700 in segment 1, 300 in segment 2), with input data shown in [Table D.1](#). There are three products ranging in cost, with the higher cost products being better quality. Segment 1 is more sensitive to price and less sensitive to the quality than segment 2. Assume there is a capacity budget of $B=500$ units. The decision variables are which products to offer to each segment and the quantity of each product offered.

The optimal solution to maximizing revenue is to offer only products 2 and 3 to segment 1 and only product 3 to segment 2. The quantities end up being 349.81 for product 2 and 150.19 for product 3, with a total revenue of

Table D.1 Input data for assortment problem example

	Product		
	1	2	3
Prices	\$5	\$10	\$25
V_{in}	1	2	3
$n=1$	$5 - \frac{p_i}{2}$	$5 - \frac{p_i}{2}$	$5 - \frac{p_i}{2}$
$n=2$	$3 - \frac{p_i}{5}$	$4 - \frac{p_i}{5}$	$5 - \frac{p_i}{5}$

\$7252.90. In this case, offering product 1 to either class would dilute the revenue earned from the other products, so it is not offered at all. Product 2 is also not offered to segment 2 because they are less price sensitive and more quality sensitive. This illustrates trade-offs that need to be considered in this type of problem.

INDEX

Note: Page numbers followed by *f* indicate figures, *t* indicate tables, and *b* indicate boxes.

A

- Absorbing state, 427
- Activity and transport system, 168*f*
- Activity-based analysis, 144
- Activity-based network design problems (AB-NDPs), 325–329
- Activity-based network equilibrium models, 144–146
- Activity scheduling, market schedule equilibrium
 - analytical properties, 147–155
 - behavior, 139–141
 - complexity of, 142–143
 - literature review, 143–147
 - model formulation, 147–155
 - solution algorithms, 156–160
 - urban transport system design, 143
- Activity utility, 161*f*
- Agent-based stochastic user equilibrium, 134*f*
- Aggregate multinomial logit model, 218–222
- Aggregation, 419–420, 451–456
- Airports, in California, 210, 211*f*, 212*t*
- A-Lefferts Blvd line, 171, 173*t*
- All-or-nothing assignment, via Dijkstra’s algorithm, 77*b*
- ARC-IT V8.1 National ITS Architecture from US DOT, 421–423
- Artificial neural network algorithms, 187–189
- Association rule learning algorithms, 187–189
- Assortment problem, 454–455, 455*t*
- Attributes, 185–186
- Automated vehicles, 3

B

- Balance equations, 431
- Bayesian algorithms, 187–189
- Bayesian learning, 197

- Bayesian network (BN) approach, 190, 191*f*
- Bayes’ theorem, 187–189
- Bellman equation, 345, 355–356, 359–360
- Bhat’s CEMDAP model, 144
- Big Data, 34–36, 186–187
- Bikeshare incentivization programs, 273
- Bike sharing services, 15–16
- Bilevel network design, 315–329
 - activity-based network design problems, 325–329
 - continuous network design problems, 322–325
 - discrete network design problems, 319–321, 320*b*
- Bimodal network of rail links, 93, 94*f*
- Binding feasible data object diffusion, 263
- BIOGEME, 451
- Birth-and-death process, 430
- Branch and bound algorithm, 156, 156*b*, 277, 303, 319–320, 320*b*
- Broyden-Fletcher-Goldfarb-Shanno (BFGS) method, 220
- Bureau of Transportation Statistics, 210

C

- California Household Travel Survey, 202, 204*t*, 222–224
- Caltrans PeMS, 31, 32*f*
- Canadian ITS architecture, 5–6, 6*f*
- Car2Go, 15–16
- Car-sharing services, 15–16
- Cause/effect relationships, 186–187
- CDFs, comparison of, 444*f*
- CEMDAP models, 144
 - scheduling models, 145*t*
- Center-based location problems, 296
- Chapman-Kolmogorov equations, 427
- Chinese Postman Problem, 287
- Chi square test, 450–451
- Cholesky decomposition method, 163
- Cholesky factorization, 446

- Chow-Regan (CR) policy, 372, 373–374*b*, 380*t*
- Clark method, 446
- Class diagram, 419–420
- Classic Nguyen-Dupuis network, 186
- Clustering algorithms, 187–189
- Cobb-Douglas function, 440
- Columbus smart city challenge
implementation vision, 35*f*
- Commercial integer programming solvers, 156
- Commodity flow survey, 269
- Common prior, 214
- Competitor algorithm, 381–382*b*
- Compulsory activity, 142
- Computational efficiency, 94
- Computer aided routing system (CARS), 293
- Congestion effects, 141, 144–146
evaluation, 67–68
user equilibrium in road networks, 69–76, 72*f*
- Consumer theory, 439–441
- Continuous network design problems (CNDPs), 322–325
gap function algorithm, 323*b*
- Control variables, 344
- Convex optimization models, 206
- Cournot-Nash equilibrium, 315–316
- CPLEX IP solver, 205
- Critical fractile, 343–344
- CR policy. *See* Chow-Regan (CR) policy
- CR-Weibull distributions, 381–382*b*, 382*f*, 384*f*
- Cryptocurrency system designs, 338
- Cumulative distribution function, 342, 343*f*
- Cutting plane algorithm, 197–198, 198*b*
- Cyber physical transport systems, 37
- D**
- Day-to-day adjustment process, 162, 170–171
- Decision-making
policy, 344
under uncertainty, 342–351
- Decomposition approach, 163*f*
- Deep learning algorithms, 187–189
- Dial-a-ride problem (DARP) model, 198, 200*f*, 265, 293
- Dial's algorithm, 87*b*
- Differential privacy, 239–244, 243*t*
consumer surplus, 245*t*
income data with, 247*t*
in transportation, 248
- Diffusion models, *k*-anonymous, 252–253
- Dijkstra's algorithm, all-or-nothing assignment via, 77*b*
- Dimensionality reduction algorithms, 187–189
- Disaggregate demand analysis, 452–453
- Discrete choice models, 244, 413
aggregation, 451–456
consumer theory, 439–441
estimation, 449–451
random utility models
generalized extreme value, 446–449
mixed logit, 446–449
multinomial logit, 441–445
probit, 446–449
revenue management, 451–456
segmentation, 451–456
- Discrete network design problems, 319–321, 320*b*
- Discrete probability mass function, 248
- Distributions, of location noise, 249*f*
- Duality gap cutting plane algorithm, 197–198
- Dynamic dispatch policy, 383*f*
- Dynamic privacy control, 266
- Dynamic programming approach, 158
- Dynamic rescheduling, 146–147
- Dynamic routing algorithm, 381–382*b*
- Dynamic traffic assignment (DTA), 144–146
- E**
- Econometric methods, 186–187
- E-hail taxi services, 273
- Elasticity, 216, 445
- Electric vehicle (EV), 15–16
- EMME, 95, 97*t*
- Engineers, in urban transport systems, 13–18
- Ensemble algorithms, 187–189
- Equilibrium decomposed optimization, 322

- Estimation, 449–451
 Euler's theorem, 283
 Euler tour, 282–283, 285*f*
 Excel Solver, 293–294*b*, 297–300*b*, 304*b*
 EXECUTED trajectories, 173
 Exit problem, 365
- F**
 Facility location problems, 294–308
 center problems, 295
 coverage problems, 295
 median problems, 295
 Fare payment system, 425–426
 FAST-TrIPs, 107–111
 Federal Aid Highway Act of 1956, 4–5
 Federal Aviation Administration, 210
 Finite difference methods, 358–359
 Finland, mobility-as-a-service (MaaS), 7*f*, 8,
 9*f*
 First-come-first-serve (FCFS), 430
 Fixed route transit assignment, 95–98
 frequency-based assignment, 98–106
 schedule-based assignment, 106–107
 transit assignment variants, 107–111
 4-node network, 98–99, 99*f*, 139, 140*f*
 Frank-Wolfe (F-W) algorithm, 82–83, 135,
 256, 260
 for transport network assignment, 83*b*
 Free-floating car-sharing system, 171
 Freight activity system, 178, 179*f*, 210.
 See also Urban freight activity analysis
 Functional flow diagrams, 421
 F-W algorithm. *See* Frank-Wolfe (F-W) algorithm
- G**
 Gap function algorithm, 323*b*
 GBM. *See* Geometric Brownian motion (GBM)
 Generalized extreme value (GEV), 446–449
 General transit feed specification (GTFS),
 31, 32*f*
 Genetic algorithms (GAs), 152, 158
 selective HAPP (G-SHARP), 158–159*b*
 Geometric Brownian motion (GBM)
 candidate project, 356
 investment opportunity, 355–356
 parameters, 349
 real options theory, 355
 for sequential decision problems, 349–350
 trajectory, 349
 GHK simulators, 446, 447*b*
 GIS, time-geographic, 43
 GitHub link, 171, 230
 Google Maps
 Queens, New York, 232–233
 real-time shortest path queries, 234*f*
 Greedy algorithm, 279, 296*b*
 Greenshield's traffic flow model, 120
 G-SHARP model, 160
 GTFS. *See* General Transit Feed Specification (GTFS)
 Gumbel distributions, 244, 442
- H**
 Half-hour interval screenshots, Google Maps, 234*f*
 Heterogeneous market activity scheduling, 170–171
 Hooke-Jeeves method, 322
 Household activity pattern problem (HAPP)
 model, 146, 202–203, 223
 genetic algorithm, 158
 methodology, 146–147
 output comparison based on parameters, 206*f*
 utility maximization, 162
 Hurricane Sandy, in NYC, 31, 32*f*
 Hypercube queueing model, 305, 435
- I**
 Income effect, 440
 Independence of irrelevant alternatives (IIA), 445
 Indicator function, 164
 Inference models, 186–187
 Information and communications technologies (ICTs), 3, 35, 111
 Information-theoretic control problem, 252
 Insertion algorithms, 152
 Instance-based algorithms, 187–189
 Integer programming (IP), 156*b*, 197–206
 cutting plan algorithm, 198*b*

- Intelligent transportation systems (ITS), 5
 Canadian ITS Architecture, 5–6, 6f
 performance measures, 423–424t
- International Conference on Travel Behavior Research (2012), 10
- International Council on Systems Engineering (INCOSE), 416
- Internet of Things (IoT), 33–37, 186–187, 341
- Inverse dial-a-ride problem, 205–206
- Inverse household activity pattern problem, 205
- Inverse integer programming, 197–206
- Inverse linear programming, 194–197
- Inverse mixed integer linear programming, 193–194
- Inverse nonlinear programming, 206–211
- Inverse optimization (IO), 146–147, 193–194, 197, 211–212
 advances and applications, 193t
 multiagent framework, 213
- Inverse shortest path, 196
- Inverse traffic assignment problem, 206–208, 210
- Inverse transportation problems
 integer programming, 197–206
 linear programming, 194–197
 machine learning applications, 187–192
 nonlinear programming, 206–211
- Inverse vehicle routing problem, 205–206, 208f
- Isoultility curves, 439–440, 440f
- Iterative optimization algorithm, 322
- ITS. *See* Intelligent transportation systems (ITS)
- J**
- Jackson network, 437
- K**
- Kang-Chow-Recker (KCR)
 trip paradox, 140
 utility paradox, 141
- k*-anonymous models
 diffusion, 252–253, 253f
 entropy value, 264f
 ten-anonymous diffusion, 264f, 267f
 tour diffusion, 258–265, 259–261b
- two-path diffused data object set, 267f
 vehicle assignment diffusion, 253–258, 255–256b
- k*-anonymous privacy control strategy, 251
- Karush-Kuhn-Tucker (KKT) optimality conditions, 80–81, 90, 206, 208–209
- k*-best traveling salesman problem, 263
- Königsberg bridge problem, 282, 282f
- Koopmans-Hitchcock transportation problem, 253–254
- Kummer function, 369
- L**
- Lagrange multipliers, 206–208
- Lagrangian function, 142
- Lagrangian relaxation method, 303
- Laplace distributed noise, 242f, 244
- Least Squares Monte Carlo simulation (LSM), 359–360, 360–361b
 multioption, 373–374b
- Likelihood ratio test, 450–451
- Linear-in-parameters utility functions, 443
- Linear program (LP), 76–77, 156
 inverse, 193
- Line planning problems, 309–314, 313b
- L_1 norm minimization, 195
- Load capacity, 169–170, 170f
- Location noise distributions, 249f
- Logistic distribution, 443–444
- Logistic regression, 443
- Logit models. *See* Mixed logit models
- LP. *See* Linear program (LP)
- Luxembourg commuter mode, 191f
- M**
- MaaS. *See* Mobility-as-a-service (MaaS)
- Machine learning techniques, 188–189t
 instance-based algorithms, 187–189
 inverse transportation problems, 187–192
 learning style, classification, 187
 regression algorithms, 187–189
 route sensitivity, 192
 urban transport problems, 187, 190t
- Manheim-Florian-Gaudry (MFG)
 framework, 36, 74
- components, determination
 Bixi Toronto, 20–22b

- EV taxi system in Hong Kong, 22–23*b*
- Kutsuplus, 23–24*b*
- supply-demand equilibration in, 111
 - on-street parking and cruising, 117–122
 - taxi–customer matching, 123–125
 - Vickrey morning commute problem, 111–116
- for transport systems analysis, 18–23, 19*f*
- Many-to-many assignment game, 236
- Marginal external cost of congestion (MECC), 71
- Market entry, 365
- Market schedule equilibrium, multimodal systems
 - activity scheduling
 - analytical properties, 147–155
 - behavior, 139–141
 - complexity of, 142–143
 - literature review, 143–147
 - model formulation, 147–155
 - solution algorithms, 156–160
 - arrivals and departures, cumulative diagram of, 165*f*
 - assignment, 170*t*
 - decomposition approach, 163*f*
 - defined, 139, 160
 - population activity scheduling, 167
 - research and design challenges, 179–181
 - transport system
 - aggregation problem, 162–169
 - integration with MATSim, 170–177
 - urban freight activity analysis, 177–179
- Markov chains, 426–429
- Markovian models, 192
- Martingales, 347
- Mathematical programming, 76, 192–193, 193*t*, 420–421
- Mathematical program with equilibrium constraints (MPEC), 275
- MATLAB, 210
- MATSim, 24–25, 26*f*, 95, 107–111, 158
 - day-to-day adjustment process, 162, 170–171
 - genetic algorithm-based solution algorithms, 176–177
 - input data, 172*f*
 - integration with, 170–177
- sampling suboptimality, 177
- simulation-based approach, 176–177
- Maximal covering location problem (MCLP), 302–303
 - formulation, 302–303*b*
 - simpler model extension, 306
 - solutions, 303*t*
- Maximum expected covering location model (MEXCLP), 306
- Mean-reverting process. *See* Ornstein–Uhlenbeck (O-U) process
- Measure of accessibility, 451
- MECC. *See* Marginal external cost of congestion (MECC)
- Median arrival times, 204*t*
- p-Median problem, 295, 296–300*b*
- Memory-based learning, 187–189
- Method of successive averages (MSA), 89, 107, 214, 215*b*, 322
 - for convenience, 166
- Metropolitan planning organization (MPO), 107–111
- Metropolitan Transportation Authority (MTA), New York, 40–42
- Metropolitan Transportation Commission (MTC), 107–111
- MILATRAS model framework, 107, 110*f*
- Mimic biological neural networks, 187–189
- Minimum spanning tree (MST), 277–281
 - algorithm, 279*b*
 - variants, 281
- Mixed integer linear programming problem, 152, 156
- Mixed logit models, 216–222, 446–449
 - aggregate multinomial, 219
 - structure, 162–165, 167*b*
- Mixed multinomial logit models, 218–222, 221*t*, 449*b*
- MNL models, 164, 443–445, 448
- Mobile Millennium/Century project, 37
- Mobility
 - monitoring, 37–42, 39*f*
 - service system, 253
- Mobility-as-a-service (MaaS), 8, 171, 177–178, 288, 330
 - Finland, 7*f*, 9*f*
- Model and solution method, 213–216
- Model calibration and validation, 212*t*

- Monte Carlo simulation, 359
- Motor Carrier Act and Staggers Rail Act of 1980, 5
- MPO. *See* Metropolitan planning organization (MPO)
- MSA. *See* Method of Successive Averages (MSA)
- MST. *See* Minimum spanning tree (MST)
- MTC. *See* Metropolitan Transportation Commission (MTC)
- Multiagent inverse optimization learning framework, 224
output dual prices, 235 f
- Multiagent inverse transportation problems, 225
California Household Travel Survey, 222–224
fixed-point problem, 214 f
heterogeneous link, 213
mixed logit model, 216–222
model and solution method, 213–216
- Multicommodity flow problem solution, 226
- Multimodal household activity pattern problem (mHAPP) models, 146–148, 152, 155 f , 166
decomposition algorithm, 167 b
genetic algorithm-based solution algorithms, 176–177
mixed logit market schedule equilibrium, 167 b
network data, 168–169 t
urban freight activity analysis, 178
- Multinomial logit models, 162–163, 220, 221 t , 244, 441–445
- Multiple day scheduling, 152
- N**
- Nash equilibrium, 144–146, 266
- National Mass Transportation Assistance Act of 1974, 5
- Nested logit model, 448 f
- Net present value (NPV), 352, 353 f , 354–355
- Network assignment models, 95
- Network design. *See also* Network design problems (NDPs)
activity-based, 325–329
- bilevel, 315–329
under coexisting systems, 330–338
games, 334–336
optimization modeling, 274
privacy-aware, 337–338
research and design challenges, 338–340
symbiotic, 331–333
test networks for, 95, 97 t
- Network design problems (NDPs), 274–275
bilevel, 276 t
classification, 277
by complexity, 276 f
directness measure, 314 b
Euler diagram, 275, 276 f
facility location problems, 294–308
line planning problems, 309–314
minimum spanning trees, 277–281
multiple operators, 276 t
single level, 276 t
software packages, 277 t
sorted savings for all node pairs, 291–292 t
Steiner Tree Problem, 281, 281 f
surrogate-based, 319 f
traveling salesman problems, 281–288
vehicle routing problems, 288–294
- Network Investment Deferral Option (NIDO), 364 f
- Network learning
Google Maps Queries in Queens, New York, 232–233
methodology, 224–228
Nguyen-Dupuis network, 228–231
with privacy awareness, 265–267
- Newton-Raphson method, 450
- Nguyen-Dupuis network, 186, 186 f , 228–231, 228 f
- NLP. *See* Nonlinear programming (NLP)
- Nonanonymous diffusion, 258
- Nonenumerative route behavior mechanism, 192
- Nonlinear programming (NLP), 93, 206–211
- Nonlinear schedule delay cost function, for work trips, 142–143, 143 f
- Normal distribution, 443–444, 449 b
- NP-hard complexity, 275–276

O

- OCTAM network, 203*f*
 OD distributions, comparison, 251*f*
 Off-hour delivery policy tests, 177
 Onboard equipment (OBE), 421–423
 On-bus fare ticketing system, 419*f*
 Online learning framework, 202, 231*f*
 link dual price trajectories, 233*f*
 network monitoring, 230
 Operating mode switching, under uncertainty, 363–371
 Operator competitive privacy control
 k-anonymous diffusion models, 252–253
 k-anonymous tour diffusion models, 258–265
 k-anonymous vehicle assignment diffusion models, 253–258
 Operator policies, 95–98
 Operator privacy, 239–240
 Optimal parameters, 450
 Optimal strategy algorithm, 102*b*
 Optimal timing, under uncertainty, 352–363
 Optimization and Scale Economies in Urban Bus Transportation (1972), 5
 Optimization model, 152, 156
 Ornstein-Uhlenbeck (O-U) process, 349–350

P

- Parameter estimation, 176, 217*t*
 Parking facility, 143–144
 Passenger groups, equilibrium link flows for, 107, 109*f*
 Pay Fare, 421, 421*f*
 PeMS, 31, 32*f*
 Perfectly anonymous diffusion, 258
 Performance measures, comparison, 224*t*
 Pickup and delivery problem (PDP), 293
 Poisson process, 436
 Pollaczek-Khintchine (P-K) formula, 434–435, 437
 Population activity scheduling, 167
 Portfolio management
 definition, 341
 in-the-money simulated path values, 362*t*
 market switching, 365*f*

- research and design challenges, 386–387
 sequential network design and timing, 371–385
 three curses of dimensionality, 346–347
 uncertainty
 decision-making under, 342–351
 operating mode switching under, 363–371
 optimal timing under, 352–363
 Prim's algorithm, 279–280*b*, 280*f*
 Privacy-aware network design, 337–338
 Privacy in learning
 differential privacy, 241–244
 network learning, 265–267
 operator competitive privacy control
 k-anonymous diffusion models, 252–253
 k-anonymous tour diffusion models, 258–265
 k-anonymous vehicle assignment diffusion models, 253–258
 user location data, 248–250
 user privacy, 240–250
 user survey data, 244–247
 Privacy preservation problem, 239
 Probability density function, 342, 343*f*
 Probit models, 446–449
 PT04 Transit Fare Collection Management, 422*f*
 Public information, 239
 Pulse, urban, 40–42

Q

- Queens freeway network, 232*f*
 Queue, 429–435
 Queueing analysis, 413
 Markov chains, 426–429
 queues, 429–435
 spatial queues and queue networks, 435–439
 Queue networks, 435–439, 437*f*

R

- Rail links, bimodal network, 93, 94*f*
 Random state variables, 344
 Random utility models
 generalized extreme value, 446–449

- Random utility models (*Continued*)
 mixed logit, 446–449
 multinomial logit, 441–445
 probit, 446–449
- Randperm, 159–160
- Real options methods, 352–363
- Rebalancing algorithm, 253
- Reference policy, 381
- Regression algorithms, 187–189
- Regularization algorithms, 187–189
- Representative utility, 441–442
- Revenue management, 451–456, 454*f*
- Ridesharing policy, 146
- Roadside fare ticketing system, 419–420*f*
 Pay Fare, 421*f*
- Road transport networks
 equilibration of users on, 74
 user equilibrium in, 69
 assignment, 76–85
 congestion effect, 69–76, 70*f*, 72*f*, 74*f*
 data and software, 94–95, 97*t*
 traffic assignment variants, 85–94
- Robust optimization model, 212–213
- Routes
 distribution of flows on, 73*b*
 generation method, 313, 313*b*
 sensitivity, 192
- Routing algorithms, 199
- S**
- Sampling elasticity, 177
- San Pedro Bay Ports, 207*f*
 in Southern California, 206
- Schedule-based assignment model, 107
- Schedule-based transport systems, 139
- Schedule effects, 141
- Schedule file, 173*f*
- Scheduling models
 in CEMDAP, 145*t*
 MATSim, 147
- Scopus h-index of transportation research, 413
- Secretary problem, 351
- Segmentation, 451–456
- Sequential decision problems, 344
- Sequential network design and timing, 371–385
- Simulated origins and destinations, 250*t*
- Simulation-based test, 425
- Sioux Falls network, 363, 364*f*
- Smart cities, 33
 Columbus smart city challenge
 implementation vision, 35*f*
 definition, 33
 functions, typology, 33*f*
 mobility provision, 129*f*
- Smart economy, 34
- Smart environment, 34
- Smart governance, 34
- Smart living, 34
- Smart mobility, 34
- Smart people, 34
- Smooth-pasting conditions, 356, 358
- Solution algorithms
 genetic algorithm, 158
 household activity pattern problem
 models, 156
 linear program, 156
- Spatial distribution, 249*f*
- Spatial queue analysis, 435–439, 435*f*
- Spatial-temporal steady-state equilibrium models, 123
- Spatial-temporal taxi equilibrium models, 125
- Stackelberg equilibrium, 315–329, 335
- State probabilities, comparison, 434*f*
- State transition diagram, 427, 427*f*
 birth-and-death process, 430*f*
- State transition function, 344
- Steady-state equilibrium, 119–120
- Steiner tree problem, 281, 281*f*
- Stochastic Bayesian learning, 197
- Stochastic differential equation (SDE), 347–349, 350*f*
- Stochastic scheduling algorithm, 146–147
- Stochastic search algorithm, 158
- Stochastic user equilibrium, 86*b*
 agent-based, 134*f*
- Stochastic variables, 343*f*
- Stock quantity, 343–344*b*, 344
- Subsequent models, 144
- Substitution effects, 179, 440
- SUE models, 89*b*, 91–92
 for transit network, 104–105
- Surrogate-based optimization, 318–319, 325
- Symbiotic network design, 331–333

- Symmetric linearization algorithm, 105*b*
 System, defined, 416
 System optimum (SO) behavior, 74
 Systems engineering, 413, 416–426, 416*f*
 Systems engineers, 417
 System splits, 310
- T**
- Taxi equilibrium models, 125*f*, 126
 Test network
 convergence of, 220*f*
 with node and link IDs, 219*f*
 TIF. *See* Travel impulse field (TIF)
 Time-geographic theory, 42–47, 44–45*f*,
 47–48*f*, 144
 TMFs. *See* Travel momentum fields (TMFs)
 Toronto, equilibrium and social optimum
 model, 120, 121*f*, 122*t*
 Tour diffusion models, k -anonymous,
 258–265
 Tour generation approach, 265
 Toy network, 227*f*
 Trade-offs, 186–187, 216, 251, 420–421,
 440
 Traffic assignment algorithms, 95, 96*f*
 test networks for, 95, 97*t*
 Traffic equilibrium model, 91
 TransCAD, 95, 98*f*
 Transit assignment models, 98
 Transit equilibrium assignment problem,
 105*b*
 Transition matrix, 427
 Transit network
 design, 310
 SUE model for, 104–105
 time-expanded network representation
 of, 108*f*
 uncongested, 102*b*
 Transport applications, for differential
 privacy, 244
 Transportation network systems, 144–146
 activity and, 168*f*
 agency/operator, 155
 aggregation problem, 162–169
 attributes, 185
 Frank-Wolfe (F-W) algorithm, 83*b*
 integration with MATSim, 170–177
 logistics systems, 197
 supply-demand curves for, 69–70, 70*f*
 as two-sided markets
 day-to-day adjustment processes,
 132–133
 motivation, 126–127
 two-sided market framework, 128–132
 Transportation research at universities,
 413–415
 h-index rankings, 414–415*t*
 Travel impulse field (TIF), 60–62
 Traveling salesman problem (TSP), 277,
 281–288
 alternative solutions, 334*f*
 bilevel problem, 318*f*
 dial-a-ride problem, 305*f*
 equilibrium solutions, 337*t*
 Euler's theorem, 283
 example network, 334*f*
 heuristic solution for, 283*b*
 input data, 314*f*
 insertion heuristic, 286*f*
 integer programming solution, 284*f*, 291*f*,
 293–294*b*, 297–300*b*, 298*f*
 line planning problem, 309*f*
 relocation ignoring/with queue delay,
 308*f*
 set covering problem, 301*f*
 substitutable and complementary toll
 roads, 336*f*
 surrogate-based NDP, 319*f*
 worst-case matched odd-degree node
 lengths, 284*f*
 Travel momentum fields (TMFs), 48–56,
 48*f*, 51*f*, 53–55*f*
 before-after analysis, 60–62, 60–61*f*
 construction, from trajectories, 51–53*b*
 transport route projections, 56–59,
 56–59*f*
 Travel-related survey, 244
 Travel times, 257*t*, 262*t*
 Trip
 distribution forecast models, 248
 economic activity, 141*f*
 Truck activity scheduling, 178
 Truck dispatching problem, 288
 Truck on-street parking policies, 177–178
 TSP. *See* Traveling salesman problem (TSP)
 24-h decision-making timescale, 144–146

U

- Uber, 254
- UE. *See User equilibrium (UE)*
- Uncertainty
- decision-making under, 342–351
 - operating mode switching under, 363–371
 - optimal timing under, 352–363
- Uncongested transit network, 102 b
- Unified Modeling Language (UML), 417, 418 t , 421
- Uniform distribution, 443–444
- Uninformed prior *vs.* optimal invariant common prior, 223 t
- Urban freight activity analysis
- economic and environmental sustainability, 177
 - multimodal household activity pattern problem framework, 178
 - policymakers, 178
 - time distribution of day arrivals, comparison, 175 f
 - welfare measurement before *vs.* after headway reduction, 174 f
- Urban public agencies, 177–178
- Urban pulse, 40–42
- Urban transport systems (UTS)
- component determination
 - EV taxis in Hong Kong, 16–17 b
 - Kutsuplus in Helsinki, 17–18 b
 - Toronto bike share, 13–15 b
 - definitions, 10–13, 11–13 t
 - elements of, 11–13 t
 - engineers, 13–18
 - evolution of, 8, 8 f
 - machine learning applications, 187–192
 - simulation tool for evaluation, 24–25
 - technology, IoT-based infrastructure, 36–37, 36 f
 - theory, 34–35
- Urban truck routing, 205–206
- User activity scheduling models, 144, 146
- User equilibrium (UE)
- agent-based stochastic, 134 f
 - behavior, 74
 - model, 79–80 b , 93–94
 - in road networks, 69
 - assignment, 76–85
 - congestion effect, 69–76, 70 f , 72 f , 74 f
 - data and software, 94–95, 97 f
 - traffic assignment variants, 85–94
- User location data, 248–250
- User privacy, 239–250
- User scheduling response, 139, 142
- heterogeneity, 139
 - to urban transport systems, 139
- User survey data, 244–247
- US National Household Travel Survey, 244–245
- US National ITS Architecture, 417–418
- Utility curves, 441 f
- Utility-generating goods, 142

V

- Value-matching, 358
- Value of time (VOT), 445
- Value—strong duality, 195
- Variational inequalities (VI), 90–91
- advantages, 91
- Vehicle assignment diffusion models, k -anonymous, 253–258
- Vehicle emissions, 146
- Vehicle routing problem (VRP), 178–179, 258–259, 263, 288–294, 290 b
- Vehicular traffic monitoring, 37
- Vickrey morning commute problem, 111–116
- V model, 416

W

- Wardrop's user equilibrium principle, 78, 90–91, 208–209
- Wiener process, 347–349
- Winner-take-all methods, 187–189

Z

- Zipcar's service, 15–16

Informed Urban Transport Systems

Classic and Emerging Mobility Methods toward Smart Cities

Informed Urban Transport Systems: Classic and Emerging Mobility Method toward Smart Cities examines how information gathered from these new technologies can be used for optimal transport planning and operation in urban settings.

KEY FEATURES:

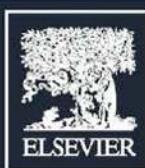
- Serves as a focal point for those in artificial intelligence, data science, and urban planning seeking to solve modern transportation problems, with a focus on urban transport systems.
- Examines classic methodologies and current analytical innovations focused on capturing, predicting, visualizing, and controlling mobility patterns, while discussing future trends.
- Provides a comprehensive overview and update of urban transport systems by considering the role of emerging mobility services in public transport, freight services, and traffic, such as shared mobility service models, data privacy, and automation.

Transportation researchers, and those drawn to these systems from related disciplines, such as artificial intelligence, energy, applied mathematics, electrical engineering, environmental science, will benefit from the book's deep dive into the transportation domain, allowing for smarter technological solutions for modern transportation problems. *Informed Urban Transport Systems: Classic and Emerging Mobility Methods toward Smart Cities* helps create solutions with fewer financial, social, political, and environmental costs for the populations they serve. The book engages both traditional and modern transportation disciplines to help appreciate their respective approaches.

Joseph Chow is an Assistant Professor in the Department of Civil & Urban Engineering and Deputy Director of the C2SMART University Transportation Center at New York University, New York, NY. He heads BUILT@NYU, the Behavioral Urban Informatics, Logistics, and Transport Laboratory. His research expertise lies in transportation systems, with emphasis on multimodal networks, behavioral urban logistics, smart cities, and transport economics. He is the elected Chair of the Urban Transportation SIG at INFORMS Transportation Science & Logistics Society, and an appointed member of the editorial board at Transportation Research Part B. He is a recipient of a CAREER Award from the National Science Foundation and the 2018 Cambridge Systematics New Faculty Award from the Council of University Transportation Centers.

SOCIAL SCIENCE

ISBN 978-0-12-813613-3



elsevier.com/books-and-journals

