M.O. Ball et al., Eds., *Handbooks in OR & MS, Vol. 8*
© 1995 Elsevier Science B.V. All rights reserved

Chapter 3

Stochastic and Dynamic Networks and Routing

Warren B. Powell

Department of Civil Engineering and Operations Research, Program in Statistics & Operations Research, Princeton University, Princeton, NJ 08544, U.S.A.

Patrick Jaillet

MSIS Department, The University of Texas at Austin, U.S.A, and Laboratoire de Mathématiques et Modélisation, ENPC, France

Amedeo Odoni

Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA 02159, U.S.A.

1. Introduction

The field of logistics is becoming increasingly dominated by the need for technologies that support real-time decision making. Using recent advances in information technologies, decisions concerning the routing and scheduling of drivers and vehicles, management of vehicle inventories, and the design of service offerings, can be made in a real-time environment with information that is constantly changing. Dynamic and stochastic models are playing an increasingly important role in such a setting: by definition, one is forced to make decisions before all the information one would wish to have becomes available and then modify these decisions as new information is received. The most common form of stochasticity arises as a result of uncertainty concerning some aspect of demand (level of demand, location, timing, etc.) but many other forms may be present, as well (length of travel times, resource availability, service breakdowns, etc.). As a rule, once a set of decisions has been made and some action has been taken, the decision-makers have the opportunity to observe an outcome of (some of) the uncertain events and must then respond to these events. The inherent dynamism of this type of operation often introduces important analytical complications: initial decisions may be greatly affected by how well the decision-makers — and the system they manage — are equipped to respond to subsequent random events.

Dynamic and stochastic models undoubtedly represent the 'wave of the future.' Their increasing prominence is driven by technology: the explosive growth in the availability of real-time information about transportation and logistics systems is turning the focus of operations researchers away from the traditional static planning models. Carriers and shippers are avidly seeking the benefits afforded by the ability to rapidly and continually 'reconfigure' the operation of a transportation system to improve service or reduce cost. In addition, even when it comes to strategic planning, methodological developments over the last few years have made possible the development of models that capture better the uncertainty and the dynamic characteristics associated with the transportation and logistics environment. Clearly such models often constitute far more accurate representations of reality than some of their deterministic and static forebears.

Model applications in this general area can, in fact, be divided into two broad classes along the strategic vs. tactical lines just mentioned. The first class of models deals with a priori optimization, or, in more technical terms, two-stage programming with recourse. A good example is the location and sizing of a warehouse. Typically, a decision on the warehouses location and capacity must be made on the basis of a probabilistic description of customer demands. Once the facility is built, the characteristics of the 'market' become better known and decisions can then be made about how to serve demands using a fleet of vehicles. However, this service may be unnecessarily costly or inefficient, if the selected location or size of the warehouse turn out to have been poor, in the first place. Note that, in such problems, the time from when the first set of decisions is made to when the random outcomes become (partially) known, is typically fairly large.

The second set of applications involves multistage problems where the process of making decisions and observing outcomes occurs on a continuous, rolling basis. A set of vehicles may be routed over the course of the day while demands are continuously being called in. With each new demand, the vehicle tours may be redesigned, not only to accommodate actual demands, but anticipated demands as well. Another example arises in dynamic fleet management, where empty vehicles are repositioned to anticipate future demands. Such decisions are made daily, as shipper demands are continuously being received at a dispatching center.

The two classes of problems, then, are both stochastic, since we have to anticipate the future, and dynamic, since decisions are made over time. However, the study of the properties of stochastic two stage problems (a priori optimization) can often be done without explicitly handling the dynamics of the problem — in essence, through a static model. By contrast, it is very common to solve stochastic, multistage problems using deterministic models. We may approximate the future using a deterministic model, updating the information on a rolling basis as random events become known. As a result, our presentation spans (i) static, stochastic models, (ii) deterministic dynamic models, and (iii) multistage stochastic models. We feel that a unified presentation of all three perspectives provides valuable insights into the formulation and solution of this difficult and important class of problems.

One question that comes up with surprising frequency is: what constitutes a dynamic model? To answer this, we must first distinguish between a problem, a model, and the application of a model. A problem is dynamic if one or more of its parameters is a function of time. This includes such problems as vehicle routing with time windows or with time-varying travel times. Note that two types of dynamic problems are covered here. The first type, which we call problems with dynamic data, are characterized by information that are constantly changing.

Dynamic data might include real-time customer demands, traffic conditions, or driver statuses. The second type is problems with time-dependent data which is known in advance. In this category, we would include problems such as vehicle routing with time windows where all the information is known in advance, but where this data is a known function of time. Other examples of time-dependent data might be customer demands or travel times, if we can assume them to be known functions of time.

Similarly, a model is dynamic if it incorporates explicitly the interaction of activities over time. The simplest dynamic model is a dynamic network, a construct widely used in routing and scheduling problems. It is useful, however, to distinguish between deterministic, dynamic models, and stochastic models which explicitly capture the *staging* of decisions and the realization of random variables. For example, it is not unusual to solve deterministic, dynamic models without recognizing in any way the dynamic structure of the problem. By contrast, stochastic, dynamic models require specific steps to be taken in the design of a solution strategy.

Finally, we have a dynamic application if a model is solved repeatedly as new information is received. Dynamic applications of models place tremendous demands on access to real-time data and on the performance of algorithms. Typically, it is necessary to update information, optimize and return results in a matter of minutes or even seconds.

A useful illustration of these concepts arises in the problem of routing a vehicle over a congested transportation network. Clearly, traffic conditions are changing over time, and hence the problem is dynamic. We might select an optimal route using a static model, if we chose not to represent dynamic conditions explicitly within the model. (For example, we could find the optimal route by simply minimizing average travel times, thus working with one particular static representation of the network.) We might also solve the static model repeatedly as new information became known, giving us a dynamic application of a static model. Yet a third alternative might be to develop a model that would consider explicitly the anticipated dynamic changes in traffic conditions and apply it once at the outset of the period of interest. In other words we would select one route and then stick to it even as conditions change. This would constitute a static application of a dynamic model. Finally, had we chosen to solve the dynamic model repeatedly as new information became known, we would have a dynamic application of a dynamic model. This example, then, indicates how one can have a static or dynamic application of a static or dynamic model.

Stochastic, dynamic models represent a rich area of research. Important dimensions of the problem that require attention include:

1. Definition of specific application areas.
2. Development of tractable mathematical models, with particular emphasis on optimization under uncertainty.
3. Development of efficient formulations and solution algorithms.
4. Evaluation of alternative models.
5. Integration of real-time optimization models and on-line databases.
6. Implementation and user acceptance of real-time decision support systems.

In this chapter, we cover aspects of the first four issues only; the last two are beyond the scope of our presentation. The following list is only indicative of the enormous variety of problems that might be addressed using dynamic and stochastic models:

1. Production and inventory planning: Each week (or month) it is necessary to develop a production, inventory and transportation plan to meet anticipated demands over a specified time horizon of perhaps several months or one year).

2. Vehicle routing: This involves the dynamic routing of a fleet of vehicles to meet real-time customer demands as they are called in. Problems in this class can arise in a real-time setting, where these decisions must be made in response to actual conditions, or in a planning environment, where future demands must be forecast and are subject to considerable uncertainty.

3. Design of a service network: A carrier must design a package of scheduled transportation services which must then be communicated to the market and modified dynamically in response to the market's feedback.

4. Repositioning of empty vehicles to anticipate future demands: When too many vehicles accumulate in a single city or region, it may be necessary to move excess vehicles to adjacent areas where better opportunities for generating revenue may exist.

5. Booking strategies: In static models, it is common to minimize costs subject to the constraint of satisfying all demands. In a dynamic context, it is usually not possible to satisfy all the demands all the time. It may be necessary to turn down customer requests that cannot be satisfied or to try to postpone certain services. Dynamic booking controls the commitment of a carrier to handle incoming demands and allows the carrier to schedule when they will be served.

6. Costing and pricing: Costing and pricing activities in transportation require understanding the interactions among different markets over time. For example, the profit from a particular vehicle movement may depend on the scheduling and timing of other movements around the network. Costing and pricing issues can arise in both a real-time setting (requiring a 'spot' evaluation) as well as in the longer run, e.g., evaluating a new contract which might affect existing conditions on a network.

7. Determination of level-of-service for specific loads/shipments: It is possible to provide different levels of service to different customers, by design. Carriers must regularly make decisions about how much effort to put into a particular shipment to ensure adequate service. With proper planning, it should be possible to set appropriate expectations when a load is accepted for service.

8. Tactical sales planning and load solicitation: In some types of transportation, it is possible to use sales and marketing adjustments to help control the size of demand served on a day to day basis. In truckload trucking, for instance, it is possible to anticipate the total number of trucks that will be available in a given region one to three days in advance. Then, the telemarketing department of the carrier can call shippers and solicit enough freight in each region to help balance the number of loads with the available capacity.

9. Fleet mix specification: This is the standard problem of determining how many of each type of vehicle to include in the fleet of a carrier.

10. Facility planning and design. Facility location and sizing decisions are customarily viewed as static problems which are solved using static models. Quite often, however, there are important advantages in examining such problems in a dynamic context. Examples include determining the size and location over time of plants and warehouses in a logistics network, as well as the location of terminals and driver domiciles for a carrier.

Because of the length of the chapter it is useful to provide a 'roadmap' though it for the reader. It consists of four major parts:

Part I A priori optimization
Section 2: A priori (two-stage) stochastic models
Part II Dynamic models in logistics
Section 3: Modelling issues for dynamic problems
Section 4: Dynamic models in transportation and logistics
Section 8: Stochastic programming models in networks and routing
Part III Dynamic networks
Section 5: Algorithms for deterministic, dynamic networks
Section 6: Infinite horizon network models
Section 7: Stochastic programming for networks
Section 9: Approximations for networks with random arc capacities
Part IV Model evaluation
Section 10: Evaluating dynamic models

Part I is dedicated to a priori optimization in routing, covering shortest paths, traveling salesman-type problems and vehicle routing. These problems arise when decisions must be made before random outcomes (typically customer demands) are known.

Part II covers dynamic models of problems arising in transportation and logistics, and includes a discussion of important modeling issues, as well as a summary of dynamic models for a number of key problem areas. Section 8, which covers stochastic, dynamic models, is presented near the end of the chapter, following a review of fundamental concepts in stochastic programming.

Part III focuses specifically on dynamic networks, which provide an important foundation for addressing many problems in logistics planning. Section 5 presents algorithms that have been specialized for dynamic networks. Section 6 discusses results for solving infinite networks, including both exact results for stationary infinite networks, and model truncation techniques. Section 7 presents basic results and concepts from the field of stochastic programming, oriented toward their application to network problems. This discussion provides a general framework for formulating and solving stochastic, dynamic network problems. That framework is used to present two stochastic programming models in Section 8. Section 9 then covers a series of results that have been derived for the special case of dynamic networks with random arc capacities. Links with random upper bounds can be used to model random demands for many problems in logistics and distribution.

This section summarizes work that has leveraged this special structure to obtain much stronger results than have been possible in other, more difficult problem areas such as dynamic vehicle routing.

Finally, Part IV addresses the special question of evaluating stochastic, dynamic models. The issue of evaluation is an important one and distinguishes sharply static from dynamic models. In static models, the choice of objective function is usually fairly obvious. That same objective function also provides the yardstick for evaluating the quality of the solution. In dynamic models, by contrast, the objective function used to make decisions on a rolling horizon basis may often have little to do with the measures developed to evaluate the overall quality of a solution.

We have sought to present our material in a way that might enhance two perspectives, namely the formulation and solution of models and the analysis of operating strategies. Our goal is to provide a structured presentation of alternative modeling and algorithmic frameworks that may be useful to both practitioners and researchers.

While every effort has been made to present a reasonably complete picture, this chapter is not intended to be a comprehensive survey of all applications of dynamic networks. Early bibliographies that the reader may wish to consult include those of Bradley [1975] and of Golden and Magnanti [1988]. Assad [1987] provides an extensive review of the literature on multi-commodity flows. A thorough treatment of network models and algorithms is provided in Ahuja, Magnanti & Orlin [1992]. Finally, of particular value is the extensive review of dynamic networks given in Aronson [1989]. This review provides a summary of specialized algorithms for dynamic networks, as well as a fine discussion of the planning horizon literature as it relates to dynamic networks. Aronson's survey also includes work drawn from other applications and a number of papers that involve static networks. In contrast, our presentation is restricted to dynamic models and related methodologies, focusing mostly on applications in logistics. In addition, while we consider deterministic models in depth, we also put considerably greater emphasis on stochastic models than did Aronson's review. We also cover a wider range of issues associated with model formulation, and the interaction between model formulation and the design of algorithms.

2. A priori (two-stage) stochastic models

2.1. *Introduction and problem definitions*

This section is concerned with a specific family of routing problems on networks. Most of these problems have been addressed only recently, and their common characteristic is the explicit inclusion of probabilistic elements in the problem definitions, as will be explained below. For this reason we shall refer to them as stochastic routing problems (SRPs). In the sense that these problems are addressed in a non-dynamic environment, they are also designated as 'static' — to distinguish them from the dynamic and stochastic routing problems (DSRPs) which will be the subject of the next section.

The objective here is to offer a broad overview that emphasizes the fundamental concepts and provides an up-to-date reference list for the numerous developments which have been published in this area during the last few years. The topics that will be reviewed deal with:

(i) the concept of 'robust' a priori solutions to SRPs that perform well on average;

(ii) the performance of the a priori optimization approach relative to the strategy of 're-optimization', i.e., the strategy of solving optimally every potential instance of the original problem;

(iii) the computational complexity and the combinatorial properties of a priori optimization;

(iv) exact and heuristic algorithms, together with theoretical worst-case and average-case bounds for the heuristics;

(v) the asymptotic behavior of a priori optimization and of re-optimization strategies;

(vi) available computational experience.

This subsection will introduce all these ideas, along with formulations of the several problems to be discussed in more detail later. The presentation shall draw heavily on the general framework, which was first established in Bertsimas, Jaillet & Odoni [1990].

There are several motivations for investigating the effect of including probabilistic elements in routing problems. Among them two are of particular importance. The first is the desire to define and analyze models which are more appropriate for those real-world problems in which randomness is not only present but a major concern, as well. There is a plethora of important and interesting applications of SRPs, especially in the context of strategic planning for collection and distribution services, communication and transportation systems, job scheduling, organizational structures, etc. For such applications, the probabilistic nature of the models makes them particularly attractive as mathematical abstractions of real-world systems.

The second motivation is interest in investigating the robustness (with respect to optimality) of optimal solutions to deterministic routing problems, when the instances for which these problems have been solved, are modified. In our case, we confine the investigation to problems on networks and the perturbation of a problem's instance is simulated by the presence or absence of subsets of the network's set of nodes. Such considerations are particularly important for $\mathcal{NP}$-hard problems, for which the effort to get an optimal solution is important, if not prohibitive.

We next discuss one of the central themes of the conceptual approach adopted in addressing and characterizing SRPs, namely the idea of a priori optimization. In many applications, one finds that, after solving a given instance of a route optimization problem, it becomes necessary to solve repeatedly many other instances of the same problem. These other instances are usually just variations of the instance solved originally. Yet, they may be sufficiently different from that original instance to necessitate every time a re-consideration of the entire problem on the part of the analyst.

The most obvious approach in dealing with such cases is to attempt to solve optimally (or near-optimally with a good heuristic) every potential instance of the original problem. We call this approach the 're-optimization strategy' and denote it with the Greek letter Σ. The approach, however, suffers from several disadvantages. For example, if the routing problem considered is $\mathcal{NP}$-hard, one might have to solve exponentially many instances of a hard problem. Moreover, in many applications it is necessary to find a solution to each new instance quickly, but one might not have the required computing or other resources for doing so.

As an alternative, we shall also investigate here a different strategy. Rather than re-optimizing every potential instance, we wish to find an a priori solution to the original problem and then update in a simple way this a priori solution to answer each particular instance/variation. Clearly, the natural questions to ask are: What is the measure of 'effectiveness' of such an a priori solution? Once such a measure has been defined, how does one find the best a priori solution? And, how does one update the a priori solution for each particular problem instance?

The above discussion is general, in the sense that it applies to any combinatorial optimization problem. In order to address these questions concretely, we restrict our attention to a class of network routing problems. Consider then a complete graph $G = (V, E)$ on n nodes on which a routing problem is defined (for example the traveling salesman problem). If every possible subset of the node set V may or may not be present on any given instance of the optimization problem (for example, on any given day, the traveling salesman may have to visit only a subset S of the nodes in V), then there are 2^n possible instances of the problem — all the possible subsets of V. Suppose instance S has probability $p(S)$ of occurring. Given a method $\mathcal{U}$ for updating an a priori solution f to the 'full-scale' optimization problem on the original graph $G(V, E)$, $\mathcal{U}$ will then produce for problem instance S, a feasible solution $t_f(S)$ with value ('cost') $L_f(S)$. (In the case of the TSP, $t_f(S)$ would be a tour through the subset S of nodes and $L_f(S)$ the length of that tour.) Then, given that we have already selected the updating method $\mathcal{U}$, the natural choice for the a priori solution f is to select f so as to minimize the expected cost

$$E[L_f] = \sum_{S \subseteq V} p(S) L_f(S), \tag{1}$$

with the summation being over all subsets of V. In other words, we would like to minimize the 'weighted average' over all problem instances of the values $L_f(S)$ obtained by applying the updating method $\mathcal{U}$ to the a priori solution f.

This choice of a measure of effectiveness for the a priori solution f that we seek, namely the expected cost (1), gives a reasonable answer to our first question. But what properties should the updating method $\mathcal{U}$ have? The most desirable property of $\mathcal{U}$ would be for $L_f(S)$ to be 'close' to the value of the optimal solution $L_{\mathrm{OPT}}(S)$, for every instance S. A less restrictive and more global property is to require $E[L_f]$ to be 'close' to the expected cost $E[\Sigma]$, over all problem instances, of the re-optimization strategy:

$$E[\Sigma] = \sum_{S \subseteq V} p(S) L_{\mathrm{OPT}}(S). \tag{2}$$

In addition, $\mathcal{U}$ must be able to update efficiently the solution from one problem instance to the next.

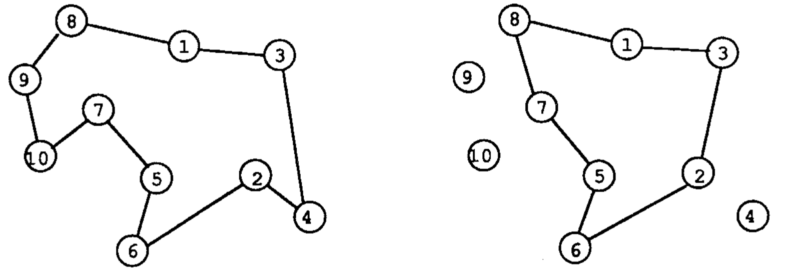
In the following definitions of the updating methods $\mathcal{U}$, the choices of $\mathcal{U}$ may initially seem arbitrary. But these choices will turn out to be natural ones. First, for every choice of $\mathcal{U}$ we are proposing, the updating of the solution to a particular instance S can be done very easily. Moreover, these updating methods are well suited for applications.

After this general discussion of the rationale behind the definitions which follow, we describe informally the problems we are considering.

The probabilistic traveling salesman problem

The probabilistic traveling salesman problem (PTSP) is perhaps the most fundamental stochastic routing problem that can be defined. It is essentially a traveling salesman problem (TSP), in which the number of points to be visited in each problem instance is a random variable.

Consider a problem of routing through a set of n known points. On any given instance of the problem only a subset S consisting of $|S| = k$ out of n points $(0 \le k \le n)$ must be visited. Suppose that the probability that instance S occurs is $p(S)$. As mentioned above, ideally we might like to re-optimize the tour for every instance, but in many cases we may not have the resources to do so or, even if we had them, re-optimization might turn out to be too time consuming. Instead, we wish to find a priori a tour through all n points. On any given instance of the problem, the k points present will then be visited in the same order as they appear in the a priori tour (see Figure 1 for an illustration). The problem of finding such an a priori tour which is of minimum length in the expected value sense is defined as the PTSP. The updating method $\mathcal{U}$ for the PTSP is therefore to visit the points



A priori tour through 10 points.

The tour when points 4, 9, and 10 need not be visited.

Fig. 1. The PTSP methodology.

on every problem instance in the same order as in the a priori tour, i.e. we simply skip those points which are not present in that problem instance.

The expectation is computed over all possible instances of the problem, i.e. over all subsets of the vertex set $V = \{1, 2, \ldots n\}$. That is, given an a priori tour τ, if problem instance $S(\subseteq V)$ will occur with probability $p(S)$ and will require covering a total distance $L_\tau(S)$ to visit the subset S of customers, that problem instance will receive a weight of $p(S)L_\tau(S)$ in the computation of the expected length. If we denote the length of the tour τ by L_τ (a random variable), then our problem is to find an a priori tour through all n potential customers, which minimizes the quantity

$$E[L_\tau] = \sum_{S \subseteq V} p(S)L_\tau(S), \tag{3}$$

with the summation being over all subsets of V.

The probabilistic vehicle routing problem

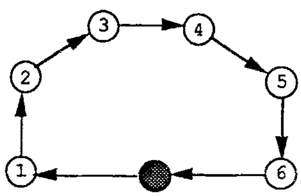
Consider a standard VRP but with demands which are probabilistic in nature rather than deterministic. The problem is then to determine a fixed set of routes of minimal expected total length, which corresponds to the expected total length of the fixed set of routes plus the expected value of extra travel distance that might be required. The extra distance will be due to the possibility that demand on one or more routes may occasionally exceed the capacity of a vehicle and force it to go back to the depot before continuing on its route.

The following two solution-updating methods can be defined: Under method $\mathcal{U}_a$ the vehicle visits all the points in the same fixed order as under the a priori tour, but serves only customers requiring service during that particular problem instance. The total expected distance traveled corresponds to the fixed length of the a priori tour plus the expected value of the additional distance that must be covered whenever the demand on the route exceeds vehicle capacity. Method $\mathcal{U}_b$ is defined similarly to $\mathcal{U}_a$ with the sole difference that customers with no demand on a particular instance of the vehicle tour are simply skipped. An example of the PVRP under both updating methods can be seen in Figure 2. In Section 2.3, we will in fact see that this problem can be further subdivided into several categories.

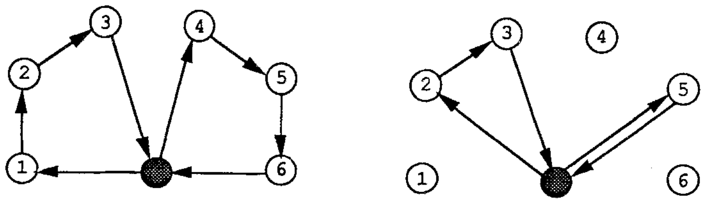
The probabilistic traveling salesman location problem

We are given a set of n nodes (customer locations) on a network. Each day a subset S of customers make a request for service with probability $p(S)$. By a specific time of each day, a service unit receives the list of calls for that day and starts a traveling salesman tour using the underlying network that visits all the customer locations in the list. The objective is to find an optimal location i for the service unit, so that the expected distance traveled

$$E[\Sigma_{\mathrm{TSLP}}(i)] = \sum_{S \subseteq V} p(S)L_{\mathrm{OPT}}(S \cup i) \tag{4}$$



(i) A prior route through 6 customers (each with a demand of 0 or 1 unit) by a vehicle of capacity 2.



Nonskipping strategy (method a). Skipping strategy (method b).

(ii) The two strategies when only customers 2, 3, and 5 have a positive demand.

Fig. 2. The PVRP methodology.

is minimized. This problem is called the traveling salesman location problem (TSLP).

The difficulty of having to compute the optimal tour for every instance can be overcome by using an a priori tour τ_p and then follow the PTSP updating method $\mathcal{U}$ described before, i.e., skip customer locations with no demand. The problem is then to find a node and an a priori tour to minimize the expected distance traveled using the PTSP approach, i.e. to minimize

$$h(i, \tau) \triangleq \sum_{S \subseteq V} p(S) L_\tau(S \cup i). \tag{5}$$

The problem of finding simultaneously an optimal location and an optimum a priori tour is called the probabilistic traveling salesman location problem (PTSLP).

The probabilistic shortest path problem

The probabilistic shortest path problem (PSPP) can be described as follows: consider the problem of finding a shortest path between a node source s and a node sink t in a complete network having a length associated with each arc. On any given instance of the problem only a subset among intermediate nodes can be used to go from s to t, the subset being chosen according to a given probability law. We wish to construct an a priori path such that, on any given instance of the problem, the sequence of nodes defining the path is preserved but only the permissible nodes are traversed, the others being skipped. The problem of finding an a priori path of minimum expected length is defined as a PSPP. The updating method $\mathcal{U}$ for the PSPP is again to visit the nodes on every problem instance in the same order as in the a priori path, skipping the nodes that cannot be used in that problem instance.

During the last decade combinatorial optimization has undoubtedly been one of the fastest growing and most exciting areas in mathematical programming. Needless to say, the related scientific literature has been expanding at a very rapid pace. Examples of particular relevance to this chapter are the three excellent review volumes on the traveling salesman problem [Lawler, Lenstra & Shmoys, 1985], on routing and scheduling [Bodin, Golden, Assad & Ball, 1983], and on vehicle routing [Golden & Assad, 1988], each of which offers several hundreds of references.

Research at the interface between probability theory and combinatorial optimization spans a period of over 30 years and in recent years has been at the center of much activity. The dominant trends of this interplay which are relevant to this paper can be summarized as follows.

Probabilistic analysis of combinatorial optimization problems in the Euclidean plane. Research in this area was initiated by the pioneering paper of Beardwood, Halton & Hammersley [1959]. After a period of more than 15 years and motivated by the significant advances in theoretical computer science, Karp [1977] used their main result to propose a partitioning heuristic, which constitutes an ϵ-approximation algorithm for the TSP in the Euclidean plane.

In the last decade, the asymptotic properties of many combinatorial optimization problems in the Euclidean plane have been investigated. The most general analysis in this direction is due to Steele [1981a], who developed the theory of subadditive Euclidean functionals to obtain very sharp limit theorems for a broad class of combinatorial optimization problems. Rates of convergence for general functionals have also recently been addressed by Jaillet [1992a] (see also Rhee & Talagrand [1988] for the TSP).

Probabilistic analysis on problems with random lengths. In the last decade there have been numerous papers dealing with the behavior of combinatorial optimization problems when the costs involved are taken from a probability distribution. Interest in this area intensified after the pioneering paper of Karp [1979] on the TSP and the attempts to explain probabilistically the success of the simplex method for linear programming. Of particular relevance to this chapter are papers on stochastic (and static) versions of the shortest path problem in which arc lengths are random [see Frank, 1969; Andreatta, Ricaldone & Romeo, 1985; Frieze & Grimmet, 1985; Hassin & Zemel, 1985; Andreatta & Romeo, 1988; and Bertsekas & Tsitsiklis, 1991].

Stochastic routing problems. In contrast to their deterministic counterparts, the professional literature on SRPs to date is very sparse. Jaillet [1985] introduced the PTSP, examined some of its combinatorial properties and proved asymptotic theorems in the plane. A summary of these results as well as a discussion on the applications of the PTSP and the PVRP are contained in Jaillet & Odoni [1988]. Bertsimas [1988] introduced the framework of a priori optimization and further studied the PTSP, PVRP as well as other problems.

Except for an isolated result in the 1970's [Tillman, 1969], VRPs with stochastic elements in their definitions have received attention only recently. Stewart & Golden [1983], Dror & Trudeau [1986], Dror, Laporte & Trudeau, [1989]; and Laporte & Louveau [1990] use techniques from stochastic programming to solve optimally small problems and find bounds for them. The definitions of these problems are different from the ones we are considering in this section.

The traveling salesman location problem has been considered by Eilon, Watson-Gandy & Christofides [1971] and Burness & White [1976], where heuristic approaches are proposed. Recently in a series of papers, Berman & Simchi-Levi [1986, 1988a, b] and Simchi-Levi & Berman [1988] solved the problem on a tree network and proposed a heuristic of relative worst error 1/2 for the general network case as well as for the Euclidean and the rectilinear metric.

The probabilistic shortest path problem was introduced in Jaillet [1988] in which a branch and bound scheme was proposed. A thorough study of the complexity as well as of some special polynomial cases of this problem is contained in Jaillet [1992b].

A final remark has to do with the relationship between network reliability theory and the class of SRPs we are considering. In network reliability theory [see for example Colbourn, 1987] the nodes are usually assumed to be always reliable and the questions addressed concern the existence of paths among pairs of nodes. In case of the SRPs the type of questions we are addressing as well as the motivation for their definition are different.

As noted earlier, SRPs could prove highly useful in many application contexts in which the explicit consideration of randomness is essential. For instance, the PTSP arises in practice whenever a company, on any given day, is faced with the problem of collections (deliveries) from (to) a random subset of some known global set of customers in an area and does not wish to or, simply, cannot redesign the tours from scratch every day. Examples in this category include a 'hot meals' delivery system described by Bartholdi, Platzman, Collins, Lee & Warden [1983], routing of forklifts in a cargo terminal or in a warehouse and, interestingly, the daily delivery of mail to homes and businesses by postal carriers everywhere. In fact it was this last application that led to the initial formulation of the PTSP in Jaillet [1985]. Jaillet & Odoni [1988] describe in considerable detail an application in a strategic planning context in which a package distribution company has decided to begin service in a particular area. After carrying out a market survey and identifying a set of potential major customers who during any single time period have a significant probability of requiring a visit, the company wishes to estimate the resources necessary to serve these customers. The PTSP then provides a model for computing approximately the expected amount of travel that will be required per time period and, by implication, the number of vehicles, drivers, etc.

In a non-routing context, PTSP models can also be of interest in many situations in which an ordering of entities of any type has to be found and that sequence has to be preserved even when some of the entities may be absent. One such example can be given from the area of job-shop scheduling: Consider the problem of loading n jobs on a machine at which a changeover cost is incurred whenever a new job is loaded. With any given ordering of the n jobs on the machine, we can then associate a total changeover cost. Any given ordering of the n jobs may also impose specific long-term requirements on the job-shop, such as a set of tasks to be performed before and after the processing of the jobs on the machine. These requirements may often be difficult to modify on a daily basis so that, if on a given day some jobs need not be processed, the relative ordering previously specified for the remaining jobs is nonetheless left unmodified. The PTSP is again relevant in analyzing such situations.

PVRPs are of course 'constrained' cases of PTSPs and thus arise in the same collection and distribution contexts as PTSPs, whenever the vehicle capacity Q becomes a practically significant issue. The capacity Q may be expressed in terms of a maximum allowable vehicle load, maximum number of stops, maximum distance per tour or some other physical or statutory limitation. For instance, in the case of the delivery of cash by a bank to a set of automatic teller machines spatially distributed throughout a city, Q might be the upper bound on the amount of money that a vehicle might carry for safety reasons. The uncertainty in this problem is due to the fact that each machine may or may not require a visit during any given time period, depending on the amount of money it dispenses. Similar applications of the PVRP can be found in most problems that combine inventory and routing considerations.

TSLPs and PTSLPs arise similarly in the complex but also very common contexts in which facility location, routing and, possibly, inventory-related decisions must be made simultaneously. Note the difference between these problems and the classical 'median' (or 'minisum') and 'center' (or 'minimax') problems in facility location theory. In the case of (P)TSLPs, once a facility is located, demands are visited through tours; therefore, the facility location problem must be 'central' relative to the *ensemble* of the demand points, as ordered by the (yet unknown) tour through all of them. By contrast, in the classical problems the facility (or facilities) must be located by considering distances to *individual* demand points, thus making the problem more tractable.

The generic PSPP, as stated, can be of interest in many applications. First consider a network in which arcs represent streets of a city, and nodes are intersections, and suppose we want to go from an origin s to a destination t along this network. The length of an arc (i, j) is defined to be the time needed to go from i to j, a value which may vary greatly. Usually, one either uses the means of the traveling times on each arc and solves a deterministic shortest path problem, or one considers the travel times as random variables and tries to solve one of many possible problems such as: finding the path with maximum probability of being the shortest, or finding the path among the shortest paths which has minimum variance, etc. (see Andreatta, Ricaldone & Romeo [1985] for a good discussion of these formulations). However, if the traffic is close to saturation, a slight perturbation of the flow can create a traffic blocking condition. In that case, we have a critical situation in which the travel time does not vary much anymore (being constantly high because of congestion) except when a blocking condition is faced (a street unexpectedly blocked for some reason, an intersection gridlocked with conflicting flows of vehicles; etc.). Depending on the frequencies of these blockings one might then select longer but less risky paths to travel from one point to another. Let us see how the PSPP offers an analytical way to choose among paths with such uncertainties. First, one can model a risky street (i, j) by adding an artificial 'probabilistic' node k that can or cannot be traversed with certain probabilities: in the first case the length of the street would be its normal travel time, in the second case it would be much higher. In the case of a risky intersection the model is even simpler: we simply model the intersection as a 'probabilistic' node. Other applications will be discussed in Section 2.5.

We now turn to a more thorough investigation of each of these problems. To keep the length of the presentation within reasonable limits, detailed derivations have been omitted. All but the more important theorem proofs are only sketchily outlined, with appropriate references given for interested readers.

2.2. *The probabilistic traveling salesman problem*

As indicated in the introduction, the PTSP is the most fundamental of the SRPs. For this reason, the PTSP will be used in this section to illustrate how the entire set of issues identified earlier can be addressed for a specific SRP.

2.2.1. *The expected length of a given PTSP tour τ*

Let us consider a PTSP defined on a given complete graph $G = (V, E)$, $|V| = n$, with a cost $d : E \to R$ and a vector $(p_1, \ldots, p_n)$ of the probabilities of presence of the vertices. To facilitate the derivation of analytic results and without loss of generality, it will be assumed henceforth that the vertices are mutually independent with regard to probability of presence.

Consider now the tour $\tau = (1, 2, \ldots, n, 1)$ and let $E[L_\tau]$ be the expected length of τ. Note that in this case the tour length L_τ can take 2^n different values, the same as the number of instances involving 'present' and 'absent' vertices. For each such instance, we would require $O(n)$ additions to compute L_τ. Thus were we to use an enumeration approach, the computational effort to compute $E[L_\tau]$ would be $O(n2^n)$ for any given tour τ. Fortunately, a much more efficient approach exists.

Theorem 1.

$$E[L_\tau] = \sum_{i=1}^{n} \sum_{j=i+1}^{n} d(i, j) p_i p_j \prod_{k=i+1}^{j-1} (1 - p_k) + \\ + \sum_{j=1}^{n} \sum_{i=1}^{j-1} d(j, i) p_i p_j \prod_{k=j+1}^{n} (1 - p_k) \prod_{k=1}^{i-1} (1 - p_k). \tag{6}$$

Sketch of the proof. This result follows directly from the following argument: the instances of τ for which $d(i, j)$ makes a contribution to $E[L_\tau]$ are those in which the vertices i and j are both present, while the vertices $i+1, \ldots, j-1$ are absent (and are thus skipped in traversing the tour). □

Theorem 1 shows that $E[L_\tau]$ can be computed in $O(n^2)$ time under very general conditions. Jaillet [1985, 1991a] gives a generalization of Theorem 1 for the case in which the independence assumption does not hold and also discusses a number of variations and extensions of the theorem.

2.2.2. Asymptotic comparison of re-optimization and a priori optimization

We turn next to the issue of characterizing and comparing the asymptotic behavior of the re-optimization and the a priori strategies for the PTSP, if the locations of the points are uniformly and independently distributed in the Euclidean plane. This comparison is important in order to assess the promise and potential usefulness of the a priori strategies. We will be quite informal; the interested reader can consult Jaillet [1992c] for a detailed and rigorous treatment of these issues, as well as for generalizations.

Let $X^{(n)} = (X_1, \ldots, X_n)$ be n points uniformly and independently distributed in the unit square. Let L^n_{TSP} be the length of the TSP defined on $X^{(n)}$.

Let $E[\Sigma^n_{\mathrm{TSP}}]$ be the expectation of the TSP solutions obtained under the re-optimization strategy defined on $X^{(n)}$.

Let $E[L^n_{\mathrm{PTSP}}]$ be the expectation of the a priori strategy, i.e. the expected length of the optimal a priori solution to the PTSP defined on $X^{(n)}$.

It is well known [see Beardwood, Halton & Hammersley, 1959] that we can characterize very sharply the solutions to the deterministic TSP.

Theorem 2. *With probability 1*

$$\lim_{n\to\infty} \frac{L^n_{\mathrm{TSP}}}{\sqrt{n}} = \beta_{\mathrm{TSP}}. \tag{7}$$

This almost sure convergence was later strengthened by Steele [1981b] to include complete convergence, i.e.,

Theorem 3.

$$\forall \varepsilon > 0, \sum_n \mathbb{P}\left(\left|\frac{L^n_{\mathrm{TSP}}}{\sqrt{n}} - \beta_{\mathrm{TSP}}\right| > \varepsilon\right) < +\infty. \tag{8}$$

We now characterize the expectation of the re-optimization strategy for the PTSP assuming that each of the n points is present with the same constant probability p, which is called the coverage probability. We remark that in the following theorem the expectation is taken over all the possible 2^n instances of the problem and the probability 1 statement refers to the random locations of the points.

Theorem 4 *With probability 1*

$$\lim_{n\to\infty} \frac{E[\Sigma^n_{\text{TSP}}]}{\sqrt{n}} = \beta_{\text{TSP}}\sqrt{p}. \tag{9}$$

Sketch of the proof. The intuitive idea in the proof is that the principal contribution to $E[\Sigma^n]$ comes from the sets S with $|S|$ close to np. The reason is that the number of points present is given by a Binomial random variable with parameters n, p and hence is almost surely asymptotically equivalent to np. In this range of $|S|$ we can apply Theorem 3 to obtain Theorem 4. Note that in Jaillet [1992c], an explanation is given on why the complete convergence of Theorem 3 is crucial for this kind of argument. □

Intuitively Theorem 4 means that solutions under the re-optimization strategy behave asymptotically similarly to those of the corresponding deterministic TSP but on np rather than n points. We next characterize asymptotically the a priori optimization strategy.

Theorem 5. *With probability 1*

$$\lim_{n\to\infty} \frac{E[L^n_{\text{PTSP}}]}{\sqrt{n}} = \beta_{\text{PTSP}}(p). \tag{10}$$

Sketch of the proof. We first prove that the PTSP belongs to the class of subadditive Euclidean functionals whose asymptotic behavior has been characterized by Steele [1981a]. Their value is almost surely asymptotic to $c\sqrt{n}$, where c depends on the functional. For a detailed proof the reader is again referred to Jaillet [1992c]. □

Comparing Theorems 4 and 5 we can observe that the a priori and re-optimization strategies have similar asymptotic behaviors almost surely. Both theorems prove the existence of a constant but without determining the value of the constant analytically; in fact, for most similar asymptotic results, the respective limiting constants are unknown and only bounds or experimental estimations have been established [see Avram & Bertsimas [1992] and Jaillet [1993] for an important exception concerning the minimum spanning tree problem]. In fact the current best known result on the relationship between $\beta_{\text{PTSP}}(p)$ and $\beta_{\text{TSP}}\sqrt{p}$ was obtained in Jaillet [1985] and is reproduced here:

$$\tfrac{5}{8}\sqrt{p} \le \beta_{\text{TSP}}\sqrt{p} \le \beta_{\text{PTSP}}(p) \le \min(\beta_{\text{TSP}},\ 0.9204\sqrt{p}). \tag{11}$$

On the other hand extensive experimental work by Johnson (1989) suggests that $\beta_{\text{TSP}} \approx 0.72$. Note that it is tempting to conjecture that $\beta_{\text{TSP}}\sqrt{p} = \beta_{\text{PTSP}}(p)$, but no correct proof exists of this result (contrary to the erroneous claim in Bertsimas, Jaillet & Odoni [1990]). Yet, the a priori strategy does seem to behave (asymptotically) equally well on average with the re-optimization strategy on Euclidean problems.

2.2.3. The complexity of a priori optimization

Having shown that, in terms of performance, a priori strategies are attractive compared with re-optimization strategies (at least for the PTSP) we now turn to the question of how difficult it is to find the optimal a priori solution from a computational complexity perspective.

We first introduce the decision version of a PTSP. Given a complete graph $G = (V, E)$, $|V| = n$, a cost $d : E \to R$, a vector $(p_1, \ldots, p_n)$ of the probabilities of presence of the vertices and a bound B, does there exist a PTSP tour f such that $E[L_f] \leq B$?

We can then characterize the complexity of the a priori strategy as follows:

Theorem 6. *The decision version of the PTSP is $\mathcal{NP}$-complete.*

Sketch of the proof. We only need to show membership in $\mathcal{NP}$, since the PTSP is a generalization of a well known $\mathcal{NP}$-complete problem [see Garey & Johnson, 1979]. Membership in $\mathcal{NP}$ is seen to hold, since, given a solution f, we can compute $E[L_f]$ in $O(n^2)$ as we have shown in Theorem 1. □

Thus, although we can compute efficiently the expected length of any given a priori solution to a PTSP, it is still $\mathcal{NP}$-hard to find an optimal a priori solution.

2.2.4. Theoretical approximations to optimal a priori solutions

In the previous section we found that it is still $\mathcal{NP}$-hard to obtain optimal a priori solutions to the PTSP. In this section we address the question of approximating the optimal a priori solution with polynomial time heuristics, whose worst case behavior we can characterize.

The first natural question to address is how heuristic approaches to the deterministic problem perform when applied to the corresponding probabilistic problem. For example, what is the performance of the well-known Christofides heuristic for the TSP [see Larson & Odoni, 1981] if applied to the PTSP? In order to find useful bounds for the routing problems (PTSP) we assume below that the triangle inequality holds. We can then prove the following:

Theorem 7. *Let L_{TSP} be the length of the optimal solution to the deterministic TSP and let L_H be the length of a heuristic solution to the same problem. Let p be the coverage probability and $E[L_{\text{PTSP}}]$ the expected length of the optimum a priori solution to the corresponding PTSP. If the heuristic has the property that $L_H/L_{\text{TSP}} \leq c$, then $E[L_H]/E[L_{\text{PTSP}}] \leq c/p$.*

Sketch of the proof. Using the triangle inequality, we know that, for any tour f, $E[L_f] \leq L_f$. In addition, we show that $E[L_{\text{PTSP}}] \geq pL_{\text{PTSP}}$. Combining these inequalities the result follows. □

Theorem 7 suggests that if the coverage probability is large then constant guarantee heuristics for the deterministic problem still behave well for the corresponding probabilistic problem. But if $p \to 0$ the bound is not informative and indeed one can find examples with $p \to 0, np \to \infty$ for which $E[L_{\text{TSP}}]/E[L_{\text{PTSP}}] \to \infty$, that is, even if $c = 1$, the optimal deterministic solution is an arbitrarily bad approximation to the optimal a priori solution. (For a number of interesting examples see Jaillet [1985, 1991a].) As an indication of the rate at which the ratio $E[L_{\text{TSP}}]/E[L_{\text{PTSP}}]$ tends to infinity, Bertsimas [1988] proves the following:

Theorem 8. *For the PTSP with triangle inequality*

$$\frac{E[L_{\text{TSP}}]}{E[L_{\text{PTSP}}]} = O(\sqrt{n}). \tag{12}$$

We next investigate the existence of constant guarantee heuristics. We restrict our attention to Euclidean problems and examine the spacefilling curve heuristic, first proposed by Platzman & Bartholdi [1989] for the Euclidean TSP. The spacefilling curve heuristic can be described as follows:

1. Given the n coordinates (x_i, y_i) of the points in the plane compute the number $f(x_i, y_i)$ for each point. The function $f : R^2 \to R$ is called the Sierpinski curve (for details on the computation of $f(x, y)$ see Bartholdi & Platzman [1982]).

2. Sort the numbers $f(x_i, y_i)$ and visit the corresponding initial points (x_i, y_i) in that order, producing a tour SF.

The key property of the spacefilling curve heuristic that makes its analysis for the PTSP possible is the following: Consider an instance S of the problem. Suppose the spacefilling curve heuristic produces a tour $SF(S)$ if we run the heuristic on the instance S. Consider now the tour SF produced by the heuristic on the original instance of the problem, i.e. when all points are present. What is the tour that the PTSP strategy would produce in instance S if the a priori tour is SF?

The answer is precisely $SF(S)$, because sorting has the property of preserving the order in which the points in S will be visited by the spacefilling curve, which is exactly the property of the PTSP strategy as well. Based on this critical observation we can then analyze the spacefilling curve heuristic.

Theorem 9. *For the Euclidean PTSP the spacefilling curve heuristic produces a tour SF with the property*

$$\frac{E[L_{\text{SF}}]}{E[L_{\text{PTSP}}]} \le \frac{E[L_{\text{SF}}]}{E[\Sigma_{\text{TSP}}]} = O(\log n). \tag{13}$$

Sketch of the proof. In Platzman & Bartholdi [1989] it is proven that the length of the spacefilling curve heuristic satisfies:

$$\frac{L_{\text{SF}}}{L_{\text{TSP}}} = O(\log n). \tag{14}$$

Consider an instance S of the problem. If the spacefilling curve heuristic is applied to the instance S, it will similarly produce a tour $SF(S)$ with length

$$\frac{L_{\text{SF}}(S)}{L_{\text{TSP}}(S)} = O(\log |S|) = O(\log n). \tag{15}$$

But since $SF(S)$ is the tour produced by the PTSP strategy at instance S then

$$\frac{E[L_{\mathrm{SF}}]}{E[\Sigma_{\mathrm{TSP}}]} = \frac{\sum_{S\subseteq V} p(S)L_{\mathrm{SF}}(S)}{\sum_{S\subseteq V} p(S)L_{\mathrm{TSP}}(S)} \leq \frac{\sum_{S\subseteq V} p(S)O(\log n)L_{\mathrm{TSP}}(S)}{\sum_{S\subseteq V} p(S)L_{\mathrm{TSP}}(S)} = O(\log n).$$

□

Note that this result does not depend on the probabilities of points being present. It holds even if there are dependencies on the presence of the points. Observe also that the spacefilling curve heuristic ignores the probabilistic nature of the problem but surprisingly produces a tour which is globally (in every instance) close to the optimal.

As a corollary to Theorem 9 we can compare the PTSP and the re-optimization strategies from a worst-case perspective. For the Euclidean PTSP, since $E[L_{\mathrm{PTSP}}] \leq E[L_{\mathrm{SF}}]$,

$$\frac{E[L_{\mathrm{PTSP}}]}{E[\Sigma_{\mathrm{TSP}}]} = O(\log n). \tag{16}$$

Platzman & Bartholdi [1989] conjecture that the spacefilling curve heuristic is a constant-guarantee heuristic. Unfortunately, Bertsimas & Grigni [1989] showed this conjecture to be false, and the existence of a constant guarantee heuristic for the Euclidean PTSP remains open.

2.3. *The probabilistic vehicle routing problem*

We shall now look at generalizations of the PTSP, which in Section 2 were introduced as the PVRP. Specifically, we still consider demands which are probabilistic in nature and our problem is to determine a priori routes of minimal expected length for vehicles with finite capacity. The complications introduced by the finite capacity of the vehicles is a major point of interest. A first problem is to consider a single vehicle and design a giant *a priori* vehicle tour through all the demand points. While covering this tour the vehicle may run out of capacity and, in such an event, it will have to return to the depot — for instance, in order to deposit the load it has picked up at the points it has already visited. Thus, the expected tour length to be minimized must also include any additional distance traveled to and from the depot whenever the vehicle reaches its capacity. There is, of course, an alternative interpretation under which the very same problem can be viewed as a multi-vehicle PVRP. This can be seen best if one sets p_i, the probability of visiting point i, equal to 1 for all i. Then the approach just described is identical to one of the two standard approaches to multi-vehicle deterministic VRPs, namely 'route first, cluster second'. Under this interpretation, the returns of the vehicle to the depot result in multiple tours, so that we are dealing with multiple-VRP tours as solutions to the overall problem. However, in the general case when some of the p_i are strictly less than 1, some criterion or criteria must be used in order to break up the giant a priori tour into clusters of customers — with each cluster served by a different vehicle.

In order to consider all these aspects in a more specific way let us now define four generalized versions of the PTSP that can be classified as PVRPs.

The capacitated probabilistic traveling salesman problem. Assume that each point x_i, requiring a visit with a probability p_i, independently of all others, has a unit demand, and that the salesman (vehicle) has a capacity q. We wish to find a priori a tour through all n points. On any given instance, the subset of points present will then be visited in the *same order* as they appear in the a priori tour. Moreover if the demand on the route exceeds the capacity of the vehicle, the salesman has to go back to the depot before continuing on his route. The problem of finding such a tour of minimum expected total length (the expected length of the tour in the PTSP sense plus the expected extra distance due to overloading) is defined as a capacitated PTSP. This problem is the 'probabilistic vehicle routing problem under updating method $\mathcal{U}_b$' as described in Section 2.1.

The m-probabilistic traveling salesmen problem (m-PTSP). Consider the problem of routing through a set of n points starting from and ending at a depot. On any given instance of the problem, only a random subset of points (each unit-demand point x_i being present with a probability p_i, independently of the others) has to be visited. We wish to find, a priori, m subtours, each starting from and ending at the depot, such that each point is included in exactly one tour, and of minimum total expected length under the skipping strategy (method $\mathcal{U}_b$). For this problem we have m vehicles with no capacity limits.

The capacitated m-probabilistic traveling salesmen problem. This capacitated vehicle problem is a natural combination of the capacitated PTSP and of the m-PTSP.

The general probabilistic vehicle routing problem. This is the same as the capacitated m-PTSP except that the demand of each customer x_i is no longer modeled by a Bernoulli random variable with parameter p_i, but rather by a more general random variable. Note that, for this problem, the two updating methods $\mathcal{U}_a$ and $\mathcal{U}_b$ of Section 2.1 are identical if the demand of each customer is strictly positive with probability one.

Several variations of this last problem have been referred to as stochastic vehicle routing problems in the existing literature [see for example Stewart & Golden, 1983; Dror & Trudeau, 1986; Dror, Laporte & Trudeau, 1989; Laporte & Louveau, 1990]. There is, however, an important difference between the approach typically adopted in these references and the one described here: These other stochastic vehicle routing problems are formulated by using techniques from stochastic programming (i.e., chance-constrained optimization, or stochastic programming with recourse) that allow one to transform these problems into deterministic VRPs, and then either solve them optimally or obtain bounds for them. One major consequence of these approaches is that it is necessary to introduce additional parameters (in the form of performance criteria) whose choice is at the analysts' discretion and may be related to routing costs only indirectly. Examples of such parameters might be: 'the probability of a vehicle having to return to the depot more than once, while serving its cluster of customers, should not exceed δ, $0 < \delta < 1$' for the chance-constrained formulation; or, 'if the vehicle reaches capacity while serving its customers a penalty α is incurred' for stochastic programming with recourse. It should be emphasized that the routing strategies described below constitute in fact, a form of stochastic programming with recourse (the recourse is to go back to the depot whenever vehicle capacity is reached), but in our case the cost of the recourse simply corresponds to extra travel distance.

We now summarize our main results on the m-PTSP, capacitated-PTSP, and capacitated m-PTSP. For details, the reader is referred to Jaillet [1987], Jaillet & Odoni [1988], Bertsimas [1988], Jaillet [1991b], and Bertsimas [1992]. To facilitate the presentation we will restrict ourselves to the case of uniform coverage probability p. We begin with some basic definitions: Let x_0 denote the location of the depot, while $x = (x_1, x_2, \ldots)$ represents an arbitrary infinite sequence of points in $\mathbf{R}^2$; $x^{(n)} = (x_1, x_2, \ldots, x_n)$ are the first n points of x. If the position of the points is random, the sequence is denoted by upper-case letters, i.e., $X = (X_1, X_2, \ldots)$. For each point x_i its distance to the depot will be written as d_i and the average distance $(\sum_{i=1}^n d_i)/n$ as $\bar{d}$. Associated with x is a sequence of i.i.d. Bernoulli random variables with parameter p describing the presence or absence of the points. $E[L_{\mathrm{PTSP}}(x^{(n)})]$, $E[L_{m\text{-PTSP}}(x^{(n)})]$, $E[L_{\mathrm{CPTSP}}(x^{(n)})]$, and $E[L_{m\text{-CPTSP}}(x^{(n)})]$ will represent respectively the optimal-solution expected lengths, through $x^{(n)}$, of the PTSP, the m-PTSP, the capacitated PTSP, and the capacitated m-PTSP. Finally, for any i and j, we will use $s(x_i, x_j)$ to denote the 'savings' quantity $d(x_i, x_0) + d(x_0, x_j) - d(x_i, x_j)$.

2.3.1. Expected length of a priori solutions

Theorem 10. *The expected length, $E[L_\tau^c]$ of the capacitated PTSP for a given tour $(x_0, x_1, x_2, \ldots, x_n, x_0)$ through $x^{(n)}$ is given by*

$$E[L_{\mathrm{CPTSP}}(x^{(n)})] = E[L_{\mathrm{PTSP}}(x^{(n)})] + \sum_{i=q}^{n-1} \sum_{j=i+1}^{n} \gamma_{ij} s(x_{\sigma(i)}, x_{\sigma(j)}), \tag{17}$$

where $E[L_{\mathrm{PTSP}}(x^{(n)})]$ is the expected length of the PTSP tour through $x^{(n)}$, and where

$$\gamma_{ij} = \sum_{k=1}^{\lfloor i/q \rfloor} \binom{i-1}{kq-1} p^{kq+1}(1-p)^{j-kq-1}. \tag{18}$$

Sketch of the proof. The expected length of the capacitated PTSP is the sum of two terms: The expected length of the PTSP, plus the expected value of extra travel distance that might be required due to limited capacity. For the first term, the reader is referred to Jaillet [1985, 1991a] for a detailed expression (i.e., a generalization of Theorem 1 which includes the presence of a depot). For the second term, note that γ_{ij} is the probability that the vehicle reaches capacity at point x_i and that the next point present along the tour is x_j (in that case the extra distance is simply $s(x_i, x_{i+r+1})$). $\square$

Expressions (17) and (18) show that the objective function of the capacitated PTSP can also be obtained in $O(n^2)$ steps. The expected lengths for the m-PTSP and the capacitated m-PTSP are easily obtained by applying Theorems 1 and 10, respectively, to each of the m subtours separately.

The following theorem establishes some relationships between the values of the optimal solutions to the various types of PVRPs. For each n, we sort the distances from the depot in a non-decreasing order $(d_{(1)}, d_{(2)}, \ldots, d_{(n)})$ (i.e., $d_{(j)} \leq d_{(j+1)}$ for $j \in [1, \ldots, n-1]$).

Theorem 11. *The optimal-solution expected lengths of the m-PTSP, capacitated PTSP, and capacitated m-PTSP are related to the expected length of the optimal PTSP as follows.*

(i) m-PTSP:

$$\max\left\{E[L_{\mathrm{PTSP}}(x^{(n)})];\ 2p\Big(\sum_{j=1}^{m-1} d_{(j)} + d_{(n)}\Big)\right\} \leq$$
$$\leq E[L_{m\text{-PTSP}}(x^{(n)})] \leq E[L_{\mathrm{PTSP}}(x^{(n)})] + 2p\sum_{j=1}^{m-1} d_{(j)}. \tag{19}$$

(ii) Capacitated PTSP:

$$\max\left\{E[L_{\mathrm{PTSP}}(x^{(n)})];\ \frac{2p}{q}n\bar{d}\right\} \leq\leq E[L_{\mathrm{CPTSP}}(x^{(n)})] \leq$$
$$\leq E[L_{\mathrm{PTSP}}(x^{(n)})] + 2p(1-(1-p)^{n-q})\sum_{j=q}^{n-1} d_{(j+1)}. \tag{20}$$

(iii) Capacitated m-PTSP:

$$E[L_{m\text{-CPTSP}}(x^{(n)})] \geq \max\left\{E[L_{\mathrm{PTSP}}(x^{(n)})];\ 2p\sum_{j=1}^{m-1} d_{(j)} + \frac{2p}{q}\sum_{j=m}^{n} d_{(j)}\right\}, \tag{21}$$

and

$$E[L_{m\text{-CPTSP}}(x^{(n)})] \leq E[L_{\mathrm{PTSP}}(x^{(n)})] +$$
$$+ 2p\sum_{j=1}^{m-1} d_{(j)} + 2p(1-(1-p)^{n-q-m+1})\sum_{j=q}^{n-m} d_{(j+m)}. \tag{22}$$

Sketch of the proof. (i) A feasible solution to the m-PTSP can consist of having each of $m-1$ vehicles visit a single point each and the last vehicle go through the remaining $n-m+1$ points in an optimal PTSP manner. The upper bound on $E[L_{m\text{-PTSP}}(x^{(n)})]$ then follows from the fact that the PTSP functional is monotone and that the expected length of the tour a vehicle which visits a single point, say x_j, is $2pd_j$; if we take the single points to be the $m-1$ closest to the depot, we obtain the best upper bound of this form. The first lower bound on $E[L_{m\text{-PTSP}}(x^{(n)})]$ is obvious, while the second one follows from the fact that, in an optimal solution to the m-PTSP, the expected length of each one of the m subtours is greater than the expected distance needed to visit the farthest point on the subtour. This, in turn, is greater than the bound given in (19).

(ii) A feasible solution to the capacitated PTSP is given by an optimal PTSP tour. To obtain the upper bound on $E[L_{\text{CPTSP}}(x^{(n)})]$ we then have to bound the expected length of the extra distance to the depot for an optimal PTSP tour (see Jaillet 1991b] for details). The first lower bound on $E[L_{\text{CPTSP}}(x^{(n)})]$ is obvious. The second one follows from the fact that, on each instance of the problem, if the set V of points present is of cardinality k, the vehicle will cover $\lceil k/q \rceil$ subtours, each consisting of less than q points. The length of such a subtour will then be greater or equal to $2(\sum_{i \in \text{subtour}} d_i)/q$. The lower bound in this instance is then $2(\sum_{i \in V} d_i)/q$. By summing over all possible instances (weighted by their probability of presence) we finally obtain the desired result.

(iii) The bounds for $E[L_{m\text{-CPTSP}}(x^{(n)})]$ are obtained by combining the previous arguments on $E[L_{m\text{-PTSP}}(x^{(n)})]$ and $E[L_{\text{CPTSP}}(x^{(n)})]$. □

We also note that there are no clear relations between $E[L_{m\text{-CPTSP}}(x^{(n)})]$ and $E[L_{\text{CPTSP}}(x^{(n)})]$. In addition, although $E[L_{\text{PTSP}}(x^{(n)})]$ and $E[L_{m\text{-PTSP}}(x^{(n)})]$ are monotone functionals, this is in general not true for $E[L_{\text{CPTSP}}(x^{(n)})]$ and $E[L_{m\text{-CPTSP}}(x^{(n)})]$.

2.3.2. *Asymptotic results*

The main objective of this asymptotic analysis is to obtain *strong* limit laws for PVRPs similar to the PTSP results given in Section 2.2.2. In order to be consistent with Section 2.2.2, we will write, hereafter, for all the expected lengths, $E[L^n_{-}]$ for $E[L_{-}(X^{(n)})]$, whenever $X = (X_1, X_2, \ldots)$ is a sequence of points independently and uniformly distributed over $[0, 1]^2$.

The m-PTSP

Let us assume that m is a non-decreasing (possibly constant) function of n, and let us consider the asymptotic behavior of $E[L^n_{m\text{-PTSP}}]$. We have the following theorem.

Theorem 12. *Consider an infinite sequence of points independently and uniformly distributed over* $[0, 1]^2$ *and let* p *be the coverage probability for each point. For any position of the depot within a finite distance* μ *of the unit square (the depot can possibly be inside the square, i.e.* $\mu = 0$*) we have:*

(i) If $m = o(\sqrt{n})$*, or if* $m = \Theta(\sqrt{n})$ *and* $\mu = 0$*, then*

$$\lim_{n\to\infty} \frac{E[L^n_{m\text{-PTSP}}]}{\sqrt{n}} = \beta_{\text{PTSP}}(p) \; (a.s.). \tag{23}$$

(ii) If $m = \Omega(\sqrt{n})$ and $m = o(n)$, then

$$\lim_{n\to\infty} \frac{E[L^n_{m\text{-PTSP}}]}{m} = 2p\mu \ (a.s.). \tag{24}$$

(iii) If $\lim_{n\to\infty} m/n = \alpha$, $0 < \alpha \leq 1$, then

$$\lim_{n\to\infty} \frac{E[L^n_{m\text{-PTSP}}]}{m} = \frac{2p}{\alpha} \int_0^{\alpha} F_D^{-1}(x)\,\mathrm{d}x \ (a.s.). \tag{25}$$

where $F_D^{-1}(x)$ is the generalized inverse of the cumulative distribution function of D, the random variable representing the distance from a random point of the unit square to the depot.

In order to prove this theorem we need a result which, because of its independent interest, we present in a general setting. The reader is referred to Jaillet [1991b] for details.

Lemma 1. *Let $(Y_i)_i$ be a sequence of i.i.d. absolutely continuous (with no singular part) real random variables with bounded support $[a, b]$. For each n, let $(Y_{(i,n)})_{1\leq i\leq n}$ be the corresponding order statistics (i.e., $Y_{(i,n)} \leq Y_{(i+1,n)}$ for $i \in [1..n-1]$). Let s be a non-decreasing integer valued function of n, $s \leq n$. Finally let F_Y be the common cumulative distribution function of the Y_i's. We then have:*

(i) If $s = o(n)$, then

$$\lim_{n\to\infty} \frac{1}{s} \sum_{i=1}^{s} Y_{(i,n)} = a \ (a.s.). \tag{26}$$

(ii) If $\lim_{n\to\infty} s/n = \alpha$, $0 < \alpha \leq 1$ then

$$\lim_{n\to\infty} \frac{1}{s} \sum_{i=1}^{s} Y_{(i,n)} = \frac{1}{\alpha} \int_0^{\alpha} F_Y^{-1}(x)\,\mathrm{d}x \ (a.s.). \tag{27}$$

Sketch of the proof of Theorem 12. For case (i) we have from (19)

$$E[L^n_{\mathrm{PTSP}}] \leq E[L^n_{m\text{-PTSP}}] \leq E[L^n_{\mathrm{PTSP}}] + 2p\,\frac{\sum_{j=1}^{m-1} D_{(j)}}{m-1}(m-1). \tag{28}$$

If $m = o(\sqrt{n})$, the result follows from the fact that the D_i's are bounded and from Theorem 5. If $m = \Theta(\sqrt{n})$ and $\mu = 0$, the result follows from Lemma 1 part (i) and from Theorem 5.

For case (ii) and case (iii) we have from (19)

$$2p\,\frac{\sum_{j=1}^{m-1} D_{(j)} + D_{(n)}}{m} \leq \frac{E[L^n_{m\text{-PTSP}}]}{m}, \tag{29}$$

and

$$\frac{E[L^n_{m\text{-PTSP}}]}{m} \le \frac{(E[L^n_{\mathrm{PTSP}}]/\sqrt{n})\sqrt{n}}{m} + 2p\,\frac{\sum_{j=1}^{m-1} D_{(j)}}{m}. \tag{30}$$

The results follow respectively from Lemma 1 part (i) and part (ii) and from Theorem 5. □

Note that in the case where $\lim_{n\to\infty} m/\sqrt{n} = \delta$ and $\mu \neq 0$, we have, with probability one, from Theorem 5, Lemma 1 and (19):

$$\begin{aligned}\max\{\beta_{\mathrm{PTSP}}(p); 2p\mu\} &\le \liminf_{n\to\infty} \frac{E[L^n_{m\text{-PTSP}}]}{\sqrt{n}} \\ &\le \limsup_{n\to\infty} \frac{E[L^n_{m\text{-PTSP}}]}{\sqrt{n}} \le \beta_{\mathrm{PTSP}}(p) + 2p\mu.\end{aligned} \tag{31}$$

The capacitated versions

Consider an infinite sequence of points independently and uniformly distributed over $[0, 1]^2$ and let p be the coverage probability for each point. For any position of the depot within a finite distance μ of the unit square (including possibly inside the square, i.e. $\mu = 0$) we have the following bounds.

$$\frac{2p}{q}E[D] \le \liminf_{n\to\infty} \frac{E[L^n_{\mathrm{CPTSP}}]}{n} \le \limsup_{n\to\infty} \frac{E[L^n_{\mathrm{CPTSP}}]}{n} \le 2pE[D] \text{ (a.s.)}. \tag{32}$$

These inequalities follow directly from Theorem 5, Theorem 11, and the strong law of large numbers.

The asymptotic properties of these capacitated problems will be characterized more sharply in the next subsection in conjunction with algorithmic procedures.

2.3.3. Algorithmic considerations

All PVRPs that are being considered in this section are obviously $\mathcal{NP}$-hard. One is therefore justified in looking for approximations to the optimal solutions.

Our goal is to present some examples of worst-case and probabilistic analyses of heuristics for the PTSP, m-PTSP, capacitated PTSP, and capacitated m-PTSP, drawing on results presented earlier.

The PTSP

As mentioned in Section 2.2 several 'good' heuristics have been proposed for the PTSP but none of them has been so far amenable to a theoretical analysis which would prove that the heuristic provides a 'constant guarantee' (in the worst-case sense). Thus, unlike the case of the TSP, the existence of a constant-guarantee heuristic for the PTSP is still an open problem. As we will now see this is an important question since most of the worst-case analyses of the PVRPs depend heavily on the analysis of the PTSP.

For the analysis of the PVRPs we will nevertheless assume that it is possible to obtain optimal or near-optimal solutions to the PTSP. Any progress on the PTSP will automatically translate into progress on the more complicated probabilistic vehicle routing problems.

The m-PTSP

In contrast to the relationship between the TSP and the m-TSP, the m-PTSP cannot be transformed into an equivalent PTSP. Nevertheless the heuristic which consists of assigning the first $m-1$ vehicles to the $m-1$ closest points to the depot, and the last vehicle to the remaining points can be analyzed mathematically with the help of the previous sections. Indeed, if $H(x^{(n)})$ is the value given by this heuristic we have from Theorem 11:

$$\frac{H(x^{(n)})}{E[L_{m\text{-PTSP}}(x^{(n)})]} \leq \frac{E[L_{\mathrm{PTSP}}(x^{(n)})] + 2p\sum_{j=1}^{m-1} d_{(j)}}{\max\left\{E[L_{\mathrm{PTSP}}(x^{(n)})];\ 2p\sum_{j=1}^{m-1} d_{(j)}\right\}} \leq 2. \tag{33}$$

If instead of having an optimal PTSP tour we have an alternative tour τ such that $E[L_\tau(x^{(n)})] \leq \kappa E[L_{\mathrm{PTSP}}(x^{(n)})]$ then the new heuristic, say $\mathcal{H}_{alt}$, is such that

$$\frac{H_{alt}(x^{(n)})}{E[L_{m\text{-PTSP}}(x^{(n)})]} \leq \frac{\kappa E[L_{\mathrm{PTSP}}(x^{(n)})] + 2p\sum_{j=1}^{m-1} d_{(j)}}{\max\left\{E[L_{\mathrm{PTSP}}(x^{(n)})];\ 2p\sum_{j=1}^{m-1} d_{(j)}\right\}} \leq \kappa + 1. \tag{34}$$

Also for any relation between m and n, we can show, from Theorem 12, that such a heuristic is asymptotically optimal with probability one for points uniformly and independently distributed over the unit square.

The capacitated PTSP

If we denote by $H^c(x^{(n)})$ the value of the heuristic, say $\mathcal{H}^c$, which consists of simply using the optimal PTSP tour for this problem, we have, from Theorem 11:

$$\frac{H^c(x^{(n)})}{E[L_{\mathrm{CPTSP}}(x^{(n)})]} \leq$$

$$\leq \frac{E[L_{\mathrm{PTSP}}(x^{(n)})] + 2p(1-(1-p)^{n-q})\sum_{j=q}^{n-1} d_{(j+1)}}{\max\left\{E[L_{\mathrm{PTSP}}(x^{(n)})];\ (2p/q)n\bar{d}\right\}} \leq q+1, \tag{35}$$

which is not very good especially when q is big. We can also see that (35) does not allow us to derive an asymptotically optimal algorithm. Let us analyze an improvement of this algorithm, introduced in Bertsimas [1988] for a simpler version of this problem. This heuristic, based on ideas contained in Haimovich, Rinnooy Kan & Stougie [1988], is the following:

Cyclic heuristic

Step 1. Find an optimal PTSP tour $t_1 = (x_0, x_{\sigma^*(1)}, x_{\sigma^*(2)}, \ldots, x_{\sigma^*(n)}, x_0)$ and consider the tours $t_k^* = (x_0, x_{\sigma^*(k)}, x_{\sigma^*(k+1)}, \ldots, x_{\sigma^*(n)}, x_{\sigma^*(1)}, \ldots, x_{\sigma^*(k-1)}, x_0)$ for $k \in [2..n]$.

Step 2. Compute the objective values $E^c_{t_k}$ for $k \in [1..n]$.

Step 3. Take the tour with the minimum objective value.

Let $IH^c(x^{(n)})$ be the value of this improved heuristic. We then have the following [see Jaillet, 1991b]:

Theorem 13.

$$IH^c(x^{(n)}) \le \left(1 - \frac{1}{n}\right) E[L_{\mathrm{PTSP}}(x^{(n)})] + 2\left(1 + \frac{np}{q}\right)\bar{d}. \tag{36}$$

In conclusion, from Theorems 11 and 13 we have the following result:

$$\frac{IH^c(x^{(n)})}{E[L_{\mathrm{CPTSP}}(x^{(n)})]} \le$$
$$\le \frac{(1-1/n)E[L_{\mathrm{PTSP}}(x^{(n)})] + 2(1+np/q)\bar{d}}{\max\{E[L_{\mathrm{PTSP}}(x^{(n)})];\ (2p/q)n\bar{d}\}} \le 2 + \frac{1}{n}\left(\frac{q}{p} - 1\right). \tag{37}$$

When n goes to infinity, we then have a constant-guarantee (independent of q) heuristic. If instead of having an optimal PTSP tour, we have a tour τ such that $E[L_\tau(x^{(n)})] \le \kappa E[L_{\mathrm{PTSP}}(x^{(n)})]$ then this alternative heuristic gives the following bound:

$$\frac{IH^c_{alt}(x^{(n)})}{E[L_{\mathrm{CPTSP}}(x^{(n)})]} \le \kappa + 1 + \frac{1}{n}\left(\frac{q}{p} - 1\right). \tag{38}$$

Moreover, with the help of the cyclic heuristic, one can get the asymptotic behavior of the capacitated PTSP. Indeed, the following theorem, obtained from Theorem 13 and equation (32), shows that the cyclic heuristic is asymptotically optimal with probability one.

Theorem 14. *Consider an infinite sequence of points independently and uniformly distributed over $[0, 1]^2$ and let p be the coverage probability for each point. For any position of the depot within a finite distance μ of the unit square (including possibly inside the square, i.e. $\mu = 0$) we have:*

$$\lim_{n\to\infty} \frac{E[L^n_{\mathrm{CPTSP}}]}{n} = \lim_{n\to\infty} \frac{IH^c(x^{(n)})}{n} = \frac{2p}{q} E[D] (a.s.). \tag{39}$$

The capacitated m-PTSP

Let $IH_m^c(x^{(n)})$ be the value of the heuristic, here called the m-cyclic heuristic, which consists of assigning the first $m-1$ vehicles to the $m-1$ closest points to the depot, and using the cyclic heuristic on the remaining $n-m+1$ points. Then, letting $n_1 = n-m+1$, we have, from the same arguments as before, the following bound:

$$\frac{IH_m^c(x^{(n)})}{E[L_{m\text{-CPTSP}}(x^{(n)})]} \leq 2 + \frac{1}{n_1}\left(\frac{q}{p} - 1\right). \tag{40}$$

One can finally show the asymptotic optimality of the m-cyclic heuristic [see Jaillet, 1991b]

Theorem 15. *Consider an infinite sequence of points independently and uniformly distributed over* $[0,1]^2$ *and let* p *be the coverage probability for each point. For any position of the depot within a finite distance* μ *of the unit square (including possibly inside the square, i.e.* $\mu = 0$*) we have:*

(i) If $m = o(n)$*, then*

$$\lim_{n\to\infty} \frac{E[L^n_{m\text{-CPTSP}}]}{n} = \lim_{n\to\infty} \frac{IH_m^c(X^{(n)})}{n} = \frac{2p}{q} E[D] \ (a.s.). \tag{41}$$

(ii) If $\lim_{n\to\infty} m/n = \alpha$*,* $0 < \alpha \leq 1$*, then*

$$\lim_{n\to\infty} \frac{E[L^n_{m\text{-CPTSP}}]}{n} = \lim_{n\to\infty} \frac{IH_m^c(X^{(n)})}{n}$$

$$= 2p\left(1 - \frac{1}{q}\right) \int_0^\alpha F_D^{-1}(x)\,\mathrm{d}x + \frac{2p}{q} E[D] \ (a.s.). \tag{42}$$

2.3.4. More general probabilistic vehicle routing problems

We have discussed here cases of unit demand at each demand point. It is in fact possible to consider the case of unequal demands in much the same way as in Haimovich, Rinnooy Kan & Stougie [1988; see for example Bertsimas, 1992]. Following Jaillet [1987], one can also consider more general probability distributions for the demand at each point, such as binomial distributions, and re-derive most of the previous results. Finally in our multi-vehicle models, we did not consider the importance of balanced routes. If this is an issue, modifications of several heuristics proposed in Haimovich, Rinnoy Kan & Stougie [1988] could be analyzed successfully for a probabilistic environment, as well.

2.4. The probabilistic traveling salesman location problem

2.4.1. Algorithmic issues

The traveling salesman location problem (TSLP) and its generalization, the probabilistic traveling salesman location problem (PTSLP), were defined in Section 2.1. Note that the difference between the two problems is that in the TSLP it is assumed that an a priori tour through the n probabilistic nodes of the network where potential demands reside is already given at the outset. The only issue is to locate a facility that will minimize the expected length of any particular instance of this a priori tour, if demands are always visited in the same order in which they appear on the a priori tour. By contrast, in the PTSLP both the facility location *and* the a priori tour must be found.

It is interesting that, until the recent work of Berman & Simchi-Levi [1986, 1988a, b] problems that involve simultaneously location and routing considerations were considered practically intractable by locational theorists [see, e.g., Burness & White, 1976] at the same time when other facility location problems on networks, such as 'median' and 'center' problems [Mirchandani & Francis, 1990], were being studied extensively.

Berman & Simchi-Levi [1986] showed initially that the TSLP can be solved as a discrete combinatorial optimization problem:

Lemma 2. *At least one optimal facility location for the TSLP is a node.*

Sketch of the proof. The proof of this lemma is easy and follows the line of the classical Hakimi-like proofs of node-optimality for median location problems [Hakimi, 1964]. Lemma 2 can be readily extended to the PTSLP. □

It follows that the TSLP on trees can be solved efficiently:

Theorem 16. *The TSLP on a tree network can be solved in* $O(n)$ *time.*

Sketch of the proof. The proof of Theorem 16 is based on showing that a 'majority rule' [Goldman, 1974] holds for the TSLP on trees. Specifically, suppose we identify any edge (i, j) of the tree and suppose we 'cut' this edge. The tree is then subdivided into two subtrees, one containing node i and the other node j. It can then be shown that the optimal facility location must be in the 'majority' subtree, i.e., the subtree which contains the set of nodes whose total probability of presence, summed over all possible instances of the problem, is the larger of the two. Once again, Theorem 16 can be readily extended to the PTSLP. □

For a general network, we note first that both the TSLP and the PTSLP are, in general $\mathcal{NP}$-hard problems. Berman & Simchi-Levi [1988b] have therefore proposed a surprisingly simple, constant-guarantee heuristic for the case in which calls from demand nodes occur independently, with the probability of a call from node i being equal to p_i. Let $d(i, j)$ denote the shortest distance between nodes i and j. Then the heuristic can be described as follows:

1. For each node i on the network do the following:
   (a) Sort all the n nodes j_1, j_2, j_n such that $d(i, j_1) \leq \cdots \leq d(i, j_n)$. [Note that $j_1 = i$.]
   (b) Then compute $f(i) = \sum_{k=2}^{n} d(i, j_k) p_{j_k} \prod_{m=1}^{k-1} (1 - p_{j_m})$.
2. Select the node i with the minimum value of $f(i)$ as the solution to the TSLP.

Given a complete minimum distance matrix, Step 1 above takes $O(n \log n)$ time and thus the entire algorithm takes $O(n^2 \log n)$ time.

In Berman & Simchi-Levi [1988b] it is shown that this heuristic has a relative worst-case performance guarantee of 1/2, i.e., it can be at most 50% off from the optimal TSLP solution. However, it was shown later by Bertsimas that this constant-guarantee bound can be improved to

$$\tfrac{1}{2}(1 - p_{i^*}) \leq \tfrac{1}{2}(1 - \min_i p_i) \tag{43}$$

where i^* is the true optimal location for the TSLP. (For an elegant proof of this result see Bertsimas [1989].) For example, for the special case in which, for all i, $p_i = p$, $p = 1/2$ implies a relative worst-case error of 25% instead of 50%.

No constant-guarantee heuristic is available, on the other hand for the PTSLP — a more difficult problem. Bertsimas [1989] has characterized the worst case performance of a nearest neighbor location heuristic and of a spacefilling curve location heuristic and showed that both provide a worst-case performance bound which is $O(\log n)$.

The discerning reader may also have already observed that any node i with $p_i = 1$ is an optimal location for both the TSLP and the PTSLP. (If a node is to be visited on every instance of the problem anyway, we are assured that there will be no extra-distance penalty, if the facility is also located on that node!) This observation, however, is true only if the distance matrix satisfies the triangular inequality.

Finally, it should be noted that only recently have some more formal attempts been made — by using large-scale mathematical programming formulations — to obtain optimal or heuristic solutions to stochastic problems, like the TSLP and the PTSLP, which combine locational and routing considerations [see Laporte, 1988, and Laporte, Louveau & Mercure, 1989].

2.4.2. Asymptotic results

Sharp asymptotic performance results have also been obtained for the TSLP and the PTSLP. For example, it is easy to show, drawing on Theorems 2 and 4, that for $p_i = p$ and demands uniformly distributed in the unit square, we have [Berman & Simchi-Levi, 1988b]:

$$\lim_{n\to\infty} \frac{E[\Sigma^n_{\mathrm{TSLP}}]}{\sqrt{n}} = \beta_{\mathrm{TSP}}\sqrt{p}, \tag{44}$$

where $E[\Sigma^n_{\mathrm{TSLP}}]$ is the expected length of the TSLP solution obtained under the re-optimization strategy. The limit in (44) is identical to that of Theorem 4, a result which is not surprising if one considers the fact that, in the limit, the location of the TSLP facility is not important when the number of potential demands to be visited is arbitrarily large. Interestingly Bertsimas [1989] has shown that the asymptotic behavior of the TSLP heuristic described earlier in this section is exactly the same as in (44). Thus the heuristic not only provides a constant-guarantee performance but is also asymptotically optimal.

For the PTSLP, by analogy to Theorem 5, we have [Bertsimas, 1989]:

$$\lim_{n\to\infty} \frac{E[\Sigma^n_{\text{PTSLP}}]}{\sqrt{n}} = \beta_{\text{PTSP}}(p). \tag{45}$$

2.4.3. A different class of stochastic facility location problems

To conclude this section, we note that an extensive literature also exists on a different set of stochastic facility location problems which, however do not involve any routing considerations. These problems are characterized by demands which appear at random times and random locations on a network and are served by stationary facilities or traveling vehicles that can serve only one demand at a time. Coupled with service times whose duration is a random variable, these conditions generate queueing phenomena. Thus, in this set of problems 'optimal facility (or vehicle) locations' are typically defined to be those that minimize the expected total time that elapses between the occurrence of a demand and the completion of service to that demand. Note that this total time includes any time spent waiting for a facility or vehicle to become available. Comprehensive reviews of this class of facility location problems are provided in Odoni [1987] and in Batta, Berman, Chiu, Larson & Odoni [1990].

2.5. *The probabilistic shortest path problem*

This section will be drawn to a large extent from Jaillet [1992b]; it will give non-technical review of the main results obtained on this problem.

2.5.1. Notation and assumptions

Let $G = (N, A, d)$ denote a complete, loopless, directed, weighted graph where N is the node set, A is the set of arcs joining the nodes of N, and d is a function $A \mapsto R$. We consider a node source s, a node sink t and paths from s to t. A path c will be given by the sequence of nodes defining it, i.e., $c = (s, n_1, n_2, \dots, n_k, t)$. The set of nodes N is partitioned into two subsets N_1 and N_2:

N_1 is the set of nodes which never experience failure ('black' nodes), of cardinality $|N_1| = m$ ($m \geq 2$ since s and t belong to N_1).

N_2 is the set of nodes with possible failure ('white' nodes), of cardinality $|N_2| = n$.

We are given a probability law $\mathbb{P}$ on Ω, the power set of N_2 (an instance $\omega \in \Omega$ defines the subset of white nodes with no failure on this particular instance). We *restrict* $\mathbb{P}$ to be such that all outcomes of equal cardinality have the same probability of occurring, i.e.:

$$\forall\omega_1 \in \Omega,\ \forall\omega_2 \in \Omega,\ |\omega_1| = |\omega_2| \Rightarrow \mathbb{P}(\omega_1) = \mathbb{P}(\omega_2). \tag{46}$$

If W is the random variable that represents the number of white nodes with no failure we have :

$$\Pr(W = |\omega|) = \binom{n}{|\omega|}\mathbb{P}(\omega). \tag{47}$$

Hence, our probabilistic models can be specified equivalently by either giving the probability $\mathbb{P}$ or the probability distribution of W. Note also that the restriction imposed on $\mathbb{P}$ implies that, given $W = k$, the k nodes are selected uniformly at random among the set of n nodes; any probability $\mathbb{P}$ satisfying (46) will then be said to be *node-invariant*.

One important specific example (hereafter named $\mathbb{P}_1$) is

$$\mathbb{P}_1(\omega) = p^{|\omega|}(1-p)^{n-|\omega|}, \tag{48}$$

which corresponds to the case in which every white node has a probability p of being present, independently of the others; we then speak informally of a Bernoulli process with parameter p.

For a given a priori path c between s and t, the length L_c covered in traversing the nodes without failure on each instance of the problem is a random variable. The general PSPP can then be stated as follows.

Problem PSPP. Given a network $G = (N, A, d)$ with node source s and node sink t and a probability $\mathbb{P}$ find an a priori path c of minimum expected length, $E[L_c]$.

2.5.2. The expected length of a given path

The most general result, based on an extension of results obtained for the PTSP is the following:

Theorem 17. *Given a node-invariant PSPP and a path $c = (0, 1, \ldots, k, k+1)$ from s to t (by convention $0 \equiv s$ and $k+1 \equiv t$) we have:*

$$E[L_c] = \sum_{r=0}^{k-2} \alpha_r A_c^{(r)} + \sum_{r=0}^{k-1} \beta_r B_c^{(r)} + \gamma_k C_c^{(k)}, \tag{49}$$

where:

$$\begin{aligned}
A_c^{(r)} &= \sum_{i=1}^{k-1-r} d_c(i, i+r+1),\\
B_c^{(r)} &= d_c(0, r+1) + d_c(k-r, k+1),\\
C_c^{(k)} &= d_c(0, k+1),\\
\alpha_r &= \sum_{j=r}^{n-2} \left[\binom{n-2-r}{j-r} \Big/ \binom{n}{j}\right] \Pr(W = n-j), \ \textit{for all } r \in [0..k-2]\\
\beta_r &= \sum_{j=r}^{n-1} \left[\binom{n-1-r}{j-r} \Big/ \binom{n}{j}\right] \Pr(W = n-j), \ \textit{for all } r \in [0..k-1]\\
\gamma_k &= \sum_{j=k}^{n} \left[\binom{n-k}{j-k} \Big/ \binom{n}{j}\right] \Pr(W = n-j),
\end{aligned}$$

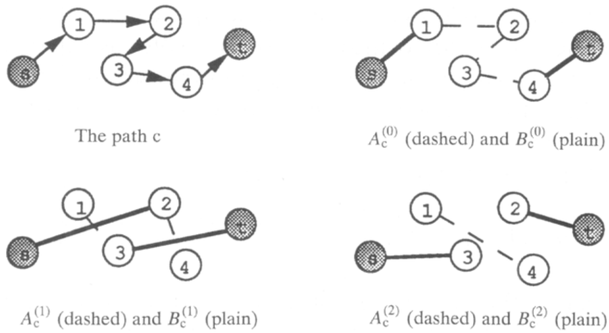


Fig. 3. A path and the arcs representing some A_c's and B_c's.

with the following convention: $d_c(i, i + r + 1) = \sum_{e=0}^{s} d(b_e, b_{e+1})$ *where* $b_0 \equiv i$, $b_{s+1} \equiv i + r + 1$, *and* $(b_1, \ldots, b_s)$ *is the sequence of black nodes drawn from* $(i + 1, \ldots, i + r)$.

Sketch of proof. On any given instance of the problem, the arc $(i, i + r + 1)$ is in the resulting path if and only if the nodes i and $i + r + 1$ are working, and the nodes $i + 1, \ldots, i + r$ are not working. The definitions of A_c's, B_c's, and C_c^k are then based on a regrouping of arcs with equal probabilities; this is illustrated in Figure 3 on a simple example. □

Note that when $k = 1$ or when $k = 0$, equation (49) gives respectively $E[L_c] = \beta_0 B_c^{(0)} + \gamma_1 C_c^{(1)}$ and $E[L_c] = \gamma_0 C_c^{(0)}$. The closed form expression (49) computes the expected length of a path from s to t through k intermediate nodes in $O(k^2)$ elementary operations (for a general, *node-invariant* $\mathbb{P}$ and assuming that the α_r's have been previously computed). Thus, Theorem 17 also shows that the recognition version of this problem belongs to the class $\mathcal{NP}$. Finally, for the case of a Bernoulli process $\mathbb{P}_1$, $\alpha_r = p^2(1 - p)^r$, $\beta_r = p(1 - p)^r$, and $\gamma_k = (1 - p)^k$.

2.5.3. The complexity of the PSPP and its relationship to the SPP

The SPP is a special case of the PSPP in which all nodes are black; it is then natural to investigate the possible links between the two problems. We show in this section that the two problems are fundamentally different so that the PSPP requires entirely new solution procedures. The following results are easily proved and confirm our earlier observations regarding the drastic changes that randomness induces into well-known combinatorial problems.

1. The PSPP is $\mathcal{NP}$-hard. Indeed consider the special Bernoulli case in which we have only two black nodes (s and t). The expected length of a path containing k ($k \leq n$) white nodes depends on $d(s, t)$ via a weight equal to $(1 - p)^k$. Since $(1 - p)^k \geq (1 - p)^n$, if we take d(s,t) arbitrarily large (everything else being equal), we can force the potential candidate paths for the corresponding PSPP

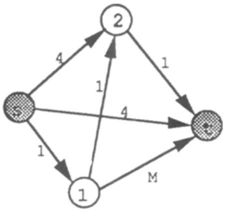


Fig. 4. The optimal deterministic shortest path can be bad.

to go through all the n white nodes. But this last problem is the probabilistic Hamiltonian path problem which is $\mathcal{NP}$-hard (being equivalent to the PTSP, see Jaillet [1985]).

2. Suppose that the distances do not satisfy the triangular inequality, since otherwise the optimal PSPP path would simply be the arc (s, t). Under that condition it is easy to construct examples in which an optimal SPP path is *arbitrarily* bad for the corresponding PSPP. See for example Figure 4 in which all arcs not shown are of length 4 and in which $M > 4$. The optimal SPP path $(s, 1, 2, t)$ of length 3 has an expected length depending on M, the length of arc $(1, t)$, which is traversed when node 1 works and node 2 fails; the expected length of $(s, 1, 2, t)$ can then be made arbitrarily large as compared to the expected length of path (s, t) (of value 4).

3. The principle of optimality (which helps solve the SPP) cannot be applied here. The main reason is that the *expected* length of a path is not an additive functional (as opposed to the length of a path), in the sense that in general $E[L_{c_1 \oplus c_2}] \neq E[L_{c_1}] + E[L_{c_2}]$ where $c_1 \oplus c_2 = (i_1, \ldots, i_k)$ stands for the concatenation of the two paths $c_1 = (i_1, \ldots, i_j)$ and $c_2 = (i_j, \ldots, i_k)$.

Based on Result 1, a polynomial time algorithm for the PSPP seems out of reach; from Result 2, the optimal SPP path cannot be considered as a good candidate for approximating the corresponding PSPP; and from Result 3, one has to be careful about utilizing classical SPP algorithms (see for example Papadimitriou & Steiglitz [1982] for the PSPP).

A practical consequence of these results is the necessity to develop entirely new solution procedures. As mentioned earlier, a branch and bound scheme was proposed in Jaillet [1988]. In Jaillet [1992b], it has been shown that, if the number of probabilistic ('white') nodes is, at any time, bounded by a constant, the PSPP can still be solved in polynomial time. Since this is the first instance in this chapter in which a large subclass of problems are solvable by polynomial time procedures, let us briefly review the detailed arguments in the next section.

2.5.4. Polynomial procedures for special cases of the problem

We now consider some special cases of the PSPP for which we are able to give polynomial time procedures. Let us first consider the simplest of these cases and show how this can be done. The other cases will be straightforward extensions of the main idea developed here.

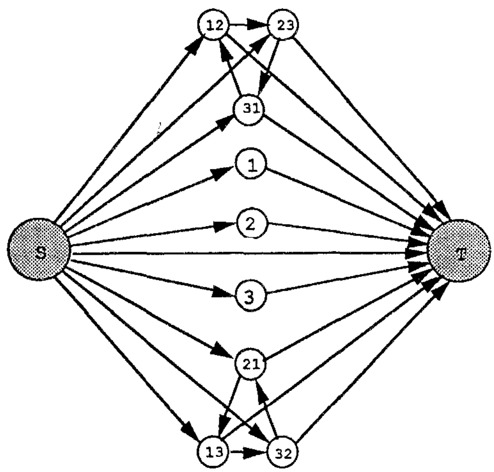


Fig. 5. The auxiliary network when $N_2 = \{1, 2, 3\}$.

A simple special case. Let us assume that $m = 2$, i.e., the only black nodes are s and t. Let us also consider a probability $\mathbb{P}$ such that

$$\Pr(W \leq n-2) = 0 \text{ and } \Pr(W = n-1) > 0, \tag{50}$$

i.e., either all white nodes are working or only one of them has a failure.

Let us then construct an auxiliary network (V, E, φ) as follows (see Figure 5 for an illustration): The set of nodes is $V = (N_2 \otimes N_2) \cup N_1 \cup N_2$ of cardinality $|V| = n^2 + 2$ (where the notation $A \otimes A$ stands for $(A \times A) \setminus diag(A)$); the set of arcs E and the arc-length function φ are defined by:

- Arc (s, t) of length $\varphi(s, t) = d(s, t)$.
- Arcs (s, v) for all $v = i$ in N_2 of length $\varphi(s, i) = \beta_0 d(s, i) + (\gamma_1/2) d(s, t)$.
- Arcs (s, v) for all $v = (i, j)$ in $N_2 \otimes N_2$ of length $\varphi(s, (i, j)) = \beta_0 d(s, i) + (\alpha_0/2) d(i, j) + \beta_1 d(s, j)$.
- Arcs (v, t) for all $v = i$ in N_2 of length $\varphi(i, t) = \beta_0 d(i, t) + (\gamma_1/2) d(s, t)$.
- Arcs (v, t) for all $v = (i, j)$ in $N_2 \otimes N_2$ of length $\varphi((i, j), t) = \beta_0 d(j, t) + (\alpha_0/2) d(i, j) + \beta_1 d(i, t)$.
- Arcs (v, w) for all $v = (i, j)$, $w = (j, k)$, $i \neq j \neq k$ of length $\varphi((i, j), (j, k)) = (\alpha_0/2)(d(i, j) + d(j, k)) + \alpha_1 d(i, k)$.

The cardinality of E is then $|E| = n(n-1)(n-2) + 1 + 2(n(n-1) + n) = n^3 - n^2 + 2n + 1$.

One can now give the fundamental relationship between the two networks:

Lemma 3. *There is a one to one correspondence between the set of paths from s to t in (N, A, d) and the set of paths from s to t in (V, E, φ). Moreover the expected length of each path from s to t in (N, A, d) is equal to the length of a corresponding path in (V, E, φ).*

We are now in a position to give our main result in this simple situation:

Theorem 18. *Given a node-invariant PSPP with $m = 2$ and with a probability $\mathbb{P}$ such that* $\Pr(W \leq n-2) = 0$ *and* $\Pr(W = n-1) > 0$*, one can find an optimal PSPP path between s and t in time $O(n^3)$.*

Theorem 18 follows directly from Lemma 3. Indeed, from Lemma 3, one can solve such a PSPP by finding the shortest path in the auxiliary network (V, E, φ). This can be done by a careful implementation of the Dijkstra algorithm in time $O(|E| + |V| \log |V|)$ [see Driscoll, Gabow, Shraiman & Tarjan, 1988, or Fredman & Tarjan, 1987].

Generalizations

We can extend the idea of Theorem 18 in several directions.

We can first consider cases in which several white nodes can fail at the same time. For example let us look at the case in which $\Pr(W \leq n-3) = 0$. The set of nodes V will be augmented to include all triplets (i, j, k) with $i \neq j \neq k$, i.e. of $n(n-1)(n-2)$ nodes. The set of arcs E and the function φ will remain the same except that we add arcs of the types $(s, (i, j, k))$, $((i, j, k), t)$, and $((i, j, k), (j, k, l))$ (with $l \neq i$) and that we delete arcs of the type $((i, j), (j, k))$. Now the PSPP can be solved in $O(n^4)$.

Another example of a generalization is to consider more than two black nodes: the set V would be augmented to include $m-2$ additional nodes, and we would have to add arcs of the types (i, v),(v, i),$((i, j), v)$, and $(v, (i, j))$ for all v in $N_1 \setminus \{s, t\}$ and i, j, $i \neq j$ in N_2, and arcs of the types (s, v), (v, t), (v, w), and (w, v) for all $v \neq w$ in $N_1 \setminus \{s, t\}$.

The most general result is obtained by considering a combination of the two previous extensions:

Theorem 19. *Given a node-invariant PSPP with m black nodes ($m \geq 2$) and a probability $\mathbb{P}$ such that* $\Pr(W \leq n-k-1) = 0$ *and* $\Pr(W = n-k) > 0$*, one can find an optimal PSPP path between s and t in time $O(mn^{k+1} + n^{k+2} + m^2)$.*

We note that similar techniques can be applied to special versions of other problems such as the PTSP (i.e., for some special cases, a PTSP can be transformed into a TSP). Finally these results can be used to obtain good heuristics to the PSPP or PTSP when the probability distribution of W — the total number of nodes that are present, is such that $\Pr(W \leq n-k) = \epsilon$ with ϵ very small (for example for a Bernoulli case with p very close to 1). One can also analyze exactly the quality of these heuristics using the general framework given in Jaillet [1985, pp. 163–169].

2.5.5. Applications

In Section 2.1. we have seen one application of the PSPP. We now summarize a few other applications. In the context of flying operations, nodes s and t can be airports and the other nodes can represent geographical areas (mountains, countries, etc.) that can or cannot be flown over by aircraft going from s to t (for example, because of weather conditions, unexpected military restrictions, etc.). The modifications of the route the plane has to take, because of such unexpected restrictions, can be costly, if not anticipated beforehand. One would prefer to consider explicitly these uncertainties in the model in order to find routes of minimum expected costs. The PSPP model is appropriate for such a situation.

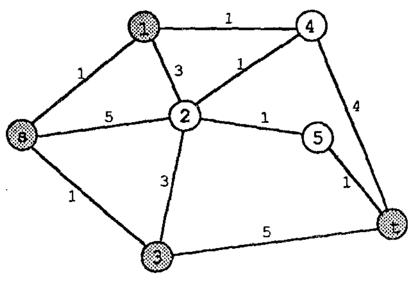
More important, consider the following class of problems: suppose one has to go from a 'city' s to another city t on a probabilistic network, possibly by passing through other cities in which one receives some 'revenue'. With the objective of minimizing the expected total net cost, this problem can be modeled as a PSPP: The underlying network is a complete graph built on all cities of interest, and the length of an arc (i, j) is the net cost of traveling between city i and city j (minimum cost of traveling from i to j minus half the total revenue in the cities, $c_{ij} - (r_i + r_j)/2$).

In the context of network reliability, the PSPP can be interpreted as providing a local (hence, easily implementable) strategy of handling node failures (by skipping them) which is optimal in a global sense (minimization of the expected cost). This can be useful either for describing operating strategies on unreliable networks, or for having a polynomial-time computable estimate of 'connecting cost versus node failures' for such networks.

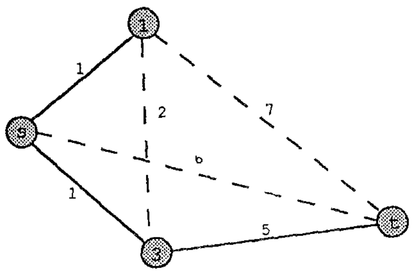
Finally, it is worth mentioning that if, for a specific application, the physical underlying network is not complete, the PSPP model can still be used by first transforming the network into a complete one. In that case a crucial condition for making the network complete with respect to the PSPP is the following: Given the selected a priori path $(s, n_1, n_2, \ldots, n_k, t)$, if nodes n_{i+1} through n_{i+r} fail, then they are skipped by either taking the arc (n_i, n_{i+r+1}) if it exists in the original network, or by taking an alternative path from n_i to n_{i+r+1} *using only nodes whose probability of failure is equal to zero*. This latter condition is to insure that we do not define a 'random' length for this missing edge, i.e., an edge whose length would depend on the presence or not of 'probabilistic' nodes. We illustrate this procedure in Figure 6 for the case of a simple symmetric network with three 'probabilistic' nodes (2,4, and 5), and four 'deterministic' nodes (s, t, 1, and 3). In order to insure that our procedure respects the previous crucial condition, we split it into three main steps: In Step 1, we consider missing arcs between 'deterministic' nodes; in Step 2, we consider in turn each 'probabilistic' node, and missing arcs between it and 'deterministic' nodes; and in Step 3, we consider in turn all missing arcs between pairs of 'probabilistic' nodes.

Transforming a network into a complete one

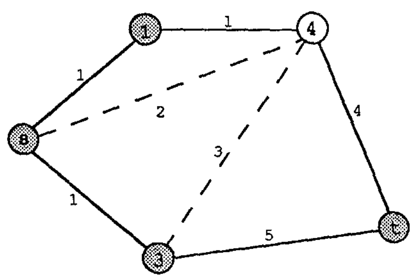
1. First, delete all 'probabilistic' nodes and their adjacent arcs, and call the remaining network the *backbone* B. Make B complete by using shortest paths. In our example, edges (s, t), $(1, 3)$, and $(1, t)$ are thus added. Note that if B is



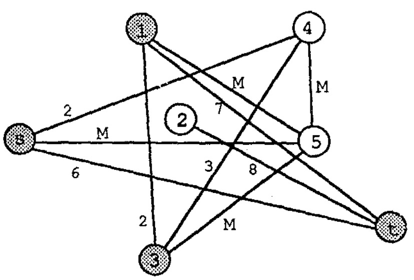
(i) The original graph. Nodes 2, 4, and 5 may fail with a probability p.



(ii) The backbone B with its initial arcs (plain lines). B is then made complete by adding arcs (dashed lines).



(iii) The complete backbone B with node 4 and its initial arcs adjacent to B (plain lines). The arcs $(s, 4)$ and $(3, 4)$ are then added (dashed lines).



(iv) The final set of all added weighted arcs.

Fig. 6. An original network and the addition of missing weighted arcs.

disconnected, every arc added between disconnected components should then be given a large weight, say M (this case is not shown in our example).

2. Then, consider a 'probabilistic' node, say i, and add it (together with adjacent arcs) to B. From i, add all necessary arcs with weight equal to that of the shortest paths that do not use t as an intermediate node. Then, remove i from B and repeat the procedure for the other 'probabilistic' nodes. In our example, the edges $(2, t)$, then $(s, 4)$, $(3, 4)$, and finally $(s, 5)$, $(1, 5)$, $(3, 5)$ are thus added. Again, if the 'probabilistic' node is not initially directly connected to B by a node other than t, each added arc will then be given the large value M (in our example this is the case for node 5).

3. Finally, consider two non-adjacent 'probabilistic' nodes, say i and j, and add them (with adjacent arcs) to B. Add arc (i, j) with weight equal to the shortest path between i and j that does not use t as an intermediate node. Then remove the pair i, j, and repeat the procedure for the other pairs of non-adjacent 'probabilistic' nodes (in our example, edge $(4, 5)$ is thus added).

Note that this entire procedure might require (see above) the choice of a single large value M.

2.6. Practical approximations to optimal a priori solutions

In this section we briefly discuss computational experience in trying to find useful heuristic solutions to SRPs using the a priori optimization approach. We use the Euclidean PTSP as an example.

In numerical experiments, Bertsimas [1988] has obtained solutions to Euclidean PTSPs by means of two different types of heuristics. The first of them is the spacefilling curve heuristic, while the second is based on seeking local optimality. The implementation of the spacefilling curve heuristic uses heapsort for the sorting part of the procedure, and thus requires only $O(n \log n)$ time to find a nearly optimal tour SF. Interestingly, this is even faster than the computation of the expected length of that tour, $E[L_{\mathrm{SF}}]$, which requires $O(n^2)$ time. Since the computed tour SF is independent of the probabilities p_i, the spacefilling curve heuristic can be used when these probabilities are not all the same, or even when they are not accurately known.

For problems involving equal probabilities $p_i = p$, and not more than a few hundred nodes, considerable success was achieved with two separate iterative improvement algorithms based on the idea of local optimality. Given a tour τ and a set $S(\tau)$ of tours which are minor modifications of τ, the tour τ is said to be locally optimal if

$$E[L_\tau] \leq \min_{\tau' \in S(\tau)} E[L_{\tau'}]. \tag{51}$$

The iterative improvement algorithm works by choosing an initial tour τ_0, then testing to see if τ_0 is locally optimal. If a better tour τ_1 is found, it then replaces τ_0 and is itself tested. Since there are only a finite number of possible tours, this procedure must eventually converge to a locally optimal tour τ_* — which, it is hoped, will be a nearly-optimal solution to the problem.

Lin [1965] used an iterative improvement algorithm for the TSP based on what he called the λ-opt local neighborhood. For a given tour τ consisting of n links between nodes, the neighborhood $S_\lambda(\tau)$ consists of those tours which differ from τ by no more than λ links. For $\lambda = 2$ this is the set of tours which can be obtained by reversing a section of τ; for $\lambda = 3$ it is the set of tours obtainable by removing a section of τ and inserting it, with or without a reversal, at another place in the tour. Both the 2-opt and 3-opt TSP algorithms were implemented, since when p is greater than about 0.5 the TSP solutions provide useful starting points for our more general PTSP routines.

Unlike the TSP case, the expected length $E[L_\tau]$ in the PTSP sense depends on all $(n^2 - n)/2$ independent elements of the distance matrix. We cannot, therefore, speak of some links leaving and others entering the tour; rather, it is only the weight given to each of the $d(i, j)$ by equation (6) which changes. We can still use Lin's λ-opt neighborhoods,but the computation of the changes in expected length becomes considerably more complicated. It takes $O(n^2)$ time to calculate the change in expected length from τ to an arbitrary tour in $S_2(\tau)$, so it would seem at first that testing for even 2-p-optimality (referred to heretofore as '2-p-opt') would take $O(n^4)$ time. We can, however, reduce this to $O(n^2)$ if we examine the tours in the proper sequence and maintain certain auxiliary arrays of information as the computation proceeds.

Another neighborhood tried by Bertsimas [1988] consists of moving a single node to another point in the tour, rather than reversing an entire section. The corresponding neighborhood, which we call the 1-shift neighborhood, has roughly twice as many members as S_2, it is a subset of S_4, and yields much better results than S_2 in our experiments.

The best general approach seems to be to first use the spacefilling curve algorithm, followed by 3-opt if p is fairly large, and then finish by applying 1-shift. The threshold point below which 3-opt ceases to be helpful is uncertain and probably depends strongly on the specifics of the problem. For a detailed description of the numerical results the reader is referred to Bertsimas 1988].

3. Modelling issues for dynamic problems

Dynamic problems pose a rich set of modeling issues which must be resolved as a model is developed. In this section, we review these issues to provide a foundation for the technical presentations on model formulation and solution. First, Section 3.1 reviews some general modeling issues that must be addressed in the formulation of a model, and discusses some important choices that have to be made in the process of formulating a dynamic model. Next, Section 3.2 provides a general taxonomy of dynamic models covering different types of models that arise in different applications. Section 3.3 discusses two key issues that arise in the formulation of dynamic models, and Section 3.4 lists alternative objective functions that can be used for dynamic optimization problems, including a review of some basic concepts and terminology from the planning horizon literature. Finally, Section 3.5 summarizes major solution approaches that have been used.

3.1. General issues

In contrast with static models, dynamic models offer a rich and difficult set of issues that must be resolved before we even understand the basic framework in which we are working. Key decisions that must be made in the choice of the model include:

Deterministic vs. stochastic — Deterministic models are fundamentally different than stochastic models, and pose different algorithmic challenges, as well as different answers to questions such as the choice of the proper planning horizon.

Myopic vs. dynamic — Most operational models are basically myopic, optimizing based on what is known in the near future. While the problem is dynamic, the models are basically static, and forecasted future activities are incorporated in at best a very heuristic way. A dynamic model actually attempts to capture future activities.

Choice of objective function — Infinite horizon problems must be approximated to handle forecasting uncertainties and ensure boundedness. A standard choice uses discounted or undiscounted costs over a finite horizon.

The planning horizon — Dynamic models are typically solved over a finite planning horizon. The modeler must choose an appropriate length for the planning horizon, which will in turn effect the type of forecasting that is required as well as the basic formulation of the model.

Spatial and temporal aggregation — The preponderance of models proposed to date use a discrete space, discrete time formulation. This requires choices about the level of temporal and, in some cases, spatial aggregation.

There are six broad research issues that arise in the design and solution of large-scale dynamic network models.

1. *Developing accurate models for optimization under uncertainty.* There are a variety of techniques for optimizing problems under uncertainty. Applied in a standard manner, these techniques are often hopelessly intractable for problems of realistic size. Approximations need to be proposed and tested on the basis of accuracy and computational efficiency. There are a variety of approaches for developing approximations, and a considerable amount of research is needed to determine the best approach.

2. *Identifying 'efficient' formulations.* There is often more than one formulation of the same problem, and some formulations lead more naturally to efficient solution algorithms. Using linear, nonlinear and stochastic networks, we show how alternative formulations can significantly impact algorithmic efficiency.

3. *Design of efficient solution algorithms.* Large stochastic, dynamic network models are intractably difficult. While significant progress may be made in the formulation of the model, continued progress in the development of solution algorithms specifically designed for dynamic networks will likely prove extremely valuable. We review a number of algorithms that have been specially developed for linear, nonlinear and stochastic, dynamic networks.

4. *Planning horizons and truncation errors.* The replacement of infinite horizon models with finite approximations inevitably produces errors. We need to evaluate truncation errors and develop methods for reducing these errors without significantly increasing the size of the model.

5. *Errors due to spatial and temporal aggregation.* It is often necessary to discretize both time and space. Relatively little research has addressed the errors produced by aggregation.

6. *Evaluating a stochastic, dynamic model.* Determining the quality of the solution from a stochastic, dynamic model is a difficult question. Tight bounds are not available, and exact solutions are not possible for problems of realistic size. Extensive rolling horizon simulations are needed to compare two models empirically, raising the difficult problem of experimental design and evaluation. We need better bounds to evaluate the quality of the solutions we obtain, and better experimental methodologies for testing and comparing approximations.

Stochastic formulations of dynamic models are proposed in Sections 7 (in particular, Section 8.2) and 9. Most of the work in stochastic models has focused on the dynamic vehicle allocation problem, but there has been some work on stochastic vehicle routing [Dror, Laporte & Trudeau, 1989; Stewart & Golden, 1983]. The investigation of alternative formulations is reviewed in Sections 5.4

and 5.5. The development of efficient solution algorithms is covered in depth in Section 5 for deterministic networks and Sections 7 and 9. Issues regarding planning horizons are covered in Sections 3.4.3, 6.2.3 and, from a different perspective, 10.4. Finally, the problem of evaluating stochastic, dynamic models (in a rolling horizon framework) is covered broadly in Section 10.

3.2. A taxonomy of dynamic models

Dynamic models come in several forms depending on the nature of the application. These include:

1. *Fixed (finite) horizon* — these models have a natural, finite horizon that most typically occurs because of the bounds of a work shift. In other cases, management might be specifically designing a weekly schedule, producing a fixed horizon.
2. *Rolling (infinite) horizon* — infinite horizon models arise when there is a continuum of decisions being made which impact the future. Examples include dynamic traffic assignment and dynamic fleet management. These problems generally require some form of model truncation.
3. *Periodic* — periodic models arise in a planning context when there is a natural period to the data, such as time of day and/or day of week. These models often arise in the context of inventory routing problems, and involve the development of routine schedules that can be run on a day to day basis.
4. *Dynamic equilibrium* — dynamic equilibrium problems arise in a (long term) planning context for fleet management problems, where a dynamic network model is being used to capture flows over time, but where the model must be solved in such a way that the inventories of vehicles in each city in the first time period are the same as those in the last time period. A typical application of this model would be in fleet sizing, where we wish to simulate the flows of vehicles moving loaded and empty over time. However, we want our model to capture a typical week or month, and we do not want the model to start out with all the fleet capacity in one city and end up with it in another city.

3.3. Formulation issues

Two issues of special importance in the formulation of dynamic models are a) the choice of stochastic versus deterministic formulations, and b) the use of a simultaneous versus recursive model structure. The first issue addresses the basic representation of the problem itself, while the second is a mathematical choice of two problem formulations.

3.3.1. Stochastic versus dynamic models

Uncertainty arises in dynamic models from five exogenous sources:

1. Uncertainty in demand forecasts.
2. Uncertainty in forecasts of external supplies of equipment and drivers.
3. Randomness in the performance of the network (e.g. weather).

4. Randomness in the management and operation of the network in future time periods.

5. Errors in the data provided to the model.

Errors in demand forecasts can often be quantified and modeled in a formal way. Demand forecasts are generally derived from time series models, and past patterns of errors can provide a measure of future errors. Uncertainty in forecasts of external supplies of equipment or drivers arises most often as a result of uncertainty in travel times for movements that originated at an earlier time. Randomness in the performance of the network might refer to uncertainty in travel times (due to weather or congestion) or network capacity. One effect of randomness in network performance is randomness in downstream vehicle supplies. For example, a container sent to a shipper may be held for a random number of days before being released back to the carrier. The fourth source of error is much more difficult to quantify. Planning at some time t requires anticipating how the system will be managed or operated in later time periods. These anticipated actions will not always be implemented as planned, producing a different source of error. Finally, all models must acknowledge the presence of errors in the input data. Such errors are difficult to model and should, as a rule, be viewed as a management problem. However, we cannot always eliminate them, and our models must acknowledge that these errors exist.

Deterministic models are most widely used in practice due to their ease of formulation and the availability of existing solution algorithms. There are, however, several reasons to support the use of stochastic models, including:

1. Deterministic models do not (in general) exhibit a natural planning horizon, and truncation errors can in practice be significant. The result is that deterministic models are often very large, and may be difficult to solve, preventing their use in an operational setting.

2. Deterministic models can exhibit 'nervousness' as a result of sensitivity to changes in forecasted information. Recommendations can change unnecessarily with modifications to forecasted information, increasing costs and reducing confidence in the model.

3. A deterministic model is an approximation, and can produce inferior results, sometimes significantly so, compared to models that handle uncertainty explicitly.

Depending on the choice of formulation, incorporating uncertainty can make a model either hopelessly intractable, or compact and easy to solve.

3.3.2. Simultaneous vs. recursive formulations

A key theme in development of dynamic network models is the role of model formulation. Within the chapter, the presentation of different solution approaches has been organized more along the lines of deterministic and stochastic models and formulations. However, it is often the case that we can choose among multiple formulations for the same model. In Section 5.4 we show how a nonlinear, dynamic network can be solved much more efficiently by using a special flow splitting formulation. In Section 5.5 we show how dynamic multicommodity networks can be solved using decomposition by using formulations that mitigate the effects of degeneracy.

In this section, we review a separate issue. Consider a generic (linear) dynamic network flow problem with the general form:

$$\min_{x_t} \sum_{t=0}^{P} c^T x_t \tag{52}$$

subject to

$$A_t x_t - \sum_{t'=0}^{t-1} B_{t',t} x_{t'} = R_t \quad t = 1, \ldots, P \tag{53}$$

where $B_{t',t}$ is the node-arc incidence matrix giving the elements of $x_{t'}$ that send flow from nodes in period t' to period t, and A_t gives the elements of x_t that send flow from period t to periods $t' \geq t$. We may often simplify the problem by assuming that $B_{t',t} = 0$ for $t' < t - 1$.

We refer to this form of the model as the *simultaneous* formulation, since we consider the optimization of flows in all time periods simultaneously. Alternatively, we can formulate the problem using a state variable. Let S_t be the flow entering each node in period t from periods $t' < t$, defined by:

$$S_t = \sum_{t'=0}^{t-1} B_{t',t} x_{t'} \tag{54}$$

Now rewrite (52) as follows:

$$\min_{x_0, S_1} c^T x_0 + Q_1(S_1) \tag{55}$$

where the function $Q_1(S_1)$ is defined recursively using:

$$Q_t(S_t) = \min_{x_t} c^T x_t + Q_{t+1}(S_{t+1}) \tag{56}$$

subject to

$$A_t x_t = R_t + S_t \tag{57}$$

$$S_{t+1} - B_{t,t+1} x_t = S_t \tag{58}$$

We refer to the formulation (55)–(58) as the *recursive* formulation. The simultaneous formulation is most commonly associated with deterministic models, while the recursive form is almost required by any stochastic model. In stochastic models, $Q_t(S_t)$ plays the role of the expected recourse function or value function. However, in Section 5.4 we see that the recursive formulation is of tremendous value for solving deterministic nonlinear, dynamic networks. At the same time, the most widely used approach for solving stochastic optimization problems, called scenario aggregation (see Section 7.4), uses the simultaneous formulation. The recursive form seems better suited to dynamic models, but at this time both approaches are being used in the research literature.

3.4. Objective functions

The travelling salesman problem is often cited as the archetypal example of a 'hard' combinatorial optimization problem. However, in one important respect, this problem is significantly easier than the dynamic models presented in this chapter — the objective function is well defined. While the determination of the optimal solution may be difficult, the comparison of two heuristics is generally straightforward. When solving a dynamic model under uncertainty, the choice of the objective function is not obvious. This issue has been studied extensively in the research literature, primarily in the context of inventory planning and dynamic lot sizing.

We briefly discuss three issues in the formulation of the objective function. Section 3.4.1 presents alternative criteria that can be used in dynamic models. Section 3.4.2 reviews methods for formulating bounded objective functions for infinite horizon problems. Finally, Section 3.4.3 provides a summary of basic terminology from the planning horizon literature for solving finite approximations of infinite horizon problems.

3.4.1. Cost criteria for dynamic models

In static models, minimizing cost, subject to constraints of servicing all the demand, is the standard criterion for evaluating the quality of a solution. In dynamic models, the choice of measurement is not as obvious. Depending on the application, we might consider:

1. *Costs*. Of course, minimizing costs (over a finite or infinite horizon) remains a standard criterion for many applications.

2. *Profits*. Unlike static models where we usually service all the demand, it is invariably the case that dynamic models must consider the possibility of not servicing all the demand. In a real-time setting, it is simply not always possible to guarantee that you will service all the demand all the time. The possibility of refusing service arises in some applications, implying that operating profits (revenue minus cost) is more relevant. In other settings, all demands are handled, but with varying degrees in the quality of the service provided, suggesting that penalties be introduced to capture the effect of poor service.

3. *Minimum covering*. If we want the fewest number of vehicles or crews required to handle a set of demands, then we want a minimum flow solution subject to covering constraints [Dantzig & Fulkerson, 1954].

4. *Maximum throughput*. We may wish to design a network that maximizes the total flow through the network, thereby maximizing the total demand served or service rendered [Orlin, 1983].

5. *Average time in network*. Traffic assignment problems are often primarily interested in the time required to push flow through the network. We may minimize average time in the network, which is fairly easy, or we may wish to minimize a nonlinear function of travel time, which is generally much harder.

6. *Maximum clearance time*. Evacuation problems typically focus on the latest time at which flow remains in the network. The minimum time (or 'bottleneck

time') transportation problem seeks to minimize the latest time that all demands are satisfied [see Szwarc, 1966, 1971a; Hammer, 1969, 1971].

3.4.2. Objective functions for infinite horizon optimization

Let x_t be the set of actions in period t, c_t be the cost vector and S_t be the state of the system given $\{S_{t-1}, x_{t-1}\}$. Let $\mathcal{X}_t(S_t)$ be a set of actions that we may take in period t, let $g_t(S_t, x_t)$ be the costs incurred in period t, and let G_t be the total optimal costs over periods $\{0, \ldots, t\}$. Three different optimality criteria have been suggested in the literature to produce bounded objective functions over infinite horizons:

Infinite discounted programming [Derman, 1970]. For a given discount factor $\{0 \leq \alpha < 1\}$ we solve:

$$G^* = \inf_{x \in \mathcal{X}} \sum_{t=0}^{\infty} \alpha^t g_t(S_t, x_t)$$

If g_t is bounded, then $\alpha < 1$ guarantees that G^* exists.

Average reward criterion [Derman, 1970]. For stationary problems with homogeneous costs and demands, it may make sense to minimize the average costs per period:

$$G^* = \inf_{x \in \mathcal{X}} \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T} g_t(S_t, x_t)$$

Relative gain optimization [Howard, 1971]. The first two formulations represent methods for avoiding unbounded objective functions. An alternative approach that can be used on the context of discrete dynamic programs minimizes the expected relative cost, where the cost of each state is measured relative to a base state which is defined as zero. Let π_{it} be the probability of being in state i at time t (given initial state S_0) and let $\mathcal{S}_t$ be the set of accessible states at time t given S_0. For a given state $i \in \mathcal{S}_t$, the expected reward in period t is given by:

$$v_{ti} = \min_{x_t \in \mathcal{X}_t} C_t(x_t, i) \tag{59}$$

where $C_t(x_t, i)$ is the cost of being in state i and taking action x_t. The total expected reward in period t is then:

$$V_t = \sum_{i \in \mathcal{S}_t} \pi_{it} v_{it} \tag{60}$$

If the underlying problem is a homogeneous Markov reward process, then it is well known that [Bhat, 1984; Heyman & Sobel, 1984]:

$$\sum_{t=0}^{T} v_{it} = v_{i0} + gT \quad \text{as} \quad T \to \infty \tag{61}$$

where v_{i0} is a state dependent constant and g is a system dependent constant. If we let state 1 be a base state, then we can define new values:

$$\bar{v}_{it} = v_{it} - v_{1t} \tag{62}$$

In the limit:

$$\lim_{T\to\infty} \sum_{t=0}^{T} \bar{v}_{it} = v_{i0} - v_{01} \tag{63}$$

which is bounded. Thus, by using relative gains, we may solve an infinite *undiscounted* optimization problem.

Remarks. The average reward criterion works only for purely stationary models, since, in the limit, no weight is put on activities in the near future. This criterion was used in Orlin [1983], [1984b] in the study of stationary dynamic networks. The relative gains model is intuitively appealing since it captures the actual tradeoff between competing activities and produces a bounded objective function without requiring the use of artificial discounting. This method was the basis of a heuristic presented in Powell [1987] for the dynamic fleet management problem for truckload motor carriers. A Markov reward model was used to develop relative salvage values to capture the value of additional vehicles in one region over another.

The discounted infinite horizon formulation is the most standard basic model. The textbook motivation for the discount factor is accounting for the time value of money. In most operational models, however, the effective planning horizon is too short for this effect to have any impact. Instead, discount factors are used to heuristically account for uncertainty in the data. For example, if the time period is one week, we might use a discount factor $\alpha = 0.3$, equivalent to a *weekly* interest rate of 70%! This approach is simple and intuitively appealing, but is a clearly heuristic approach to handling forecasting uncertainties. For example, this method discounts all activities in period t by α^t, whereas in practice there may be very different degrees of uncertainty associated with different activities in the same time period. For example, containers dispatched from Chicago to Tokyo will arrive in Tokyo with a very high probability in about two weeks, and yet the demands for containers in Tokyo two weeks from now may be quite uncertain.

3.4.3. Planning horizons

The most common approach to approximating infinite horizon problems is to solve a finite, or truncated, problem. Let $P < \infty$ be a specified *planning horizon*. Then, for any α, $\{0 \le \alpha \le 1\}$ we solve:

$$G^*(P) = \inf_{x\in\mathcal{X}} \sum_{t=0}^{P} \alpha^t g_t(S_t, x_t)$$

This method, which may be used with discounted or undiscounted costs, is the most standard approach used in practice. Although a discount factor is not required to ensure boundedness, it is often used in practice as a heuristic approach for accounting for uncertainty. Planning horizon methods possess the intuitive appeal of working in a manner similar to the way people work [Morton, 1981]. The idea is that, for sufficiently large P, the quality of the solution in the first period, x_0, will be sufficiently good for practical purposes.

This raises the problem of choosing an appropriate planning horizon, and determining the properties of the solution provided by a finite approximation. In this section, we briefly review some of the basic concepts and terminology that has been developed within the planning horizon literature. First, the term *planning horizon* is used fairly loosely to describe how far into the future the model extends in order to make decisions now. The term is not well defined, because there are different ways that forecasted data can be used in a dynamic model. For example, *salvage values* [Grinold & Hopkins, 1973] can be used to capture the value of extra flows into a node at the end of the planning horizon. These salvage values approximate activities beyond the formal end of the planning horizon.

Excellent discussions of the planning horizon literature can be found in Bean & Smith [1984], Sethi & Bhaskaran [1985], Bhaskaran & Sethi [1987], Bes & Sethi [1988] and Morton [1979]. In this literature, general concepts of planning horizons have been replaced with more precise notions of *forecast horizons* and *decision horizons*. A forecast horizon (or exact forecast horizon) τ^f is a period such that planning horizons $P \geq \tau^f$ produce the same solutions for periods $\{1, \ldots, \tau^d\}$, where $\tau^d \leq \tau^f$ is called a decision horizon. τ^f is called a *near forecast horizon* if the first period solution 'closely approximates' the infinite horizon optimal solution. A *weak forecast horizon* exists when conditions on future data must be imposed to produce a forecast horizon.

There is an extensive literature on planning horizons, most of it motivated by dynamic lot size and inventory problems. One of the earliest results on forecast horizons is given by Wagner & Whitin [1958] for the dynamic lot size problem, which can be formulated as a dynamic network with concave costs. Wagner–Whitin show how a forecast horizon can accelerate the solution algorithm, as well as identify how far into the future forecasted data can impact first period decisions. Comprehensive surveys of forecast/decision horizons are given by Bhaskaran & Sethi [1987], Morton [1979] and Bensoussan, Crouhy & Proth [1983]. Bes & Sethi [1988] provide a formal treatment of forecast/decision horizons for stochastic, dynamic problems.

Research in this area tends to fall along two general lines. The first depends on the identification of *regeneration points* which produce forecast horizons by creating points in time when the system effectively restarts itself. Morton [1979] provides a general framework for the use of regeneration points in undiscounted, infinite horizon problems. The use of regeneration points has proved particularly effective in the solution of deterministic problems with concave costs [Zangwill, 1969; Lundin & Morton, 1975] which often arise in lot size models and production planning. The second line of investigation has looked at convex, stochastic problems which tend to exhibit a convergence of state vectors in future periods as a result of the ergodic structure of the underlying stochastic process [see, for example, Morton & Wecker, 1977]. A closely related line of investigation looks at deterministic convex problems with discounted costs [Bean & Smith, 1984].

In contrast with the extensive research literature on inventory and production problems, planning horizon results for dynamic networks (with the notable exception of concave cost networks) are much more limited. This is even more true of the types of dynamic networks that arise in dynamic fleet management. Aronson & Chen [1986] present a specialization of network simplex for dynamic networks, drawing on related work for staircase linear programs [Aronson & Thompson, 1984; Aronson & Chen, 1985]. This algorithm is oriented toward dynamic production planning problems which consist of a static transportation problem in each period, where inventory arcs that carry excess supply from one period to the next. Aronson & Chen note that the simplex basis tends to exhibit breaks which create regeneration points which can be used to identify decision horizons (see Section 5.2). However, a break in the basis for a P-period problem is not guaranteed to remain as the planning horizon is increased. For this reason, Aronson & Thompson [1984] introduce the notion of an *empirical* (or *computational*) decision horizon, which is a decision horizon that is likely to hold with a high probability. Aronson & Chen [1989] summarize empirical studies of decision horizons in dynamic networks for production planning applications.

3.5. Solution approaches

We divide solution strategies into seven broad areas, five of which deal explicitly with stochastic problems. These are:

1. *Deterministic model/linear programming.* We assume that all deterministic formulations can be formulated in some way as a linear program. Of course, some of these linear programs are exceptionally large, and specialized algorithms have been devised to handle both the side constraints [Lasdon & Terjung, 1971; Assad, 1987; Kennington & Helgason, 1980; Farvolden, Powell & Lustig, 1993] and the dynamic structure [Ho & Manne, 1974; Aronson & Chen, 1986].

2. *Chance constrained programming* [Charnes & Cooper, 1959]. For the case of random demands, it may be useful to specify the fraction of demand that should be satisfied, representing a level of service constraint (see Section 9.1.3).

3. *Stochastic programming/scenario aggregation.* Stochastic programs can be approximated by optimizing decisions over a specified set of *scenarios*. The solution for each scenario must be aggregated to produce a single recommended action in the first time period (see Section 7.4).

4. *Stochastic gradient methods.* These methods approximate expected future costs by taking a sample of future activities and then calculating a sample gradient, which can then be used to optimize decisions in the first time period. Several techniques are available for using sample information within an optimization procedure (see Section 7.5).

5. *Approximations of stochastic programs.* This class of methods refers to techniques that replace the original stochastic program with an approximation that

simplifies the problem in such a way that it can be solved with classical techniques. These methods are heuristic and must be evaluated experimentally. The simplest approximation that is widely used in practice is stochastic programming with simple recourse, but other methods can be used (see Section 9).

6. *Markov decision processes.* While MDP's have not been successfully applied to the solution of large scale, stochastic networks, there is an extremely rich theoretical literature on MDP's with considerably more attention given to the study of the properties of optimal solutions, the errors due to model truncation and the convergence of optimal policies.

7. *Optimal control.* Optimal control theory is the only approach that explicitly considers problems in continuous time. Research in this class of problems is limited.

4. Dynamic models in transportation and logistics

We consider the following sequence of models:

- dynamic shortest paths;
- dynamic traffic assignment (system optimization);
- dynamic production and inventory planning;
- dynamic vehicle allocation;
- dynamic assignment;
- dynamic vehicle routing;
- dynamic service network design;
- dynamic facility planning.

The first two problems address the task of routing flows over a network. The first considers a single vehicle or traveler, while the dynamic traffic assignment problem considers many vehicles moving between multiple origins and destinations.

Next we introduce the classical dynamic production and inventory planning problem. Strictly speaking, this problem area is outside of the scope of this chapter [see Shapiro, 1993]. However, there is a substantial literature on dynamic networks that is motivated by these applications, and many of the concepts regarding dynamic models can be traced to results derived in the context of production and inventory planning.

The next three areas deal with routing of vehicles, in order of difficulty. The dynamic vehicle allocation problem is typically approached as a (possibly pure) dynamic network. Vehicles are modeled as aggregate flows over a network, with very little ability to enforce driver work rules. The dynamic assignment problem considers the assignment of drivers to individual tasks (e.g. moving loads from one point to another) on a rolling basis. Finally, the dynamic vehicle routing problem addresses the problem of clustering customers into loads.

The last two problems are the hardest from a combinatoric perspective. We review the limited literature on these problems, but do not cover them in any depth in this chapter.

The presentation begins in Section 4.1 with a short discussion of basic conventions in notation and vocabulary. The rest of the section presents a sample of specific models, in order of increasing complexity.

4.1. Nomenclature and conventions

A dynamic network normally begins with a physical, spatial network that is then defined over time. We assume that movements occur between a set of cities $\mathcal{C}$, which can represent a city, terminal, region of the country, a single shipper or trailer pool, port, or any other physical facility.

Our problem is generally to move goods (generally shipments, but also passengers, equipment or other objects) between pairs of cities. In general, goods moving from an *origin city* r to a *destination city* s, and might be designated D^{rs}. However, for vehicle routing problems, the pickup or delivery location may be fixed at a warehouse, and therefore might be referenced by a single index (e.g. D^r). A city pair (r, s) is usually referred to as a *market* or, in freight terms, a *traffic lane*. In dynamic problems, market demands are defined over time by D_t^{rs}, where t refers to the time the freight enters the network. An important dimension of dynamic flow problems is the nature in which demands are called into the system, known as the *booking process*. The mathematical specification of the booking process also reveals the structure of forecasting uncertainties for future demands. In truckload trucking, rail and container movements, an individual demand usually fills the entire container. In this presentation, we use the word *load* to refer to shipments that occupy an entire vehicle, while *shipment* is used specifically for customer orders that require in-vehicle consolidation.

Vehicles moving between an origin i and destination j often move through a series of intermediate transshipment points. We refer to this itinerary as a *route*, made up of individual transportation segments called *legs*. The timing of the movement of vehicles over a route is referred to as a *schedule*.

In our view of the world, goods move in containers (trailers, boxcars) pulled by vehicles (tractors, locomotives, ships) whose movements are determined by transportation schedules and managed by drivers. We let x and y refer to the flows of loaded and empty containers, respectively. The flow of loaded containers on leg (i, j) in time t is denoted x_{ijt}. The flows of loaded vehicles are similarly denoted v_t, and the movement of empty vehicles is denoted u_t. The flow of goods is referred to using w.

We try to use a consistent vocabulary and notation throughout, but in some instances there is an established literature that has evolved in the context of a particular problem. For example, in freight transportation (such as trucking or rail) we may refer to the dynamic vehicle allocation problem, but we are really referring to the containers (trailers, boxcars) instead of the individual tractors or locomotives. In the dynamic traffic assignment problem, we consider only the movement of goods (no containers or vehicles). In truckload trucking, we ignore the goods, since the movement of a truckload shipment from r to s is equivalent to the movement of the trailer (and hence we focus on the movement of trailers, and simply distinguish between loaded and empty trailers).

4.2. *The dynamic shortest path problem*

Section 2.5 provided results for finding a priori shortest paths over a stochastic network. In this section, we consider deterministic and stochastic shortest path problems which involve time varying data.

4.2.1. *Deterministic shortest paths*

The deterministic dynamic shortest path problem was first introduced by Cooke & Halsey [1966]. They assume that travel times are multiples of δ for some $\delta > 0$. Also, they assume that these travel times are known at times $t_0, t_0 + \delta, t_0 + 2\delta, \ldots$ Now let $f_i(t)$ be the minimum travel time from node i to the destination node d given that you are visiting node i at time t. Cooke and Halsey then introduce the functional:

$$f_i(t) = \begin{cases} \min\limits_{j \neq i} [c_{ij}(t) + f_j(t + c_{ij}(t))] & \forall i \neq d \\ 0 & i = d \end{cases}$$

which can be solved using dynamic programming. This is equivalent to solving a shortest path problem over a time expanded network, where each node in the dynamic network corresponds to a point in space-time in the original network. Dreyfus [1969] simplified this approach by adapting Dijkstra's algorithm, whereby labels v_i are maintained for each node i. At each iteration, the node i with the minimum v_i is made permanent. The labels v_i are updated using:

$$v_j = \min[v_j, v_i + c_{ij}(v_i)] \quad \forall j$$

More recently, Kaufman & Smith [1992] show that Dreyfus' method will solve the deterministic dynamic shortest path problem only if it satisfies a consistency condition given by:

$$s + c_{ij}(s) \leq t + c_{ij}(t) \quad \forall s, t, s \leq t$$

This assumption is known as the overtaking condition in traffic problems. It effectively assumes that if car i enters a link later than car j, then it will also leave the link later. In other words, we do not allow car i to overtake car j on the link.

A variation of the dynamic shortest path problem allows cars to wait at intermediate nodes (sometimes referred to as parking). This was first studied by Halpern [1977], who showed that if the overtaking condition is not imposed, then optimal paths may involve cycling (where cars effectively travel around the block waiting for more favorable conditions). He proposed *parking* at nodes so that cars could wait at a node (instead of cycling) until a more favorable time to traverse the node.

Orda & Rom [1990] consider three types of dynamic networks. The first allows unlimited waiting at any node; the second allows waiting only at the origin node, and the third where no waiting is allowed anywhere. They show that if the departure time from the source node is unrestricted, then a shortest path can be found that is simple and achieves a delay as short as the most unrestricted path. They also give examples that show that under the assumption of no waiting (and arbitrary link travel time functions) can produce networks with no finite optimal path.

4.2.2. Stochastic shortest paths

Shortest path problems over stochastic, dynamic networks come in several forms depending on the motivating application. Bertsekas & Tsitsiklis [1991] consider a very general stochastic shortest path problem where at each node i we choose a probability distribution over the set of outbound arcs so that the expected cost to the destination is a minimum. The problem is formulated as a Markov decision process and used to investigate the properties of an optimal policy. Standard methods are proposed for finding the optimal policy.

Andreatta & Romeo [1988] consider a network where individual arcs may or may not be present. A vehicle arriving to a node i may then learn that arc (i, j) is no longer present, and must then adopt a recourse strategy to develop a new path. Each arc in the network has a known probability of being blocked, but they also assume that once an arc (i, j) is found to be blocked, no other arc emanating from i can be blocked. Andreatta [1987] shows that this problem is harder than $\mathcal{NP}$, and also shows that the cost of path derived from a deterministic approximation of the network can be arbitrarily bad relative to the optimal stochastic path.

Psaraftis & Tsitsiklis [1990] investigate a problem motivated by the routing of ships over the sea subject to weather delays. The assumption is that at specific 'nodes' the ship may move immediately over one or more arcs, or wait until more favorable conditions arise. Two variations of this problem arise because we can assume that the ship will wait and then choose an outbound arc, or choose an outbound arc first and then wait for better conditions.

Assume that costs are known functions of variables that evolve according to a Markov process. More precisely, the cost of traversing each arc (i, j) of the graph is a known function $f_{ij}(e_i)$ of the state e_i of a variable at node i at the time the vehicle departs from node i on its way to node j. The variables are mutually independent, and each is governed by a finite-state Markov process, with discrete time transitions. The actual state e_i of the variable at node i is known by the vehicle only when it is at this node. If the vehicle reaches node i and then decides to immediately go to node j, it incurs a cost $f_{ij}(e_i)$. The vehicle may also want to wait at node i in anticipation of a more favorable state at i. It can wait as much as it wants, but at a cost of C per state transition. The problem is then to decide on a policy that minimizes the expected total cost of a traversal, say from source node 1 to destination node n.

The main results obtained by Psaraftis & Tsitsiklis [1990] are the following: (1) Focusing on an individual arc of the network, the authors develop and analyze different policies for traversing that arc. Recognizing that this problem can be formulated as an 'infinite-horizon, total cost, stochastic dynamic programming problem' [see Bertsekas, 1987], they investigate two methods, the *successive* *approximation* procedure and the *policy iteration* procedure, and prove the computational superiority of the latter. They also consider linear programming in order to solve this infinite horizon dynamic programming problem, and mention some simple case for which closed form solutions are obtainable. (2) Considering the entire network, they then consider a suboptimal policy in which the direction decision precedes the 'go/no go' decision, the latter being solved as before. Finally they develop an optimal policy in which the previous two decisions are reversed, and they show that it is also better in terms of computational effort.

4.3. *The dynamic traffic assignment problem*

The dynamic traffic assignment problem addresses the problem of routing goods over a network from origin (representing the origin location as well as the time the flow entered the network) to the destination (the time of arrival to the destination is an output of the model). In between, the flow will move over one or more links which may be capacitated or have nonlinear costs. The problem is in routing flows between multiple origins to multiple destinations, and handling the competition for limited capacity over a network of transportation services.

In this section, we consider the pure traffic assignment problem. We also focus only on freight problems which can be formulated as system optimization (global cost minimization) problems, and exclude passenger applications, such as the dynamic traffic assignment problem for auto traffic, where passenger behavior is typically modeled as an equilibrium problem. For research on dynamic traffic equilibrium problems, see Friesz, Luque, Tobin & Wie [1995], Friesz, Bernstein, Smith, Tobin & Wie [1993], Ran, Boyce & LeBlanc [1992], and the references cited there.

We consider three versions of the dynamic traffic assignment problem. Section 4.3.1 considers the capacitated, linear traffic assignment problem in discrete time, which arises when routing freight flows over a fixed network of transportation services. Next, Section 4.3.2 describes the nonlinear dynamic traffic assignment, which models congestion effects without hard capacity constraints. Both of these formulations are presented in a discrete time framework. Finally, 4.3.3 describes evacuation models, which represent an important special case of the dynamic traffic assignment problem.

4.3.1. *The capacitated version*

To formulate the dynamic traffic assignment problem, we distinguish between flows of shipments, 'containers' and scheduled transportation services (vehicles). The dynamic 'vehicle' allocation problem focuses on the flows of containers. Here, we focus on the flows of shipments over a scheduled network of transportation services. Define:

D_t^{rs} = total flow of shipments moving from origin r to destination s, originating in period t.

$h^{rs}(\tau)$ = service penalty assessed on shipments moving from r to s delivered in τ time periods.

P_t^{rs} = set of feasible space-time paths joining r and s, departing at time t, moving over a dynamic network.

τ_{pt}^{rs} = travel time for path $p \in P_t^{rs}$.

$$\sigma_{ij}^{p,rs}(t_1, t_2) = \begin{cases} 1 & \text{if link } (i, j) \text{ is on path } p \text{ joining } r \text{ and } s, \text{ departing in period } t_1 \text{ and moving over link } (i, j) \text{ departing in period } t_2, \\ 0 & \text{otherwise.} \end{cases}$$

w_{pt}^{rs} = total flow on path p joining r and s, departing in period t.

w_{ijt} = total flow on link (i, j) departing in period p.

v_{ijt} = total transportation capacity available between i and j, departing at time t, determined by the routing of vehicles.

c_{ij}^w = cost per unit to move a shipment over link (i, j).

The flow of vehicles is given by v_{ijt}, which is assumed to be expressed in the units of the flow w_{ijt}. Note that we are not concerned with how the capacity v_{ijt} is obtained (this is considered later). We have also adopted a path flow formulation, since recent experimental work has shown that this approach significantly outperforms more classical link-based formulations (see Section 5.5.1).

It is important to model transfer activities, where shipments must be unloaded, sorted and reloaded. These activities, as well as their costs and constraints, are easily modeled using standard network techniques (for example, splitting nodes to model transfer costs). Interestingly, while there is a cost for handling shipments at transshipment points, the cost per shipment c_{ij}^w for a transportation link is typically zero, since transportation costs are normally associated with the flows of vehicles reflected by v_{ijt}. Interestingly, existing dynamic traffic assignment models have not really taken advantage of this.

The traffic assignment problem is most easily visualized using the network in Figure 7. Service penalties are handled using a fairly standard device. The flow D_t^{rs} which originates at time t is assumed to exit through a supersink for destination s. The costs on links into the supersink reflect service penalties, where the model is significantly simplified if the service penalty function, $h^{rs}(\tau)$, depends only on the destination and is linear in τ. A mathematical statement of what we refer to as the dynamic traffic assignment problem is given by:

$$F_w(v) = \min_{w_{pt}^{rs}} \sum_{t=0}^{P} \left[(c^w)^T \omega_t + \sum_{r \in C} \sum_{s \in C} \sum_{p \in P_t^{rs}} w_{pt}^{rs} h_p^{rs}(\tau_{pt}^{rs}) \right] \tag{64}$$

subject to, for $t = 0, \ldots, P$

$$\sum_{p \in P_t^{rs}} w_{pt}^{rs} = D_t^{rs} \qquad r, s \in C$$

$$w_{pt}^{rs} \geq 0 \qquad r, s \in C, \ p \in P_t^{rs}$$

$$w_{ijt} - \sum_{t' \leq t} \sum_{r \in C} \sum_{s \in C} \sum_{p \in P_t^{rs}} w_p^{rs}(t') \sigma_{ij}^{p,rs}(t', t) = 0 \qquad i, j \in C$$

$$w_{ijt} \leq v_{ijt} \qquad i, j \in C \tag{65}$$

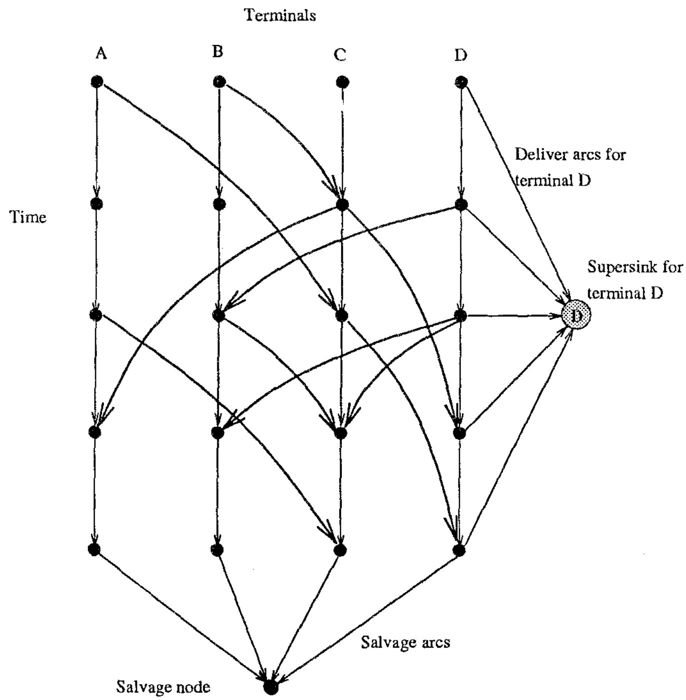


Fig. 7. Space-time network for dynamic traffic assignment.

$F_w(v)$ is the optimum shipment routing costs given a set of transportation services v_t.

From an algorithmic perspective, the complicating feature is the bundle constraint (65). The development of algorithms for this problem is discussed in Section 5.5.

Problem (64) focuses on moving a shipment through a network of transportation services, with transfers through intermediate transshipment points for sorting. The same problem arises when moving a loaded container (vehicle) through the network. In rail and container shipping, the shipment and loaded railcar or container are equivalent. In LTL trucking, the problem of moving a trailer loaded with shipments from origin to destination is also a type of traffic assignment problem. In this problem, however, the movements are constrained not by available transportation capacity, v_{ijt}, but rather by the availability of drivers governed by work rules.

4.3.2. *The nonlinear version*

The model above does not account for the possibility of congestion on the links of the network. By this, we refer explicitly to a process whereby travel times on a link increase as a nonlinear function of the flow on the link. Travel times that are a function of flow challenge the basic framework of a dynamic network, where a link represents an explicit movement in space-time. Classical dynamic networks can approximate congestion effects by capacitating links, forcing flow that cannot move over a link to hold in inventory to a later point in time. By contrast, a link with a nonlinear congestion function would produce link traversal times that increase in a nonlinear fashion with flow.

An alternative model, motivated by models of auto traffic but using a system optimization routing objective, has been developed which accounts for congestion on each link which is a function of the flow. This work was initiated by Merchant & Nemhauser [1978], which considered the problem of routing users over a dynamic network into a single destination. This research motivated a series of papers. Ho [1980], drawing on ideas in the original Merchant and Nemhauser paper, introduced an algorithm that guarantees an optimal solution to a piecewise linear version of the original model. Carey [1987] introduces a slight revision that transformed the nonconvex problem into a convex one. Most recently, Birge & Ho [1987] introduced a stochastic formulation of the same model. In this model, Ho's results are used for each scenario of a stochastic program. Merchant and Nemhauser [1978b] and Carey [1986] describe special constraint qualifications necessary to guarantee optimality.

We use the formulation in Carey [1987], but adopt our own notation from above. The basic formulation introduced by Merchant and Nemhauser models congestion by restricting the rate at which flow leaves a link, a rate that varies as a nonlinear function of flow. Flow that is unable to leave the link in a given time period is assumed to hold until the next time period. As a result, we have to divide the 'flow' on a link into three categories: flow entering a link (in a given time period), flow leaving a link, and flow that is held in inventory from one time period to the next. Keeping with our earlier notation, we use ω_t to denote flows, and define:

w^e_{ijt} = flow *entering* link (i, j) at the beginning of time period t.

w^h_{ijt} = flow *held in inventory* on link (i, j) at the beginning of time period t, which has been held over from the previous time period.

w^d_{ijt} = flow *departing* from link (i, j) during period t.

$g_{ij}(w^h_{ijt})$ = maximum allowable outflow from link (i, j) in period t.

$c_{ij}(w^h_{ijt})$ = total cost incurred from holding w^h_{ijt} units of flow on link (i, j) in period t.

R_{it} = exogenous flow entering the network at node i in period t (with a common destination).

We assume that all values are assigned initial values for $t = 0$. The dynamic traffic assignment with congestion can be stated as follows:

$$\min_{w^e, w^h, w^d} \sum_{t=0}^{P} \sum_{i,j} c_{ij}(w^h_{ijt}) \tag{66}$$

subject to, for $t = 0, \ldots, P$

$$
\begin{aligned}
&(w^h_{ijt} + w^e_{ijt} - w^d_{ijt}) - w_{ij,h,t+1} = 0 && i, j \in C \\
&\sum_{k \in C} w^e_{jkt} - \sum_{i \in C} w^d_{ijt} = R_{jt} && j \in C \\
&w^d_{ijt} - g_{ij}(w^h_{ijt}) \leq 0 && i, j \in C \\
&w^e_{ijt}, w^h_{ijt}, w^d_{ijt} \geq 0 && i, j \in C
\end{aligned}
\tag{67}
$$

The important characteristic of this formulation is the exit functions $g_{ij}(w^h_{ijt})$ which restrict the amount of flow allowed to leave the link in a given time period. Note that we do not restrict the problem if we require the exit functions to satisfy $g(w) \leq w$. If $g(w) = w$, then the problem is unconstrained and we have an instance of an uncapacitated traffic assignment problem. Normally, congestion reduces the efficiency of the network, resulting in situations where $g(w) < w$. It is reasonable to expect that $g(w)$ is concave in w, which implies that constraint (67) is convex (that is, the set of values w^h_{ijt} that satisfy (67) is a convex set). In their original paper, Merchant and Nemhauser assume that (67) holds with equality, which is not convex, and demonstrate that the resulting problem can have local optima. They investigate a form of the model where the exit functions $g_{ij}(w^h_{ijt})$ are replaced with piecewise linear approximations, and introduce the following cost function assumption (CFA):

(a) $c_{ij}(w^h_{ijt})$ is nonnegative, convex in w^h_{ijt} and nondecreasing in t.

(b) Let (i, j) and (k, l) represent two distinct arcs on a path to the destination in time period t, with (k, l) closer to the destination. Then $\partial c_{ij}(u, t)/\partial u \geq \partial c_{ij}(v, t)/\partial v$ for all u and v.

(c) The inequality in (b) is a strict inequality when $t = P$.

Merchant and Nemhauser consider the case where the nonlinear functions $g_{ij}(w^h_{ijt})$ are replaced with piecewise linear approximations. For this case, they show that CFA guarantees that the global optimal solution can be expressed as a basic solution of a linear program. Ho [1980] exploits these properties and presents a method for finding the optimal solution by solving at most $P + 1$ linear programs (where P is the length of the planning horizon). Birge & Ho [1987] extend this methodology to handle uncertainty in the input flows R_t.

Carey [1987] introduces the formulation we use above where the exit functions are used as inequalities as in (67). The difference $g_{ij}(w^h_{ijt}) - w^h_{ijt}$, which gives the extent to which the actual flow departures fall below the capacity of the link, is referred to as a *flow control*, with the interpretation that a master controller might want the actual flow departure to be below the capacity of the link (which is consistent with a system optimization formulation). This raises the question of the conditions that would produce positive flow controls. For this question, Carey introduces the following exit function assumption (EFA):

(a) $0 \leq g_{ij}(w_t) \leq w_{ijt}$;

(b) $0 \leq \partial g_{ij}(w_t)/\partial w_t < 1$;

(c) $g_{ij}(0) = 0$.

Carey shows that under EFA, and a weaker form of CFA, that constraint (67) will always be binding. The CFA conditions are needed to ensure that the model will try to push as much flow as possible into the supersink before the end of the planning horizon. Condition (b) eliminates conditions that encourage holding flow in inventory, while condition (c) is a tie-breaking condition to force flow out of the network before the final period.

It should be noted that as stated, this model does not incorporate travel times. If there is no congestion, then all flow moves through the network in the same time period. The model can be modified to incorporate travel times through suitable changes to constraint (67), although a modification to EFA may be necessary to ensure zero flow controls.

This line of research has focused on problems with flows into a single destination, allowing a single commodity formulation. Carey argues that multiple destination problems can be transformed into single destination problems by adding a supersink. Appropriate use of constraints on links into the supersink can guarantee that the correct amount of flow exits the network from a particular destination, but such a formulation is unable to guarantee conservation of flow when commodity flows are specified in an origin/destination format.

4.3.3. Evacuation models

A special case of the traffic assignment problem is one that minimizes the time required to move all flow from source to sink. Chalmet, Francis & Saunders [1982] presents a network model to solve a building evacuation problem. Jarvis & Ratliff [1982] show that if the cost of exiting in period t is c_t, and $c_1 \leq c_2 \leq \ldots \leq c_P$, then the optimal solution a) minimizes the average transit time, b) maximizes the total number of evacuees in the interval $(1, t)$ for $1 \leq t \leq P$, and c) minimizes the time at which the network is cleared. Choi, Hamacher & Tufekci [1988] present algorithms to handle building evacuation problems with side constraints. Finally, Hamacher & Tufeki [1987] develop additional properties of flows for evacuation problems.

A separate line of research has considered stochastic evacuation problems. Karbowicz and MacGregor Smith [1984] present a k-shortest path algorithm for handling random events when evacuating a stochastic networks. MacGregor Smith & Towsley [1981] describe a queueing network model for building evacuations. More recently, MacGregor Smith [1993] presents a method for multiobjective stochastic evacuation problems. In this model, efficient routes are identified which are pareto optimal in terms of total distance travelled and total evacuation time. The goal of the process is to identify a set of evacuation routes for each origin. The algorithm proceeds by identifying *noninferior* (NI) paths, and then testing these paths using a queueing network simulator. This simulator measures the level of congestion resulting from a set of paths. After each iteration, new paths are generated and then reevaluated using the queueing network simulator.

Evacuation problems are important in traffic assignment problems for freight. Here, the problem is routing freight over a dynamic network of transportation services. In contrast with classical routing problems, we are interested primarily in how long it takes to move shipments over the network, as opposed to minimizing costs (which are determined by the flows of vehicles, rather than the flows of goods). Evacuation problems are characterized by an initial supply of flow into the network, whereas traffic assignment problems have a continuous flow of goods that must be moved to their destination.

4.4. Production and inventory planning

Production and inventory planning is a special type of routing problem, but the field is important since it is home to many of the basic concepts in dynamic models. Section 4.4.1 provides a brief summary of some of the literature in this area, with special emphasis on dynamic network models of production and inventory planning. Then, Section 4.4.2 presents a dynamic network model that illustrates a dynamic inventory network model, and summarizes some key developments in this context.

4.4.1. Survey and applications

Production and inventory planning is one of the most widely studied topics in the operations research literature. These problems have often been formulated using dynamic networks, and hence represents an important step in the historical development of dynamic models. We do not attempt to review this substantial literature, but instead refer the reader to reviews such as Zahorik, Thomas & Trigeiro [1984] and Bahl, Ritzman & Gupta [1987], or any of the popular textbooks in the field. Important applications of dynamic networks for production, inventory and transportation planning include Glover, Jones, Karney, Klingman & Mote [1979] and Klingman & Mote [1982]. These applications demonstrate the effectiveness of linear network models for aggregate production planning. Our motivation for reviewing this literature stems from the observation that the problem of planning the movements of large fleets of vehicles can be viewed in the context of multilocation inventory problems. While classical inventory and distribution problems focus on getting goods from plants to customers, dynamic fleet management can be viewed as moving empty vehicles and containers from surplus areas to deficit areas.

Dynamic production and inventory planning problems are often formulated using dynamic transportation problems. An early introduction to this problem is given in Szwarc [1971b], and an excellent review of the literature on dynamic transportation problems is given in Bookbinder & Sethi [1980]. Aronson [1989] reviews in depth algorithms for dynamic transportation problems motivated by production planning and scheduling, in particular, the forward network simplex algorithm introduced by Aronson & Chen [1986]. This work, which is reviewed further in Section 5, shows that solution algorithms can be accelerated by taking advantage of the dynamic properties of the network. Bowman [1956] showed that the linear production planning problem over time could be solved as a single transportation problem, assuming no constraints on inventories. Bellmore, Ecklof & Nemhauser [1969] reformulate this problem as a dynamic transshipment network, which is actually a dynamic transportation problem, and describe a modification of the Busacker & Gowen [1961] algorithm for dynamic networks (this algorithm solves a sequence of least-cost flow augmenting path problems). A number of authors have also considered a minimum time formulation of the problem [Szwarc, 1966; Hammer, 1969, 1971; Tapiero & Soliman, 1972; and Tapiero, 1975].

The research on stochastic multilocation inventory problems is much more limited. For the most part, the literature on inventory planning problems has focused on the analysis of policies with good performance under uncertainty with stationary demand patterns. Stochastic dynamic models quickly become computationally intractable. Shapiro [1993] reviews some of the literature on stochastic production and distribution models. An important contribution to this literature is a series of papers by Karmarkar who uses a stochastic programming framework to study distribution strategies as well as developing bounds and approximations for the objective function [see Karmarkar, 1979, 1981a, b, 1987]. Karmarkar [1981b] shows that the form of an optimal policy for a multiperiod, multilocation inventory problem is the same as for one-period, which is to say that a base-stock policy is optimal. In this policy, if the inventory falls below a base-stock y, then it is optimal to bring the inventory up to the level y. Karmarkar [1987] then develops bounds and approximations for the expected recourse function for this class of problems.

4.4.2. Models

In logistics applications, the determination of how much to ship, when and where, is typically determined by the solution of a production and inventory planning problem. The classical production and inventory planning problem is illustrated in Figure 8. Here, we have modeled demands as upper bounds on links from each time period into a supersink. Thus, it is possible not to satisfy all the demand, presumably at a penalty.

A more general form of this basic problem is the multiregion, dynamic production and inventory planning problem, which can be stated mathematically as:

Decision variables

w_{ijt} = flow of goods from city i to city j, in period t.

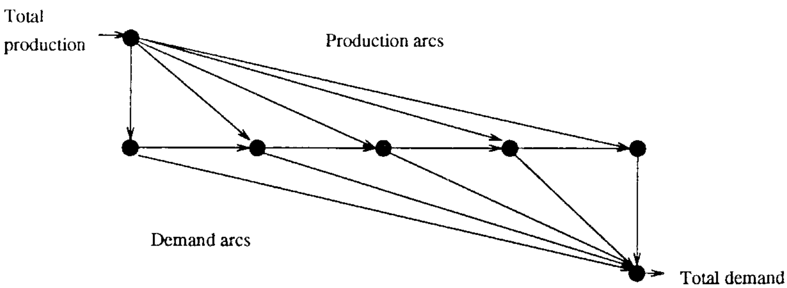


Fig. 8. for production and inventory planning at a single location.

w_t $= \{\ldots, w_{ijt}, \ldots\}$.

W_{it} = net surplus or deficit of goods at city i in time t, where W_{it} is positive if i is a surplus point, and negative if a deficit.

B_{t_1,t_2} = node-arc incidence matrix where the rows correspond to cities $i \in \mathcal{C}$ at time $t = t_2$, and columns correspond to flows between cities (i, t_1) and (j, t_2). An element equals minus one if flow departing from i at time t_1 arrives in j at time t_2, and zero otherwise, for $t_1 \leq t_2$.

A_t = node-arc incidence matrix where the rows correspond to cities i at time t, and columns correspond to flows between cities i at time t and cities j at time $t' \geq t$. An element is minus one if the arc is entering node (i, t) and minus one if it is emanating from the node.

$$\min_{w_t} \sum_{t=0}^{P} \alpha^t c^T w_t \tag{68}$$

subject to, for $t = 0, \ldots, P$

$$\begin{array}{llll} A_0 w_0 & & & = W_0 \\ B_{0,1} w_0 + A_1 w_1 & & & = W_1 \\ B_{0,2} w_0 + B_{1,2} w_1 + A_2 w_2 & & & = W_2 \\ B_{0,3} w_0 + B_{1,3} w_1 + B_{2,3} w_2 + A_3 w_3 & & & = W_3 \\ \vdots & \ddots & & \vdots \\ B_{0,t} w_0 + B_{1,t} w_1 + B_{2,t} w_2 + & \ldots & + A_t w_t & = W_t \end{array} \tag{69}$$

Equations (68)–(69) describe a general, dynamic production planning problem. If a time period is relatively long, such as a month, then we generally have that $B_{t_1,t_2} = 0$ for $i \neq j$. In this case, we have an instance of a dynamic inventory network as shown in Figure 9, with the specific form of a dynamic transportation problem. A good example of a practical application of this network is given in Glover, Jones, Karney, Klingman & Mote [1979] and Klingman & Mote [1982].

The solution of dynamic inventory networks of this type has been investigated in depth by Aronson & Chen [1986]. This work applies the principles of forward linear programming, described in Aronson & Thompson [1984] and Aronson [1980], and in particular the review article Aronson [1989]. These methods are based on the principles of decision horizons which tend to arise in practice in this class of models (see Aronson & Chen [1985, 1989].

4.5. *The dynamic vehicle allocation problem*

The dynamic vehicle allocation problem arises in a variety of settings which involve the management of fleets of vehicles over time. Below review applications in trucking, rail, container shipping, and air. Related applications can be found in the management of taxi cabs and rental fleets (especially the so-called 'one-way' rentals of trucks). Following this, we present some simple models for the dynamic vehicle allocation problem.

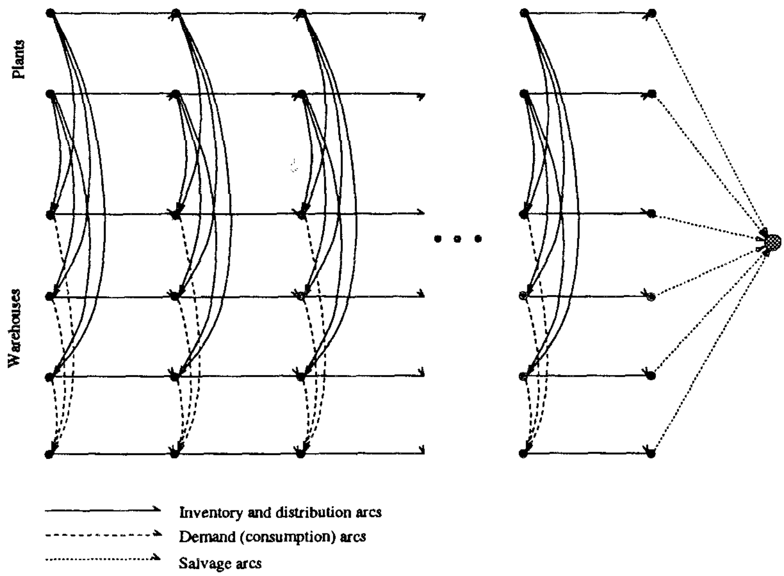


Fig. 9. Dynamic network for production and inventory planning.

4.5.1. Applications

Trucking. For truckload trucking, the primary problem addressed in the research literature has been the dynamic vehicle allocation problem for managing large fleets of trucks. Powell, Sheffi & Thiriez [1984] present a nonlinear dynamic network model for accounting for uncertainties in forecasted demands. Powell [1986] refines this model to allow for stochastic vehicle inventories. This model extends [Powell, Sheffi & Thiriez, 1984] to allow for the possibility that trucks that are not needed will remain in inventory, and extends the model in Jordan & Turnquist [1983] by tracking both loaded and empty movements. Powell [1987] further extends this model by providing the most realistic model of future vehicle trajectories. Whereas prior research makes very restrictive assumptions about how a truck may be used under uncertainty, Powell [1987] provides for a very general model of truck dispatching under uncertainty. A more formal mathematical model is provided in Powell [1988], where several different models are formulated for the same problem. This paper provides the first formal model of the dynamic vehicle allocation problem as a dynamic network with random arc capacities. Frantzeskakis & Powell [1990] introduce a new heuristic for solving multistage dynamic networks with random arc capacities, motivated by truckload motor carriers. These results have been further extended in Powell & Cheung [1992]. A real application of these results is given in Powell, Sheffi, Nickerson, Butterbaugh & Atherton [1988].

Rail. As one of the oldest modes of transportation, railroads have been a popular subject for operations research. The main challenge has been managing large fleets of rail cars. Feeney [1957], Leddon & Wrathall [1967], Gorenstein, Poley & White [1971] and Herren [1973, 1977] represent a number of early examples of efforts to optimize fleets of rail cars. Misra [1972] formulates the problem as a linear program, while White [1972] presents a dynamic transshipment network over a finite planning horizon. White & Bomberault [1969] use the dynamic structure of the network to develop a specialized algorithm, one of the earliest efforts to specialize an algorithm for a dynamic network. Mendiratta [1981] and Mendiratta & Turnquist [1982] present inventory models for managing empty cars, taking into account the decentralized nature of the decision-making process. Jordan & Turnquist [1983] present the first stochastic model of the empty car management problem. Ratcliffe, Vinod & Sparrow [1984] use a simulation model of empty freight cars. Glickman & Sherali [1985] address the problem of pooled fleets of empty cars, recognizing that railroads share fleets of cars. Shan [1985] uses a dynamic, multicommodity network flow model to handle multiple car types, using resource directive decomposition to solve the resulting network. Chih [1986] extends this model to handle multiple railroads. Despite the increasing sophistication of these models, there is little real evidence that they are being adopted and used by the railroads. By contrast, much simpler myopic models, such as the transportation formulations given in Turnquist [1986] and Turnquist & Markowicz [1989], are seeing wider acceptance.

Other researchers are going beyond the problem of managing empty cars. Haghani [1989] presents a combined model for train makeup and empty car repositioning, representing one of the earliest efforts to address both the flows of loaded and empty cars. Chih, Hornung, Rothenberg & Kornhauser [1990] consider the problem of managing locomotives. Smith & Sheffi [1990] present a locomotive distribution model that accounts for uncertainty in the need for locomotives, using a simple recourse strategy to handle the effects of uncertainty. Kraay, Harker & Chen [1989] address the problem of dynamically managing the movement of trains over a rail line, which requires optimizing the use of sidings to allow for train passings.

Container shipping. Ocean applications of dynamic models arise when planning the movement of ocean vessels, and the optimization of fleets of containers over a global logistics network. Dantzig & Fulkerson [1954] provide one of the earliest applications of optimization over a dynamic network to minimize the number of tankers required to meet a given schedule. Other efforts to optimize the movement of vessels include Brown, Graves & Ronen [1987], Psaraftis, Orlin, Bienstock & Thompson [1985], Fisher & Rosenwein [1989], and Perakis & Papadakis [1989]. Ermoliev, Krivets & Petukhov [1976] and Florez [1986] consider the optimization of containers. Crainic, Gendreau & Dejax [1992] looks at the dynamic management of containers over land within a region near a port.

Air. Dynamic problems in air transportation include the assignment of aircraft to routes, crew scheduling, pricing and booking problems, and the dynamic management of aircraft between airports (the air traffic control problems). Dantzig & Ferguson [1956] use the fleet assignment problem as an early example of linear programming under uncertainty. Magnanti & Simpson [1978] describe a series of dynamic network models with side constraints to handle fleet assignment. The crew scheduling problem for airlines has become a popular area of research. Surveys of this area are given by Arabeyre, Feranley, Steiger & Teather [1969], Marsten & Shepardson [1981], and more recently by Crainic & Rousseau [1987]. The common approach used in this area is based on set partitioning problems to choose from among the best set of possible crew schedules. Methods contrast based on whether a generator is used to generate all possible 'reasonable' schedules, or whether column generation techniques are used. A different approach is suggested by Ball & Roberts [1985] which uses a matching algorithm to sequentially generate possible crew schedules.

The second problem is the dynamic management of aircraft moving between airports, sometimes referred to as the flow management problem in air traffic control. An important reference in this is Andreatta & Romanin-Jacur [1987]. An excellent discussion of models and issues arising in the flow management problem is given in Odoni [1986]. Mulvey & Zenios [1987] give a nonlinear, dynamic network model for routing aircraft. Bielli, Calicchio, Micoletti & Ricciardelli [1982] also formulate the flow control problem as a dynamic network.

4.5.2. Models for the dynamic vehicle allocation problem

An important class of problems that can be solved as pure networks are dynamic fleet management problems involving a single equipment type. Dynamic fleet management refers broadly to the problem of managing a fleet of vehicles (containers) over time to maximize a set of objectives. A component of this problem is the *dynamic vehicle allocation problem* which specifically addresses decisions regarding the repositioning of empty vehicles, as well as load acceptance and rejection. A presentation of alternative models and formulations of the dynamic vehicle allocation problem is given in Powell [1988].

The basic objective of dynamic vehicle allocation is to generate revenues by carrying a set of *loads*, each of which fills an entire vehicle (there is no consolidation). Carrying a load implies a vehicle must move from one city to another, at which point the vehicle becomes empty and must be assigned to a new load. In some instances, more loads will terminate in a particular area than originate, thus requiring excess vehicles to be repositioned empty out of one area and into another in anticipation of future loads. Thus, there are three types of activities for vehicles: moving loaded, moving empty, and holding in inventory (doing nothing). Empty moves may also be divided between moving empty to satisfy an actual demand, and moving empty to satisfy a forecasted demand.

Dynamic fleet management encompasses the dynamic vehicle allocation problem, as well as load solicitation, load evaluation, spot pricing, tactical sales planning, and service planning. Solutions to this broader set of questions depends, however, on the solution of the basic dynamic vehicle allocation problem. We first present the single vehicle version, and then provide a multifleet model.

Single vehicle type. Our basic formulation of the DVA can be expressed easily as a dynamic network. First define:

Decision variables

x_{ijt} = number of vehicles moving loaded from city i to city j, originating in period t.

y_{ijt} = number of vehicles moving empty from city i to city j, originating in period t.

Physical parameters

$\mathcal{C}$ = set of cities (points) where loads originate and terminate.

τ_{ij} = travel time (in nonnegative integer periods) from city i to city j (we use the same time for all loaded and empty movements).

r_{ij} = net contribution (revenue minus direct operating costs) generated from moving loaded from i to j (we assume, for notational simplicity, that contribution is not a function of time).

c_{ij} = cost of moving empty from i to j (c_{ii} is the marginal cost of holding vehicles in inventory). As with revenue, we assume that costs are not a function of time; this assumption is easily relaxed.

Activity variables

R_{it} = number of vehicles entering the system for the first time in period t.

D_t^{rs} = market demand, giving the number of loads *available* to be moved from origin r to destination s, originating in period t.

Model parameters

P = length of the planning horizon.

α = discount factor per period.

In our notation, we use a superscript (r, s) to denote transportation *markets* which express the originating and terminating point of a load or shipment. We use a subscripted (i, j) to express transportation movements between cities. In our basic model of the dynamic vehicle allocation problem, which is most directly applicable to truckload trucking, vehicles (trucks) are assumed to move directly from origin to destination. Later, we consider instances of problems where the transportation of a shipment from origin to destination must pass over several transportation links.

The basic dynamic vehicle allocation problem seeks to maximize total (discounted) profits over the planning horizon P, as follows:

$$\max_{x_t, y_t} \sum_{t=0}^{P} (r^T x_t - c^T y_t)\alpha^t \tag{70}$$

subject to, for $t = 0, \ldots, P$

$$\sum_{j\in\mathcal{C}}(x_{ijt} + y_{ijt}) - \sum_{k\in\mathcal{C}}\left[x_{ki,(t-\tau_{ki})} + y_{ki,(t-\tau_{ki})}\right] = R_{it} \qquad i \in \mathcal{C} \tag{71a}$$

$$x_{ijt} \leq D_t^{ij} \qquad i, j \in \mathcal{C}$$

$$x_{ijt}, y_{ijt} \geq 0 \qquad i, j \in \mathcal{C} \tag{71b}$$

$$x_t, y_t = 0 \qquad t < 0$$

The discount factor α is sometimes included in dynamic models as a heuristic mechanism for accounting for forecasting uncertainties. The choice of planning horizon and the choice of α is discussed in Section 3.4.3.

Equation (71a) enforces flow conservation at the beginning of each time period, where we use the convention of flow out minus flow in. Thus $R_{it} > 0$ when vehicles enter the network for the first time. Equation (71b) is the *demand constraint* where market demands limit the number of loaded movements. Note that there is a single transportation leg (i, j) associated with each market (r, s).

The object of this basic formulation of the DVA is to determine loaded movements x_{ijt}, empty movements, y_{ijt}, and inventory movements, y_{iit}, to maximize total profits over a fixed horizon. This problem also determines which loads should be rejected, given by $D_t^{ij} - x_{ijt}$ (it is common to assess a penalty r_{ij}^p for refusing loads; this results in a modified revenue $\hat{r}_{ij} = r_{ij} + r_{ij}^p$). With a little creativity, this model can be used to handle some of the other dimensions of dynamic fleet management.

There are several important variations of the basic DVA. First, a static form of the problem considers the myopic problem of repositioning vehicles empty now to meet known and forecasted demands within a specific time period (perhaps a week). This model is often used by railroads and container companies, and does not explicitly model the loaded movement. A more general form of this model also considers only the repositioning of empty vehicles, but actually models the holding of vehicles in inventory over time. If empty movements are allowed in future time periods, then this model is specifying not only where to move empty, but when. Another example is the ground holding problem for air traffic control, which consists of determining when to allow an aircraft to depart from an airport, anticipating future demands and landing capacities of other airports. The ground holding problem is an instance of a problem in which we already know *where* the vehicles are going but need to know *when* they should depart.

Multiple vehicle types. The single commodity formulation of the DVA is a powerful framework with practical applications in industry, particular in truckload trucking. At the same time, most applications require consideration of multiple equipment types, where it is possible to substitute some equipment types to handle a given demand. Of course, if there are no substitutions, then the problem can be solved as a sequence of single commodity DVA's. More commonly, some level of substitution is allowed. To formulate this problem, we start with (70)–(71) and introduce the index ℓ to represent flows of equipment type ℓ. So $x_{ijt}(\ell)$ and $y_{ijt}(\ell)$

represent loaded and empty flows of equipment type ℓ, and $D_{ijt}(\ell)$ is the market demand for equipment type ℓ. In addition, define:

$\mathbf{E}$ = set of equipment types.

$D_t^{rs}(\ell)$ = market demand for equipment type ℓ.

$x_{ijt}(\ell, m)$ = flow of equipment type ℓ used to satisfy demand $D_t^{ij}(m)$, $\ell, m \in \mathbf{E}$, $i, j \in \mathcal{C}$.

$x_{ijt}(\ell) = \sum_{m \in \mathbf{E}} x_{ijt}(\ell, m)$.

$\sigma_{ijt}(\ell, m) = \begin{cases} 1 & \text{if equipment type } \ell \text{ can be used to service the demand } D_{ijt}(m), \\ 0 & \text{otherwise.} \end{cases}$

$\mathbf{E}^m = \{\ell \mid \sigma^{\ell,m} > 0\}$.

We use $\sigma(\ell, m)$ as an indicator variable. However, we could allow $0 \leq \sigma(\ell, m) \leq 1$, (where $\sigma(\ell, \ell) = 1$), and give it the interpretation as the fraction of $D_t(m)$ that is realized if it is serviced using equipment type ℓ (allowing lost demand when a less appropriate equipment type is used). The multiple equipment DVA can now be stated as follows:

$$\max_{x_t(\ell), y_t(\ell)} \sum_{\ell \in \mathbf{E}} \sum_{t=0}^{P} \left[r(\ell)^T x_t(\ell) - c(\ell)^T y_t(\ell)\right]\alpha^t \tag{72}$$

subject to, for $t = 0, \ldots, P$

$$\sum_{j \in C} \left(x_{ijt}(\ell) + y_{ijt}(\ell)\right) - \sum_{k \in C} \left(x_{ki,(t-\tau_{ki})}(\ell) + y_{ki,(t-t_{ki})}(\ell)\right) = R_{it}(\ell) \qquad i \in \mathcal{C},\ \ell \in \mathbf{E} \tag{73}$$

$$\sum_{\ell \in \mathbf{E}^m} x_{ijt}(\ell, m)/\sigma(\ell, m) \leq D_{ijt}(m) \qquad i, j \in \mathcal{C},\ m \in \mathbf{E}, \tag{74}$$

$$x_{ijt}(\ell),\ y_{ijt}(\ell) \geq 0 \qquad i, j \in \mathcal{C},\ \ell \in \mathbf{E} \tag{75}$$

Problem (72)–(75) is similar to (70)–(71) with the obvious extensions to account for multiple equipment types. The important difference is that the presence of equipment substitutions transforms the demand constraint (71b) into the bundle constraint (74). This problem is similar in style to the aircraft fleet assignment problem formulated in Simpson & Magnanti [1978]. In this model, *demands* represent scheduled flights, where a particular flight may be satisfied by more than one type of aircraft (if too small an aircraft is used, the conversion factors $\sigma^{\ell,m}$ can be used to model lost passengers). If the problem is not too large (the definition of which seems to change annually) the problem can be solved as a linear program (possibly subject to integrality constraints). Alternatively, decomposition methods can be used to exploit the underlying network structure (as described in depth in the Simpson and Magnanti report). However, if the upper bounds are equal to one, as would occur in airline fleet assignment models, then integrality becomes more important, and different solution approaches are warranted. Jones, Lustig, Farvolden & Powell [1993] investigate the impact of model formulation on the performance of Dantzig-Wolfe decomposition in the context of multifleet assignment problems. This research demonstrated that significant improvements can be realized by using formulations that produce sparse columns in the master problem. Thus, generating a single path in a subproblem is better than generating an entire tree. This effect is particularly pronounced in the context of dynamic networks.

4.6. The dynamic assignment problem

The dynamic assignment problem arises when a set of workers (drivers) must be assigned to a set of tasks over time, responding to new demands as they are called in. The distinguishing characteristic of the dynamic assignment problem over other routing problems is that a worker is never handling more than one task at a time. Different variations of the problem arise in part based on the characteristics of the demands being served. First, tasks may remain at one location, or move from an origin to a destination. Second, we may wish to assign workers to one task at a time, or we may wish to build tours through a series of tasks, typically lasting no longer than a work shift. The tour building requirement tends to arise when tasks are of short duration.

These variations produce the following types of dynamic assignment problems:

1. *The dynamic traveling salesman problem.* A driver must visit a series of points, where the task at each point is relatively short and constant. The problem is to route a driver through a series of points, where demands for a visit are arising dynamically over time.

2. *The dynamic traveling repairman problem.* Tasks occur at one location and are of random length. While it may be possible to develop tours through multiple tasks, often it is necessary to assign workers to one task at a time.

3. *The dynamic driver assignment problem.* This problem arises in truckload trucking, where a driver must be assigned to a load, which has both an origin and a destination. Also, these loads are generally of fairly long duration (one to four days), making it impractical to assign drivers to tours which cover several loads.

4. *The (dynamic) full truckload routing and scheduling problem.* This problem has received very little attention in the literature, but arises when a driver (crew) must be routed over a tour consisting of multiple tasks, and where each task exhibits an origin, destination and service window.

4.6.1. The dynamic traveling salesman problem (DTSP)

Psaraftis [1988] introduces a version of the dynamic travelling salesman problem. Let G be a complete graph of n nodes. Demands for service are independently generated at each node of G according to a Poisson process of parameter λ. These demands are to be serviced by a salesman who takes a (known) time of t_{ij} to travel from node i to node j of G, and spends a (known) time t_0 servicing each demand (on location). If at time $t = 0$ the salesman is at node 1, what should his 'optimal' routing policy be?

'Optimal' here may be defined with respect to a number of different objectives. As in dynamic routing in a communications network [see for example Bertsekas

& Gallager, 1987], there are two main classes of performance measures that are affected by routing decisions: 1) throughput measures (e.g., maximize the average expected number of demands serviced per unit time), and 2) delay measures (e.g., minimize the expected time from the appearance of a demand until its service is completed).

At this point, we are unaware of models for formulating and solving the DTSP explicitly as a dynamic model. Of course, one solution is to find an optimal a priori tour and apply this solution in a dynamic setting. Several other variants of the DTSP can be considered. The graph can be incomplete, symmetric, or Euclidean. The demand generation process can be node specific and may not be Poisson; the service time, t_0, can be a random variable, etc.

A different formulation of the DTSP is the *time-dependent traveling salesman problem*. In this instance, the data is static, but travel times vary as a function of time. Picard & Queyranne [1978] first consider a version of the time-dependent TSP motivated by a machine scheduling problem. Fox, Gavis & Graves [1980] introduce a formulation where travel times between cities are all one period, but where the cost of travel is a function of time. More recently, Malandraki [1989] and Malandraki & Dial [1992b] introduces and develops heuristic solution techniques for the TSP where travel times between cities are a function of time.

4.6.2. The dynamic repairman problem — the uncapacitated case

As a canonical example of a logistics application of the dynamic stochastic vehicle routing problem (DSVRP), consider the following utility repair problem: A utility company (electric power, gas, water, etc) maintains a large, geographically-dispersed network of facilities. The network is subject to failures which occur randomly with respect to both time and location. The company operates a fleet of repair vehicles which are dispatched from a depot in response to reports of local failures. Routing decisions are made on the basis of a real-time log of currently pending failures and possibly some characterization of the process of future failures. Repair vehicles and crews spend a random amount of time servicing each failure before they are free to move on to the next 'call' to be serviced. The utility company wishes to operate its fleet of repair vehicles in a way that minimizes the average downtime due to failures.

The one DSRP that has been investigated in depth to date is the Dynamic Traveling Repairman Problem (DTRP), a Euclidean model of a dynamic VRP. Demands for service arrive according to a renewal process with intensity λ to a connected, bounded Euclidean service region $\mathcal{A}$ of area A. Upon arrival, demands assume an independent and identically distributed (i.i.d.) location in $\mathcal{A}$ according to a continuous probability density function $f(x)$ defined over $\mathcal{A}$. Demands are serviced by m identical vehicles that travel at constant speed v. At each location, vehicles spend some time s in on-site service that is i.i.d. with finite first and second moments denoted by $\overline{s}$ and $\overline{s^2}$ respectively.

A policy for routing the vehicles is called stable if the number of unserved demands in the system is bounded almost surely for all times t. Let M denote the set of stable policies. If a policy is stable, $\rho \equiv (\lambda\bar{s})/m$ is the fraction of time vehicles spend in on-site service. We write T_μ to indicate the system time of a particular policy $\mu \in M$. The DTRP is then defined as the problem of minimizing T_μ. We let T^* denote the optimal value of system time.

Bertsimas and van Ryzin [1991] have analyzed the DTRP for the case in which arrivals are Poisson, demands are uniformly distributed and the entire region is served by a single uncapacitated vehicle.

In the case of light traffic it can be shown that a policy based on locating the server at the median, $x*$, of $\mathcal{A}$ and serving demands in a first-come, first-served (FCFS) order, returning to the median after each service is optimal. This result provides an elegant connection between a dynamic stochastic routing problem (this version of the DTRP) and a static counterpart, the PTSLP (Section 2.4): the optimal location, $x*$, for the static problem is also the location where the server should be located in the dynamic problem and the server-operating policy under the PTSLP is also the optimal operating policy for the DTRP when the traffic is light! The optimal expected system time, $T*$, in this case satisfies

$$T^* \to \frac{E\left[\|X - x^*\|\right]}{v} + \bar{s} \text{ as } \lambda \to 0. \tag{76}$$

In the case of heavy traffic, for this single, uncapacitated vehicle problem, policies exist that have finite system times T_μ for all $\rho < 1$. (Recall that ρ is the fraction of time the vehicle spends in on-site service.) This is surprising in that the condition is independent of the service region size and shape; it is also the mildest stability restriction one could hope for. It can be shown that there exists a constant γ such that

$$T^* \geq \gamma^2 \frac{\lambda A}{v^2(1-\rho)^2} - \frac{\bar{s}(1-2\rho)}{2\rho}. \tag{77}$$

Note that this bound grows like $(1-\rho)^{-2}$ as $\rho \to 1$. Thus, though the stability condition is similar to that of a traditional queue, the system time increases much more rapidly as congestion increases. [In Bertsimas & van Ryzin [1993b] the value $\gamma = 2/(3\sqrt{\pi}) \approx 0.376$ is derived.]

Several interesting operating policies μ that have finite system times, T_μ, for all $\rho < 1$ are investigated in Bertsimas and van Ryzin [1991].

(i) *The partition policy*. This is a policy that applies to the case where $\mathcal{A}$ is a square. It consists of dividing the region into n equal subregions which are served sequentially such that each subregion is adjacent to the next subregion in the sequence except, possibly, for the last one. Within each subregion, demands are serviced in FCFS order until no more demands are left. Then the vehicle moves on to the next subregion in the sequence. The number of subregions n must be optimized in each problem instance.

(ii) *The traveling salesman policy*. As demands arrive arrange them into sets of size n. When all n demands have arrived, consider this the 'arrival of a set'. Service sets in FCFS order by forming a TSP tour on the set of demands. The size of the TSP sets n must be optimized in each problem instance.

(iii) *The space filling curve policy.* Visit demands in the order in which they are encountered in continuous sweeps of the space-filling curve (applies to the case where $\mathcal{A}$ is a square).

(iv) *The nearest neighbor policy.* After each service completion, serve next the demand that is closest to the current location of the vehicle.

It is remarkable that all these diverse operating policies have the same type of asymptotic behavior as the lower bound noted in (77) above, namely, as , $\rho \to 1$,

$$T^* \sim \gamma_\mu^2 \frac{\lambda A}{v^2(1-\rho)^2}, \tag{78}$$

where the constant γ_μ depends on the policy μ. Hence, by comparing this bound to the lower bound in (77), it can be seen that the ratio $\frac{T_\mu}{T^*}$ is bounded by a constant as $\rho \to 1$. Thus all the policies provide a constant factor guarantee in heavy traffic.

The provably best policy of the four listed above is the Traveling Salesman (TS) policy. For a version of this policy, it can be shown that $\gamma_\mu = \beta_{\text{TSP}}/\sqrt{2} \approx 0.51$, where β_{TSP} is the Euclidean TSP constant. Relative to the value noted earlier, this gives a best provable guarantee of

$$\lim_{\rho \to 1} \frac{T_\mu}{T^*} \leq \frac{\beta_{\text{TSP}}^2}{2\gamma^2} \approx 1.8. \tag{79}$$

In fact, Bertsimas and van Ryzin [1991] conjecture that the TS policy is optimal in heavy traffic and that the factor of 1.8 is due to slack in the lower bound (77). If true , this conjecture implies, once again a strong connection between a DSRP and a static counterpart, in this case the PTSP on the Euclidean plane.

The TS policy for the DTRP can be extended to a mixed objective involving both waiting time and travel cost. By increasing the size n of the sets that are formed, travel distance per demand can be reduced at the expense of increasing the mean system time. Indeed, one can show that to minimize system time, one should essentially maximize the amount of travel per demand served. Thus, travel cost and system time are conflicting objectives that can be balanced by sizing routes in an appropriate way.

4.6.3. The dynamic driver assignment problem

The driver assignment problem arises in applications where at any point in time, the problem is to assign a driver to a single task. This problem arises in truckload trucking, where a load might take one or more days to complete. As a result, dispatchers assign a driver to a load, and wait for the load to be delivered before planning the next assignment.

In its simplest terms, the driver assignment problem can be formulated as a simple network assignment problem, with arcs connecting driver nodes to load nodes, as illustrated in Figure 10. The cost of the arc from a driver node to the load node would represent the cost of moving empty from the driver's current location to the location of the load. In commercial applications, models such

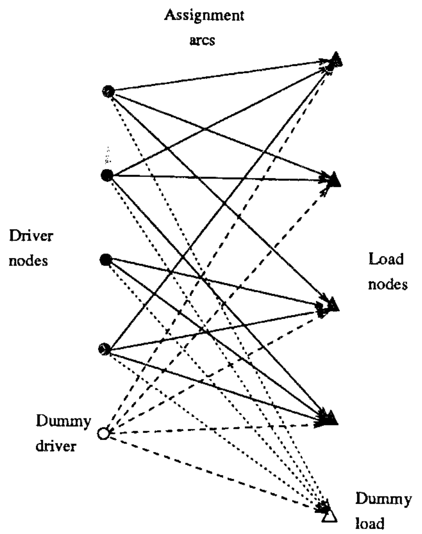


Fig. 10. Static driver assignment network.

as these are run in real-time, responding to changes about the set of drivers and loads. This is especially valuable for carriers which use satellite tracking and mobile communication units to obtain real-time information about the status of each driver.

The model has two attractive features: First, it is extremely simple and can be easily optimized using well-established network optimization codes. Given an optimal solution, the network can be reoptimized following a change in the data in a fraction of the data. Second, the model is extremely flexible, since other, nonquantifiable issues, can be handled by adding various bonuses and penalties to the assignment arcs. For example, one issue that arises with some frequency in the truckload industry is the problem of getting drivers home. Loads go from one city to the next, and there is no guarantee that a driver sent, say, from his home in Atlanta to Chicago will then be sent back home to Atlanta. His next load might be to Dallas. After about two weeks, the carrier might want to try to get the driver home. This can be accomplished in a heuristic fashion by adding a bonus on arcs out of a particular driver to loads which return to points near his home. The size of the bonus can be chosen by asking management how many additional miles they are willing to run a driver to get him home. Needless to say, it is very easy to design a rule-based system for generating these bonuses, taking into account a variety of issues.

One limitation of the basic assignment model is that it is unable to account for the future impacts of decisions made now. For example, if there are too many drivers in one part of the country, it is unable to recommend that drivers be repositioned empty to another region. Also, if too many loads have been booked, it cannot determine the most profitable loads to accept, and which should be 'rejected.' Powell [1991] presents such a model, that combines the basic assignment model with a stochastic, dynamic network model that incorporates forecasted loads. The resulting model is a pure network that can be solved in real-time.

4.7. Dynamic vehicle routing problems

Dynamic vehicle routing arises when we must route vehicles to serve a series of customers with data that is changing over time. The problem differs from the dynamic assignment problem in that multiple customer orders can be consolidated onto the same vehicle, subject to vehicle capacity constraints. Thus, we add the element of clustering orders into tours which satisfy vehicle capacity constraints.

Our discussion of dynamic vehicle routing considers three classes of applications.

The first, introduced in Section 4.7.1 is the capacitated version of the dynamic traveling repairman problem, considered earlier. The second, described in Section 4.7.2, comprises deterministic dynamic vehicle routing problems which exhibit *time-dependent* data, which is all known in advance. Thus we may know a week's worth of demands, and we need to develop a route and schedule for each vehicle that serves these demands over the course of the week. Third, Section 4.7.3 summarizes the literature on stochastic vehicle routing. This literature focuses primarily on a priori (or two-stage) models, but these can form the basis for a stochastic, dynamic *model* that can be used to solve dynamic *problems*. Finally, Section 4.7.4 describes specific applications of vehicle routing models in a dynamic setting, where models were solved in real time. The section on stochastic vehicle routing only discusses the literature. Later, Section 8.1, presents specific models in greater detail. The reason for this delay is that we want to present some of the key concepts in stochastic programming in Section 7.

4.7.1. The dynamic traveling repairman problem - capacitated case

We first visited the uncapacitated dynamic traveling repairman problem as a type of dynamic assignment problem. The capacitated version of this problem is, in effect, the dynamic vehicle routing problem, where customers must be clustered into tours which satisfy vehicle capacity constraints.

Bertsimas and van Ryzin [1993a] examine the case where the region $\mathcal{A}$ is now served by a homogeneous fleet of m vehicles operating out of m depots (whose locations need not be distinct, and where each vehicle is restricted to visiting at most q customers before returning to the depot. A lower bound for the minimum expected system time analogous to that in (77) is obtained. It is also shown that when the vehicle capacity q is very large, policies with the same constant factor performance guarantees as in the single-server case can be constructed by simply partitioning $\mathcal{A}$ into m equal subregions and serving each one independently using a single-server policy. For q finite, constant factor guarantee policies can still be constructed. The best of these policies is based on modifying the Traveling Salesman policy (cf. above) using the tour partitioning heuristic of Haimovich, Rinnooy Kan & Stougie [1988]. Analytical expressions are given for the constant factor performance guarantees and for stability conditions under this policy. As might be expected, when the vehicle capacity q is finite, stability is no longer independent of the geometry of the service region, unlike the uncapacitated vehicle case.

All the results reviewed so far apply to the case of uniformly distributed demand and Poisson arrivals. In Bertsimas and van Ryzin [1993b] most of these results are extended to the case where demands are distributed in $\mathcal{A}$ according to a general continuous density $f(x)$ and arrivals occur according to a general renewal process. These generalizations require quite different analytical approaches and proof techniques. An additional interesting extension is that results are obtained for operating policies which are 'spatially fair' (i.e., provide the same mean system time for all locations within $\mathcal{A}$) as well as for policies which are 'spatially discriminatory' (i.e., provide different mean system time s to different regions of $\mathcal{A}$).

The DTRP and its variations is clearly a rich problem which, in a dynamic context can play a role analogous to that of the TSP — for deterministic routing problems — or of the PTSP — for stochastic but static problems. It is interesting that the analysis of the DTRP yields simple expressions for the system time that provide insights into the effects of: traffic intensity; on-site service characteristics; number, speed and capacity of vehicles employed; service region size; the distribution of customer locations; and fairness of service constraints.

A recurring observation in the work of Bertsimas and van Ryzin is that static vehicle routing methods, when properly adapted, can yield optimal or near-optimal policies for DSRPs. This is an encouraging result on two levels. On a theoretical level, it suggests that there is indeed a connection between the properties of the static and the dynamic problems. On a practical level, the implication is that exact algorithms, heuristics and insights that have been developed over the years regarding static VRPs and SRPs are relevant to DSRPs and, in fact, form the basis for effective analysis and implementation of DSRP systems.

4.7.2. Deterministic dynamic vehicle routing models

In this section, we consider dynamic *models* of vehicle routing problems, focusing on a single snap-shot of data. These models can be used in real-time to solve dynamic problems, or to develop vehicle schedules over the course of the day using data that is (assumed) known in advance.

Deterministic models of dynamic vehicle routing problems arise in two settings: time-dependent demands, and time-dependent travel times. Time-dependent demands arise generally when demands must be satisfied subject to specific time constraints. Time-dependent travel times usually arise in models which are trying to capture time-of-day congestion effects.

Time-dependent demands. One case of time-dependent demands arises when demands must be satisfied within a specified time window. However, a richer version of this problem arises in the context of the *inventory routing* problem, where the determination of customer demands is based on the need to maintain customer inventories. While a customer might require a delivery no later than day t, a carrier (for example, a company providing fuel oil) might decide to deliver on day $t' < t$ if a truck will be in the area. The basic tradeoff is between routing and inventory holding costs.

Hausman & Gilmour [1967] provide what is likely the first explicit model of vehicle routing which considers the assignment of demands both to days of the week as well as to vehicle tours. The problem is solved using a local search heuristic. More recently, Russell and Gribbin [1991] describe a multiphase approach which uses a sequence of optimization models to first assign customer demands to days of the week, and then build vehicle tours. Their model assumes that each customer must be served with a specific frequency (perhaps twice a week) and the problem is to determine the days of the week to serve the customer which satisfies this frequency. Local improvement heuristics are also used to improve tours. Dror, Ball & Golden [1985] provide a more comprehensive computational comparison of algorithms for inventory routing problems.

Time-dependent travel times. The second class of problems includes time-dependent travel times. We have already reviewed the work on dynamic traveling salesman problems. The only work that we are aware of that deals explicitly with (capacitated) vehicle routing and time-dependent travel times are Malandraki [1989] and Malandraki & Daskin [1992a]. This formulation assumes that each link has a fixed travel time that is a function of the time a vehicle arrives to the link. This creates discontinuities in the travel times from one time period to the next, but vehicles are allowed to wait at a node, which is shown to have the effect of smoothing the travel time function. The objective function is to minimize the total time required to complete all tasks. The decision variable is defined as:

$$x_{ijm} = \begin{cases} 1 & \text{if any vehicle travels directly from node } i \text{ to node } j \\ & \text{starting from } i \text{ during time interval } m \\ 0 & \text{otherwise} \end{cases} \tag{80}$$

In contrast with the classical formulation of the vehicle routing problem, their formulation includes an index that identifies the time interval the vehicle enters the link. Within this time period, the travel time is assumed constant. The intent is to capture major changes in travel times between, for example, peak and off-peak travel. A nearest-neighbor tour construction heuristic is proposed, as is a linear programming-based cutting plane algorithm. Testing is performed on problems with 10 to 25 nodes.

4.7.3. *Stochastic vehicle routing models*

Stochastic vehicle routing models generally consider the problem of designing vehicle routes with random (forecasted) demands, although there is limited research on the case of stochastic travel times [see Laporte, Louveau & Mercure, 1992]. Routing subject to stochastic demands can be divided into three cases: 1) finite horizon (e.g. routing over a single work shift), 2) infinite/rolling horizon, and 3) periodic. Most vehicle routing problems have a well-defined beginning and end, covering one day's work, where all drivers are expected to return to the home depot at the end of the shift. Rolling horizon problems arise in some situations where drivers are not expected to return home each day (in some cases, drivers may not return home for many days), giving the problem an infinite horizon flavor [see Bell, Dalberto, Fisher, Greenfield, Jaikumar, Kedia, Mack & Prutzman, 1983]. Finally, periodic models typically involve the problem of developing schedules over the course of the week. This includes inventory routing problems, described earlier (see Trudeau & Dror [1992] for a discussion of stochastic inventory routing problems).

The problem of routing vehicles in the face of uncertain demands is an old problem, and was probably one of the motivating applications behind the field of stochastic programming. However, the earliest references of routing vehicles under stochastic demands fall in our presentation under the category of dynamic vehicle allocation problems. The early literature on vehicle routing with stochastic demands considers two-stage problems, where a first-stage decision, such as setting a fleet size or locating a warehouse, is followed in the second stage by the actual routing of the vehicle. In this context, it is necessary to obtain expected routing costs, in a probabilistic sense, given the first stage decision. Some of the first papers which touched on this include Golden & Yee [1979] and Dempster, Fisher, Jansen, Lageweg, Lenstra & Rinnooy Kan [1981], which suggested that these problems could be formulated as multistage, stochastic integer programs. Following on this idea, Spaccamela, Rinnooy Kan & Stougie [1984] show how results from probabilistic analysis of traveling salesman problems [Beardwood, Halton & Hammersley, 1959] can be used to approximate the second stage problem for vehicle routing. The goal of this research, however, is not so much to design the vehicle tour but rather to improve decisions in other areas which depend on routing costs.

The earliest explicit treatment of stochastic demands in the design of vehicle routes appears to be Tillman [1969] who presented a modification of the Clarke-Wright savings algorithm for terminals with Poisson-distributed demands. The first in-depth treatment of this problem is Stewart & Golden [1983] (the references in this paper cite earlier papers by the same authors that appeared in conference proceedings), who formulate the stochastic vehicle routing problem using both a chance-constrained and 'penalty function' approach. The penalty function approach is a form of stochastic programming with *simple recourse*, and subsequent papers differ primarily in the type of simple recourse strategy used. An excellent review of this research is provided in Dror, Laporte & Trudeau [1989]. Dror & Trudeau [1986] present an alternative recourse model which they show to improve on the original models presented in Stewart & Golden [1983].

4.7.4. *Dynamic applications in vehicle routing*

Interestingly, the limited progress that has been made in developing explicit models of dynamic vehicle routing problems that combine stochastic and dynamic elements has not stopped the implementation of practical tools for these problems. Brown & Graves [1981] provide one of the earliest descriptions of an application of an on-line optimization tool for real-time dispatching of petroleum tank trucks. Also at the same time, Gavish [1981] describes an on-line logistics planning system covering a) selection of leased vehicles, b) assignment of demands to depots, c) choice of return depot following the completion of each trip and d) workload allocation among depots. A series of heuristics were developed to solve these problems, requiring 5 to 10 minutes on a large mainframe of that era.

Bell, Dalberto, Fisher, Greenfield, Jaikumar, Kedia, Mack & Prutzman [1983] describe a similar application of a set partitioning model for routing trucks for Air Products and Chemicals. This model used a fixed planning horizon, using deterministic approximations of forecasted demands. Lagrangian relaxation was used to provide an optimal integer solution to the set partitioning problem. Other, unpublished, examples of real-time uses of optimization for routing and scheduling can also be found. However, the long-term success of these approaches is quite low, and we are unaware of any applications of these advanced techniques in a real-time context which are still operational three years after installation (since these projects typically do not make their way into the research literature, it is quite possible that long term, successful applications have occurred).

An open question in real-time applications is whether advanced optimization methods are really warranted. Bagchi & Nag [1991] describe a completely heuristic, expert systems-based approach for dynamic vehicle scheduling. The Gavish system [Gavish, 1981] uses optimization-based heuristics, and is basically similar in style. As these systems mature, it is expected that a hybrid of optimization models and rule-based expert systems will emerge.

4.8. Dynamic service network design problem

The dynamic service network design problem addresses the task of determining an optimal movement of vehicles, denoted by the vector v_t, which represent transportation services. The function $F_w(v)$ in equation (64) captures the optimal costs of transporting shipments given v. The service network design problem consists of determining the flow of vehicles v_{ijt} moving between two terminals (cities) in time t (arriving at node j in time period $t + \tau_{ij}$). We view v_{ijt} as the set of scheduled transportation services which will move goods. Since vehicles may have to be repositioned empty, we let:

u_{ijt} = number of vehicles moving empty from city i to city j, originating in period t.

V_{it} = net surplus or deficit of vehicles at city i in time period t.

The service network design problem can now be stated as:

$$\min_{v_t, u_t} F_w(v) + \sum_{t=0}^{P} c^T (v_t + u_t) \tag{81}$$

subject to, for $t = 0, \ldots, P$

$$\sum_{j \in C} (v_{ijt} + u_{ijt}) - \sum_{k \in C} \left[v_{ki,(t-\tau_{ki})} + u_{ki,(t-\tau_{ki})} \right] = V_{it} \qquad i \in C \tag{82}$$

$$v_t, u_t \geq 0 \tag{83}$$

$$v_t, u_t = 0 \qquad t < 0 \tag{84}$$

Rather than write the problem as a single, large optimization problem which combines traffic assignment along with the routing of loaded and empty vehicles, we have captured the traffic assignment component in the function $F_w(v)$.

Relatively little attention has been given to the dynamic service network design problem, with most prior work focusing on static formulations. Crainic, Ferland & Rousseau [1984] and Crainic, Gendreau & Dejax [1992] propose nonlinear network models for the service network design problem for trucking and rail. Local improvement heuristics are used to search for optimal integer frequencies. Powell & Sheffi [1989] propose an interactive optimization system for less-than-truckload trucking which optimizes the decision of whether to offer transportation service between two points (the frequency of service is handled implicitly). Haghani [1989] proposes a dynamic model of the service network design problem for rail, similar in style to (81). The model is solved using a linear programming package on some small networks. Farvolden & Powell [1991b] define the dynamic service network design problem for less-than-truckload motor carriers, and suggest a subgradient-based local improvement heuristic for deciding where to offer transportation services. The subgradient captures the effect of a change in the vector v_t on the flows of shipments w as well as empty vehicles u_t.

4.9. *Dynamic facility planning*

Our last topic in algorithms for dynamic networks is that of dynamic facility planning. This encompasses strategic planning of optimal dynamic capacity expansion for networks, and the tactical problem of dynamically repositioning servers and facilities in an operational setting. Examples of problems include adding or dropping plants, warehouses and terminals from a logistics network, serving new markets, or adding link capacity (a problem that arises particularly in communication networks). A number of papers have been written on this topic since the early paper by Erlenkotter [1969]. Reviews of solution approaches are given in Fong [1974], Erlenkotter [1981], Minoux [1987] and Christofides and Brooker [1967]. Zadeh [1974] considers optimal expansion of communication networks, and Fong & Srinivisan [1976] address the combined problem of capacity expansion and shipment planning. Algorithms for multiregion capacity expansion problems are given in Erlenkotter [1975], Fong & Srinivasan [1981a, b, 1986]. Doulliez & Rao [1975] formulate the problem as a specialized shortest path problem, and Laporte & Dejax [1989] show how a combined dynamic location and routing problem can be solved as a vehicle routing problem over an expanded graph. Van Roy & Erlenkotter [1980] suggest a dual-based procedure. Berman [1981] and Berman &

Leblanc [1984] introduce the problem of dynamically repositioning mobile service units over a network. Bean & Smith [1985] address the issue of capacity expansion over an infinite horizon, and Higle, Bean & Smith [1984] consider the problem of capacity expansion under stochastic demands.

5. Algorithms and formulations for deterministic dynamic networks

The special structure of dynamic networks would seem to encourage the development of specialized algorithms. Section 5.1 provides a general introduction to dynamic networks including a historical perspective and some general terminology. Section 5.2 describes a class of inductive algorithms that have been developed for the efficient solution of dynamic networks. Next, Section 5.3 presents a specialization of Dantzig-Wolfe decomposition in a dynamic context. Section 5.4 describes a variable transformation that substantially accelerates the solution of nonlinear dynamic networks. Section 5.5 then raises issues that arise in the solution of dynamic multicommodity network flow problems. Section 5.6 briefly discusses results for dynamic concave cost networks that arise in production planning, and Section 5.7 reviews the literature on optimal control methods for dynamic networks. Finally, Section 5.1.4 notes special measures that can be taken in the generation of dynamic networks.

5.1. An introduction to dynamic networks

Dynamic network models have proved to be an effective modeling framework for a range of planning problems that arise in logistics. These include production and inventory planning, which determines when and where goods should be shipped, as well as the operational problems faced by private fleets and common carriers which must provide the transportation services. Common carrier applications include truckload and less-than-truckload (LTL) trucking, railroads, airlines, and international container operations. These models typically track four types of activities - the movements of vehicles (locomotives, tractors, ships), the scheduling of drivers and crews that guide them, the management of fleets of containers (railcars, trailers, ocean containers), and the flows of goods (shipments/passengers) moving in the containers.

5.1.1. Historical perspective

The use of dynamic networks in an optimization framework was well established in the 1950's. Dantzig & Fulkerson [1954] formulates a tanker scheduling problem using dynamic networks. Ford & Fulkerson [1956] and Gale [1959] derive results for maximum flows over dynamic networks. One of the earliest comprehensive discussions of the use of dynamic networks for common carrier applications is provided by Magnanti & Simpson [1978], which focuses on airline problems. Vemuganti, Oblak & Aggarwal [1989] provides a more recent survey of network models for fleet management, focusing on fleet sizing decisions.

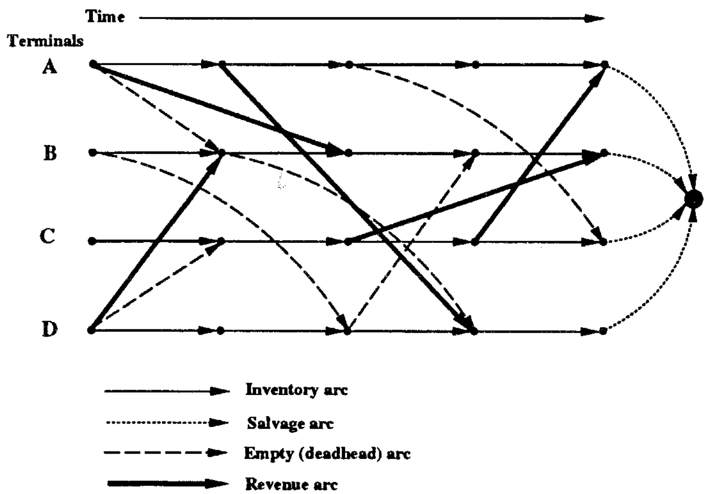


Fig. 11. Dynamic space-time network for airline fleet assignment.

As an illustration, Figure 11 shows a space-time network describing a set of flights over the course of the day. The flows over the network are in units of aircraft. *Revenue* arcs represent (potential) flights moving from one airport at one point in time to another airport at a future point in time. Revenue arcs generally carry an upper bound of one. Other arcs are *empty repositioning* movements and *inventory* arcs (referred to in the airline literature as ground holding arcs). Finally, Figure 11 uses a supersource and supersink, requiring *source* arcs, which carry aircraft from the supersource to the locations of the aircraft at the beginning of the day. Salvage arcs carry aircraft into the supersink at the end of the day. The number of aircraft that are required at each airport at the beginning of the day is an output, rather than an input. However, the network in Figure 11 does not guarantee that the number of aircraft at each airport at the beginning of the day is the same as at the end of the day. For this reason, we need to impose an *equilibrium* condition that specifies that the number of aircraft at each location is the same at the beginning and the end of the day. This equilibrium condition can be posed as a side constraint, or by introducing arcs that loop from the last time period for each city to the first time period for the same city.

5.1.2. Types of dynamic networks

Dynamic networks can be described as variations of two broad flavors. In a *fully dynamic* network, every link moves forward in time from t to some time $t + \tau$, $\tau > 0$. A special case of a fully dynamic network occurs when $\tau = 1$ for all links; we refer to these as *staged networks*. Later, when we deal with dynamic networks under uncertainty, we will have to split time periods into stages to capture the sequence of making decisions, and then observing realizations of random variables. For these problems, it is often convenient to transform networks with links that span more than one time period into networks where all links move forward exactly one time period.

At the other extreme are networks that are *dynamic inventory networks*, which are dynamic sequences of static problems (such as that illustrated in Figure 9). The only dynamic arcs (those moving forward in time) are inventory or holding arcs. Operational problems tend to look more like fully dynamic networks (see Section 4.5), while production planning problems are often modelled using a large time step, where all activities take place within a time period with the exception of inventory holding (illustrated in Section 4.4.2). Hybrid networks arise when a time step is chosen such that some activities move forward in time, while others move within the same time period. Regardless of the time step, however, it is very important that the nodes of the network be defined to ensure that the network is acyclic. For example, links within a time period may always move from plants to customers. If links can move between warehouses within a time period (that is, a link can go from warehouse i to warehouse j and vice versa) then it is important to introduce devices such as splitting each warehouse into inbound and outbound nodes, so that a link from warehouse i to warehouse j actually goes from the inbound node of i to the outbound node of j.

Dynamic networks can also be distinguished by how activities change over time. *Transient networks* exhibit data that, at least in the initial time periods, exhibits no particular pattern, generally reflecting the initial conditions of the problem. On the other extreme are *stationary networks* which exhibit data that is the same from one time period to the next. Finally, *periodic networks* are characterized by data that varies over a fixed cycle (typically a day or a week), which is assumed to be repeated from one cycle to the next.

5.1.3. Graphical representations

One of the most valuable aspects of network models is the ease with which they can be displayed graphically. In contrast with many static networks, dynamic networks pose the problem of dealing with a fourth dimension (time) that produces several strategies for displaying models, with no single strategy appropriate or even preferred for all situations. Four methods that are commonly used include:

1. *Two dimensional figures*. This approach is widely used for displaying small networks. Space is represented in one dimension, and time in the second dimension. Depending on the author, these networks may be displayed with time proceeding left to right, or from top to bottom (occasionally from bottom to top). It is useful for illustrative purposes in papers, but not very practical for viewing real networks.

2. *Three dimensional displays*. Most commonly, this approach consists of a series of two-dimensional maps, stacked from bottom to top to represent time periods. Arcs can move from a point on a map in one time period to a point in different map representing a later time period. On a computer terminal, it is possible to develop true three-dimensional representations of networks which can be manipulated by the user to rotate from one view to the next. These systems

are not widely available, but will come increasingly so as powerful graphics workstations and microcomputers come into wider use.

3. *Simulation.* For practical problems, two or three-dimensional displays of networks can quickly become too cluttered to be of any value. An alternative, then, is to display a dynamic network by literally showing a simulation of activities through space time.

4. *Task schedules.* In this category, we include graphical displays of schedules, such as Gantt charts, which show timing without the spatial movement. These might be used in conjunction with a static (spatial) display of movements.

The goal of graphical displays of dynamic networks is to capture the interaction of activities over space and time. To date, no single approach has emerged as a dominant technique for accomplishing this goal.

5.1.4. Network generators and data handling

A final comment should be made on the structure of network generators for dynamic networks. It is common in dynamic settings for a considerable amount of the data to be essentially static. For example, the distance between two cities does not change over time (although the time to get between two points can change over time, see Malandraki [1989] and Malandraki & Daskin [1992a]). Conventional models do not take advantage of this property, choosing instead to replicate data over many time periods. Klingman & Mote [1982] describe a specialization of a primal network simplex code for a multiperiod production, distribution and inventory planning problem. The code requires that all the parameters of the problem (network structure, costs and bounds) be fixed over time. Using this property, the code is structured to store all this data only once. In addition, each arc is priced over all time periods, with the most favorable arc being used in pivot operations. We are not aware of any work that takes advantage of partial stationarity of data over time, where, for example, demands may vary over time but other activities do not.

5.2. Inductive algorithms for linear networks

The temporal structure of a dynamic network has encouraged the development of specialized algorithms which take advantage of the properties of dynamic networks. White & Bomberault [1969] and White [1972] appears to be the first efforts to design an algorithm explicitly for dynamic networks. The papers outline an inductive procedure which begins with a network with a single node, and successively adds nodes to solve larger problems. Nodes are added to the network in time order. As each node is added, the algorithm looks to push flow from supply nodes to satisfy a demand node.

The methods reviewed above exploit only the dynamic structure of the problem to design specialized algorithms. A different approach exploits *properties* of the underlying *model* to develop efficient algorithms. The most notable example of this is the forward network simplex developed by Aronson & Chen [1986] for multiperiod distribution and inventory problems, for networks with the structure in Figure 9. Their algorithm works by solving a truncated problem over periods $t = 1, \ldots, T - 1$, and then using the solution as an advanced start for period T. The implementation of the network simplex algorithm exploits the fact that when the horizon is expanded to period T, flows in earlier periods may not be affected. Thus the only arcs that are priced out are those in period T, and those in periods $t^o, \ldots, T - 1$, where t^o is the earliest time period affected during the pivoting process following an augmentation. As T increases, t^o tends to increase as well, producing substantial reductions in the number of arcs that are priced. Stanley [1987] introduces a forward convex simplex algorithm for nonlinear dynamic networks.

The performance of the forward network simplex algorithm is due in large part to the properties of the underlying model. As time increases, breaks in the basis tend to occur, connected only by artificial links through the root node. These breaks represent *regeneration* points in the solution, which arise naturally in production and inventory planning problems. The result of these breaks is that pivots in one block of the network do not have an impact on other blocks. As the algorithm progresses forward in time, blocks of the network in early time periods become separate by breaks from later parts of the network. Realizing this, Aronson and Chen modified the simplex algorithm to detect the earliest time period affected by a pivot.

The datasets in Aronson & Chen [1986] use randomly generated supplies and demands in each period which are stationary over time. The model also includes extra 'outsourcing' supplies that allows demands to be satisfied in any period from an outside supplier, but at a higher cost [Aronson, 1990]. The model effectively trades off between inventory holding costs and the outsourcing cost to determine how many periods product should be held in inventory. This tradeoff tends to create frequent periods when no product is held in inventory. The result of these zero inventory periods are breaks in the basis, where pivots in one time period cannot affect decisions in another period with an intervening break. These *regeneration points* [Morton, 1979] create empirical decision horizons and forecast horizons where decisions in early time periods are unaffected by activities in later time periods [Aronson & Thompson, 1984].

The forward network simplex algorithm uses this property to reduce the number of arcs that need to be priced. Alternatively, if we are only interested in the decisions for the initial time periods, we could stop the algorithm as soon as a decision horizon is detected. In the case of dynamic networks, a gap in the basis is referred to as an empirical decision horizon because they are not guaranteed to remain as the problem grows. The term refers to the fact that these breaks seem to hold with a high probability. Exact decision horizons can be obtained for special types of problems. For example, Wagner & Whitin [1958] and Zangwill [1968, 1969], obtain exact decision/forecast horizons for dynamic networks with concave arc costs, and Aronson & Chen [1985] obtain similar results for a special production planning model.

This line of research falls in the class of *forward algorithms* [Morton, 1981] which exploit the presence of *regeneration points* in the model [Lundin & Morton, 1975;

Morton, 1979]. Interestingly, these regeneration points are almost guaranteed *not* to occur in dynamic fleet management. The balancing of supplies and demands in each period that arises in production and inventory planning, resulting in zero inventory arcs, does not occur in fleet management problems.

5.3. A decomposition approach for linear dynamic networks

A broader class of specialized algorithms can be drawn from the literature on dynamic linear programs with the general form:

$$\min \sum_{t=0}^{P} c^T x_t$$

subject to

$$\begin{aligned} &A_0 x_0 = R_0 \\ &B_{t-1} x_{t-1} + A_t x_t = R_t \\ &x_t \geq 0. \end{aligned}$$

One solution approach, presented by Ho & Manne [1974], uses a creative adaptation of Dantzig-Wolfe decomposition. Their approach involves solving the problem iteratively in *cycles*. Each cycle involves solving a sequence of subproblems SP_t, starting for $t = P, P-1, \ldots, 0$. The unusual aspect of the method is that problem SP_t works simultaneously as a subproblem for periods $t+1, \ldots, P$, and as a restricted master problem (in the terminology of Dantzig-Wolfe decomposition) for periods $0, 1, 2, \ldots, t-1$. Let x_τ^k be the kth extreme point solution for period τ, $\tau = 0, \ldots, t$ and let λ_{jt}^k be the weight given to x_t^j in the kth cycle, where $x_t^k = \sum_{j=1}^{k-1} \lambda_{jt}^k x_t^j$ and:

$$\sum_{j=1}^{k-1} \lambda_{jt}^k = 1$$

Let p_t^k be the value of the kth proposal in period t, defined by:

$$\begin{aligned} p_2^k &= c^T x_1^k \\ p_t^k &= c^T x_{t-1}^k + \sum_{j=1}^{t-1} p_{t-1}^j \lambda_{j,t-1}^k \end{aligned}$$

Now define

$$S_t^k = B_{t-1} x_{t-1}^k$$

In network terms, S_t^k is the supply of flow in the kth iteration into period t produced by flows in period $t-1$. Subproblem $SP(t)$ is now:

$$(SP_t) \quad \min_{x_t, \lambda_{jt}^k} [\sum_{j=1}^{k-1} p_t^j \lambda_{jt}^k] + [c - \pi_{t+1}^k B_t] x_t^k \tag{85}$$

subject to

$$A_t x_t^k + \sum_{j=1}^{k-1} S_t^j \lambda_{jt}^k = R_t \tag{86}$$

$$\sum_{j=1}^{k-1} \lambda_{jt}^k = 1 \tag{87}$$

$$\lambda_{jt}^k, x_t^k \geq 0 \tag{88}$$

The first set of brackets represents a restricted master problem for periods $t = 0, \ldots, t-1$. The second set of brackets is the relaxed subproblem, where the vector π_{t+1}^k is the dual variables for constraint (71) from subproblem SP_{t+1}. For $t = P, \pi_{t+1}^k = 0$. For each cycle through the time periods, the duals π_t^k are improved and an additional extreme point is added to the set, improving the quality of the solution for periods $t = 0, \ldots, t-1$.

It is unlikely that the Ho and Manne algorithm would ever be directly applied to dynamic networks. It is useful, however, in that it exposes a structure for solving dynamic networks. Problem SP_t is a truncated problem, where the vector π_{t+1} plays the role of a salvage value [Grinold, 1977]. From one perspective, the Ho and Manne procedure can be viewed as a mechanism for finding salvage values in dynamic networks. If the dynamic network were being used on a rolling horizon basis, it may only be necessary for π_{t+1} to be a reasonable approximation, yielding a different interpretation of convergence.

5.4. *Flow splitting algorithms for nonlinear, dynamic networks*

Classical formulations for dynamic networks consider flows in all time periods simultaneously. The standard form for these models can be stated as:

$$\min \sum_{t=0}^{P} c^T x_t$$

$$A_0 x_0 = R_0$$

$$B_{t-1} x_{t-1} + A_t x_t = R_t \quad t = 1, \ldots, P$$

An alternative view of the same problem is to first introduce a *state* variable S_t defined by:

$$S_t = B_{t-1} x_{t-1}$$

which gives the total flow into each node for period t. We can then replace the decision variables x_{ijt} with:

θ_{ijt} = fraction of flow through node i at time t that should be moved over link (i, j).

Clearly

$$x_{ijt} = S_{it}\theta_{ijt}$$

where

$$\sum_{j\in C}\theta_{ijt} = 1 \tag{89}$$

Thus the problem can be reformulated in terms of the state variables S_{it} and the flow splitting variables θ_{ijt}.

This approach to solving dynamic networks involves both the introduction of a state variable S_{it} and a flow splitting variable θ_{ijt}. In this section, the flow splitting variable, with its simple constraint structure (89), plays the central role in the development of specialized algorithms. However, as we consider more difficult problems later in the presentation, it is the introduction and use of the state variable S_{it} that plays an important role in the more important task of developing effective models. For example, Section 9.2 uses a state variable in conjunction with a more sophisticated class of routing strategies than can be accomplished using a simple flow splitting rule.

Let $c_{ij}(x_{ijt})$ be the total cost of x_{ijt} units of flow in period t where we assume $c_{ij}(x)$ is nonlinear, continuously differentiable and convex. For the traffic assignment problem, let

x^r_{ijt} = flow on link (i, j) at time t headed for destination r.

$x_{ijt} = \sum_r x^r_{ijt}$.

The nonlinear dynamic traffic assignment problem can be stated as:

$$\min \sum_{t=0}^{P} c_{ij}(x_{ijt}) \tag{90}$$

such that

$$\sum_{j\in C}\left[x^r_{ijt} - x^r_{ji,(t-\tau_{ji})}\right] = R_{it} \tag{91}$$

$$x^r_{ijt} \geq 0 \tag{92}$$

Gallagher [1978] first used this model to develop a minimum delay routing algorithm for communication networks. The problem with this formulation is that the flow conservation constraints (91) which are so easy to handle using sequential algorithms are not well suited to distributed computation. For this reason, he introduced the flow splitting formulation with

θ^r_{ijt} = fraction of flow through node i at time t with destination r to be routed over link (ij).

In addition he defined the state variable:

$S^r_{it}(\theta)$ = flow through node i at time t with destination r.

where $S^r_{it}(\theta)$ is a nonseparable function of the vector θ. Let

$$x_{ijt} = \sum_r S^r_{it}(\theta)\theta^r_{ij,t}$$

be the total flow on link (i, j), written now as a function of θ.

The nonlinear dynamic assignment problem (90) can now be expressed as follows:

$$\min \sum_{t=0}^{P} c_{ij}(x_{ijt}(\theta)) \tag{93}$$

such that

$$\sum_j \theta^r_{ij,t} = 1 \tag{94}$$

$$\theta^r_{ij,t} \geq 0 \tag{95}$$

In this formulation, the flow conservation constraints (91) are replaced with convexity constraints (94). Problems (90)–(92) are convex, separable (in x_{ijt}) nonlinear programming problems with network constraints. Problems (93)–(95) are convex, *nonseparable*, nonlinear programming problems with constraints that are *separable* by node. For dynamic, acyclic networks, the nonseparable structure of the objective function can be handled easily by defining:

$$C_t(\theta, S_t(\theta)) = \sum_{i,j\in C} c_{ij}(x_{ijt}(\theta)) + C_{t+1}(\theta, S_{t+1}(\theta)) \tag{96}$$

where

$$S^r_{j,t+1} = \sum_{i\in C} S^r_{i,(t+1-\tau_{ij})}(\theta^r_{ij,(t+1-\tau_{ij})})$$

Equation (96) allows us to define a simple backward recursion for the derivatives. Let $F(\theta)$ be the total costs over the network, where

$$F(\theta) = C_0(\theta, S_0(\theta))$$

The derivative of $F(\theta)$ with respect to θ^r_{ijt} is then

$$\begin{aligned}\frac{\partial F(\theta)}{\partial \theta^r_{ijt}} &= \frac{\partial C_t(\theta, S_t(\theta))}{\partial \theta^r_{ijt}} \\ &= \frac{\partial}{\partial \theta^r_{ijt}} \sum_{i,j\in C} c_{ij}(x_{ijt}(\theta)) + \frac{\partial}{\partial \theta^r_{ijt}} C_{t+1}(\theta, S_{t+1}(\theta))\end{aligned} \tag{97}$$

Equation (97) provides a simple backward recursion that is used to calculate two sets of derivatives,

$$\frac{\partial C_t(\theta, S)}{\partial \theta^r_{ijt}}$$

and

$$\frac{\partial C_t(\theta, S)}{\partial S^r_{it}}.$$

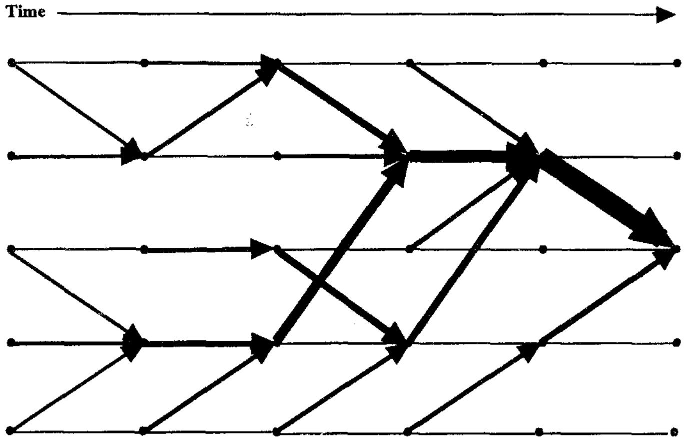


Fig. 12. Funneling in dynamic networks.

This formulation was developed independently by Powell, Sheffi & Thiriez [1984] for a nonlinear formulation of the dynamic vehicle allocation problem. Bertsekas, Gafni & Gallagher [1984] exploited the special structure of the constraint set to develop projection algorithms for a capacitated version of the problem. Bertsekas, Gafni & Gallagher [1984] develop second order algorithms for the same problem. Powell, Berkkam & Lustig [1992] compare the efficiency of the x and θ formulations of the problem, using the specialized algorithms in Genos [Mulvey & Zenios, 1987] to solve the x formulation. These experiments showed that state-of-the-art nonlinear network algorithms could show an almost pathologically poor performance on networks with as few as 10 or 20 time periods. By contrast, even the Frank–Wolfe algorithm worked reasonably well when using the θ formulation, with a gradient projection algorithm performing even better.

The reason for the poor performance of the x formulation, using either Frank–Wolfe or simplicial decomposition, is the nature of an extreme point solution. The solution to a linear subproblem in Frank–Wolfe, which involves routing flows over a shortest path into a supersink for DVA type networks, is illustrated in Figure 12. Following the shortest path, flows from different nodes into the supersink tend to fall into a single path in later time periods, a property we refer to as *funneling*. As a result of this property, which is basic to dynamic networks, extreme point solutions provide especially poor descent directions for first-order search algorithms. By contrast, the θ formulation at every iteration involves solving a subproblem out of every node, thereby providing an adjustment to the allocation of flows out of every single node. The effect is that the solution to the linear subproblem looks more like a dense forest than a narrow path.

5.5. *Algorithms for capacitated dynamic multicommodity flow problems*

Capacitated, multicommodity flow problems over dynamic networks pose a special challenge that arises in dynamic vehicle routing, crew scheduling, and multifleet assignment problems. As a rule, the principal characteristic of these problems is that they are capacitated multicommodity flow problems first, and dynamic problems second. Most of the research focuses on handling the bundling constraints (also referred to as GUB constraints) without focusing on the special properties of dynamic networks [see, for example, Magnanti & Simpson, 1978; Shan, 1985; Chih, 1986; and the survey by Assad, 1987]. Farvolden, Powell & Lustig [1993] develop a primal partitioning method for the capacitated dynamic traffic assignment problem which takes advantage of an empirical property of the dynamic structure. Flows moving over the network which encounter arcs at capacity tend to 'spill' forward in time, a property that produces a near-triangular working basis. This is exploited in the development of an efficient implementation of the simplex method. Ford & Fulkerson [1958b] and Bellmore & Vemuganti [1973] consider maximizing multicommodity flows over dynamic networks. Bellmore & Vemuganti [1973] show how a stationary solution can be expanded into a dynamic solution (over an infinite dynamic network) to obtain a bound on the optimal solution. Tapiero & Soliman [1972] use optimal control theory to solve multicommodity flow problems in continuous time. Gallagher [1978] and Bertsekas [1979] present methods for nonlinear, multicommodity flow problems over dynamic networks, which is reviewed in Section 5.4.

Below, we review some issues that arise that are specifically related to dynamic networks. First, Section 5.5.1 describes how dynamic problems magnify degeneracy in certain formulations. Next, Section 5.5.2 discusses how certain decomposition strategies are producing efficient solution algorithms for multicommodity flow problems defined over dynamic networks.

5.5.1. Degeneracy in dynamic networks

Multicommodity problems over dynamic networks with moderate to long planning horizons tend to exhibit massive degeneracy that is not found in static problems. This property has been especially pronounced in crew scheduling problems for airlines (one problem generated by American Airlines involves over 12 million tours), and where the interaction of tours over time produces the characteristically high level of degeneracy. The problem is illustrated using the network in Figure 13, where a 16 link network has 256 paths joining origin and destination. Assume that all the links have an upper bound of one, and that we are trying to route four separate drivers over the network. This network can be thought of as having four time periods, with four paths per time period joining a set of *regeneration* nodes. A regeneration node represents a point in the network, which

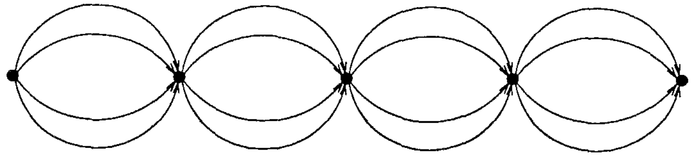


Fig. 13. Degeneracy in dynamic networks.

might often be a driver domicile, where paths might come together. If an n period network has k possible paths in each period, then there will be k^n paths in the network, with only k possible paths with nonzero flow. Furthermore, there are four conservation constraints and 16 upper bound constraints, all of which are binding, implying a basis with rank 20, with 16 basic variables with value zero (assuming an integer solution). Real networks are more complicated, but this simple analysis quickly illustrates the dramatic effect longer planning horizons can have on the potential number of tours.

Researchers are aware of this issue [Desrosiers, 1990], and have developed specialized strategies to help reduce this effect. In airline crew scheduling, one approach that has been used is called graph partitioning. Ball, Bodin & Dial [1983] and Ball & Roberts [1985] use this concept to develop a heuristic tour construction method for crew scheduling in transit and air, respectively. A link in a dynamic network represents a crew handling a flight and is called a duty period. Duty periods are then grouped into pairings, which consists of a small number of duty periods spanning several days, and always beginning and ending at a crew domicile. If each duty period is grouped into a pairing, it is possible to form a new graph, where each node in the network represents a pairing, and links from one pairing to the next represent either a rest period (layover) or empty movement to another domicile. The difficult part of this process is building the pairings, which Ball and Roberts do via a matching-based heuristic. Once this is done, the crew scheduling problem can be solved by finding paths over a greatly reduced graph, where nodes represent pairings. The final part of the procedure then involves performing local switches of duty periods between pairings. The effect of this procedure is to reduce the complexity of the graph, which reduces the number of possible tours and the level of degeneracy.

5.5.2. Decomposition methods

Since dynamic networks exhibit massive numbers of potential paths, it is natural to experiment with decomposition methods based on column generation. It has been conventional wisdom that decomposition methods should be used only when the problem is too large to be solved in-core [see, for example, Ho & Loute, 1983]. However, recent experimental evidence is starting to change this view [Desrosiers, Soumis & Desrochers, 1984; Jones, Lustig, Farvolden & Powell, 1993]. Farvolden, Powell & Lustig [1993] propose a path-based, column generation procedure for solving capacitated, multicommodity traffic assignment problems that arise in freight transportation. Their approach requires solving shortest path subproblems to generate columns, and an LP master problem to determine the best set of paths to use in a current solution. Since every origin-destination pair represents a different commodity, there are a potentially large number of convexity constraints in their master problem, creating a very large linear program. They propose a primal partitioning algorithm that isolates a relatively small working basis. The approach is similar to other algorithms proposed for GUB problems (Lasdon & Terjung [1971] used a similar method for capacitated lot-sizing problems). The master problem has constraints for upper bounds on links, as well as flow conservation constraints over the paths. The standard approach keeps these two types of constraints clustered together in the master problem (that is, all the flow conservation constraints are kept in one set of rows, while the binding upper bound constraints are kept in a different set of rows). In contrast, Farvolden, Powell & Lustig [1993] mixed both types of constraints within the working basis. For dynamic problems, the resulting working basis, with a proper ordering of rows and columns, exhibited a *near triangular* structure which both accelerated the algorithm and increased the degree of integrality in the optimal solution. This near triangular characteristic is an empirical observation, and seems to arise only with dynamic networks.

Jones, Lustig, Farvolden & Powell [1993] extends this idea to explore the effect of problem formulation on the performance of Dantzig-Wolfe decomposition for capacitated multicommodity flow problems. For traffic assignment problems, which involves routing flow from M origins to N destinations, it is possible to formulate the subproblem using N trees, one for each destination, or $M \times N$ paths. Using trees, the master problem involves N convexity constraints, whereas paths requires $M \times N$ convexity constraints. Using modern linear programming technology, Jones, Lustig, Farvolden & Powell [1993] show that working with paths can significantly improve algorithmic efficiency (sometimes by several orders of magnitude). While this work is not specifically tied to dynamic networks, the value of using paths over trees seems to arise only in the context of dynamic networks where degeneracy is much more pronounced. Jones [1992] further experiments with a dynamic decomposition scheme which involves path splitting over networks with long planning horizons (e.g. greater than 10 time periods); instead of using a single path from source to sink over the entire planning horizon, paths are split into two segments, joined in the middle of the planning horizon. Four paths in each segment, joined at the middle, can have the effect of 16 paths over the entire planning horizon. Once again, it is the dynamic structure of the problem that is creating the difficulty.

5.6. *Dynamic networks with concave costs*

Dynamic networks with concave costs arise frequently in production and inventory planning to capture economies of production and shipping. Wagner & Whitin [1958] formulated the basic production planning problem as a dynamic network with concave costs, and presented a dynamic programming algorithm for its solution. They also give one of the earliest planning horizon results, which has been exploited in the formulation of models as well as the streamlining of algorithms. Zangwill [1968] shows that optimal flows in dynamic networks with concave costs exhibit a tree structure (that is, no cycles), which can be exploited to accelerate dynamic programming algorithms. This result is then used [Zangwill, 1969] to extend Wagner-Whitin's main result to multistage production networks [see also Veinott, 1969; Graves & Orlin, 1985; and Erickson, Monma & Veinott, 1987]. For a thorough mathematical treatment of production planning, see Bensoussan, Crouhy & Proth [1983].

The central result of research on dynamic networks with concave costs is the tree structure of an optimal solution. The implication is that any movement of flow, whether it be production at a plant, shipping from plant to warehouse, or shipping warehouse to customer, is always in an amount that will exactly satisfy the demands for a particular set of time periods. As a result, the problem can be reformulated not in terms of how much flow should be moved, but rather which set of demands should be satisfied. Thus, when we move flow from plant to warehouse in time period 5, it may be to satisfy customer demands in time periods 8, 9 and 10. We can then formulate this problem in terms of deciding which time periods should be satisfied by a particular set of flows. The resulting problem can then be solved in a fairly compact way using dynamic programming. This result has been known for decades, and has not seen wide acceptance. Despite the relative complexity of this model, it still falls short of important real world concerns. High on this list is the ability to deal with many different items simultaneously, subject to bundling constraints, as well as handling uncertainty in demands. Both of these issues tend to destroy the tree structure of the optimal solution that is the foundation of the dynamic programs that have been formulated.

5.7. Optimal control methods for continuous time formulations

The preponderance of models and algorithms for dynamic networks assume a discrete time formulation. While considerably less progress has been made in the area of continuous time formulations, several authors have studied the problem in some depth. Frank [1967] and Frank & El-Bardai [1969] use optimal control theory to optimize flows over communication networks. Tapiero [1971] is the first paper to formulate the dynamic transportation problem using optimal control. Tapiero [1975] solves the single commodity dynamic transportation problem using optimal control; this is extended in Tapiero & Soliman [1972] to the multicommodity case. A nice survey of these results is given in Bookbinder & Sethi [1980]. Ellis & Rishel [1974] consider the airline flow management problem as an optimal control problem, which they solve in discrete time using dynamic programming. D'Ans & Gazis [1976] formulate the management of flow over congested transportation networks using optimal control.

At this point, optimal control methods for network optimization have not been applied in a practical setting. While progress has been made, considerably more work is required in both the formulation and solution of actual problems. In addition, we need to evaluate the value of a continuous time formulation over existing discrete time models that are widely used today.

6. Infinite horizon network models

The previous section presented a number of ways in which algorithms could be designed for dynamic problems. In this section, we address one of the most important problems that arise in dynamic models, which is the challenge of

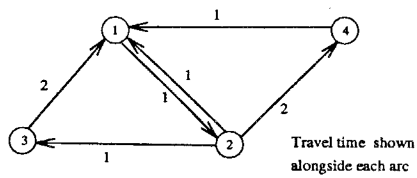


Fig. 14. A sample static network.

dealing with infinite planning horizons. The presentation begins in Section 6.1 with stationary dynamic networks, which helps to provide insights into the structure of optimal solutions of infinite networks. Then, Section 6.2 reviews a series of model ending techniques for approximating infinite horizon problems with finite approximations.

6.1. *Stationary infinite-horizon networks*

One of the earliest results for dynamic networks is given by Ford & Fulkerson [1956, 1958a, b, 1962] who considered the problem of finding maximal dynamic flows from the solution of a static network (see Figure 14). Consider a static graph $\mathcal{G} = \{\mathcal{N}, \mathcal{A}, \tau, l, u\}$ where $\mathcal{N}$ and $\mathcal{A}$ are sets of nodes and (directed) arcs, τ is a vector of (nonnegative) travel times, and l and u are vectors of lower and upper bounds. Assume flow enters through a super source q and leaves through a supersink r, and that $i \in \mathcal{N}$ are transshipment nodes. The problem is to determine the maximum flow that can be pushed from q to r over P time periods (all flow must exit the network by time period P). Let $V(P)$ be the total flow leaving the source node over time periods $1, \dots, P$. Then the maximum dynamic flow problem for stationary graphs can be restated as:

$$\max V(P)$$

subject to

$$\begin{aligned}
&\sum_{t=0}^{P} \sum_{i \in \mathcal{N}} \left[x_{qit} - x_{iq,(t-\tau_{iq})}\right] - V(P) = 0 && \\
&\sum_{t=0}^{P} \sum_{i \in \mathcal{N}} \left[x_{ir,(t-\tau_{ir})} - x_{rit}\right] - V(P) = 0 && \\
&\sum_{j \in \mathcal{N}} [x_{ijt} - x_{ji,(t-\tau_{ji})}] = 0 && i \in \mathcal{N} \\
&x_{ijt} \geq l_{ij} && i, j \in \mathcal{N} \\
&x_{ijt} \leq u_{ij} && i, j \in \mathcal{N}
\end{aligned}$$

Figure 15 shows how the original static network can be transformed into a time-expanded, stationary dynamic network. The path from 1, 2, 3, 1 is shown in bold, which spans four time periods. As a result, we can build four distinct paths which cover the same nodes at different points in time.

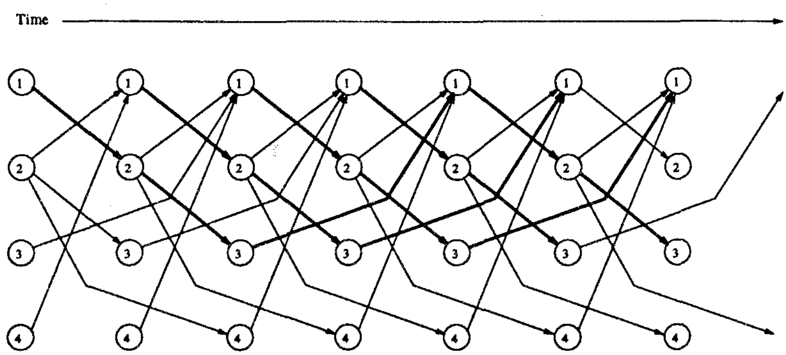


Fig. 15. Stationary, dynamic representation of the static network.

Our interest in this problem is not so much to develop an efficient solution algorithm, but rather to learn about the properties of an optimal solution for both finite and infinite networks. The central result for maximal dynamic network flows, given in Ford & Fulkerson [1958a], is derived from the following intuition. Let $\mathcal{M}$ be a set of directed chains from q to r, and let τ_p be the travel time for path $p \in \mathcal{M}$. We can construct $(P - \tau_p + 1)$ distinct copies of this path, each of which can carry a flow $f_p(t), t = 1, \ldots, (P - \tau_p + 1)$. This is done by starting the first path at time 1; the second path holds in inventory for one time period and then starts at time 2; the last path holds in inventory for $(P - \tau_p)$ time periods and then starts in period $(P - \tau_p + 1)$. Now let f be the vector of path flows. We say that f is *periodic* if $f_p(t) = f_p(t+k)$, for $t^o \leq t \leq t'$, where t^o and t' define boundaries of network boundary conditions. Furthermore, we say that f^* is *stationary* if $k = 1$. Let F^{*d} be the set of all optimal dynamic path flows and let F^{*s} be the set of optimal *stationary* path flows. We can now state the following result:

Theorem 20 [Fulkerson, 1962, Theorem 9.1, p. 149]. $F^{*s} \in F^{*d}$.

Thus, to find the optimal dynamic path flows, we can restrict our attention to stationary path flows, implying $f_p(1) = \ldots = f_p(t) = f_p$. The maximum dynamic flow problem can then be restated as:

$$\max_{f_p} \sum_p (P - \tau_p + 1) f_p \tag{98}$$

Let $v = \sum_p f_p$ be the maximum flow (where $v = V(P)$), and let the link-path incidence relations over the static graph be given by:

$$\delta_p^{ij} = \begin{cases} 1 & \text{if link } (i, j) \text{ is in path } p \\ 0 & \text{otherwise} \end{cases}$$

Clearly

$$\tau_p = \sum_{ij} \delta_p^{ij} \tau_{ij} \tag{99}$$

$$x_{ij} = \sum_{p} \delta_p^{ij} f_p \tag{100}$$

Substituting (99) and (100) and the definition of v into (98) gives the following optimization problem over the static graph:

$$\max_{v, x_{ij}} \left[(P+1)\, v - \sum_{ij \in \mathcal{A}} x_{ij} \tau_{ij} \right]$$

subject to a static version of the flow conservation constraints. Thus, given a stationary dynamic network, there exists an optimal solution which is also stationary, and which can be found by solving a static network problem. A set of static flows can then be expanded into a set of stationary flows.

The study of infinite networks leads to understanding the properties of solutions to stationary networks with finite planning horizons. Gale [1959] provides the following result.

Theorem 21. *Let $G(P)$ be the P-period expansion of $\mathcal{G}$, and let $f(P)$ be the optimal dynamic flows for $G(P)$. Then:*

(a) If $f(P)$ is optimal for $G(P)$, it is not *necessarily optimal for $G(P')$ with $P' \leq P$.*

(b) There does exist a flow pattern of (P) that is *optimal for all $G(P')$, $P' \leq P$. In this case, $f(P)$ is termed a* universal maximal dynamic flow.

(c) A universal dynamic flow is not necessarily stationary.

Result (c) is surprising, considering that stationary optimal flows exist for all $G(P)$. The implication of this result from a modeling perspective is that a stationary optimal flow pattern $f(P)$ may be unexpectedly sensitive to the choice of planning horizon P. Gale [1959] also finds universal dynamic flows with time varying capacities.

These basic results have been extended by others. Wilkinson [1971] and Minieka [1973] show how the original Ford & Fulkerson [1958a] algorithm can be modified to produce universal dynamic flows. Minieka [1974] provides a further modification of the Ford and Fulkerson algorithm for networks where arcs may be added or dropped from the network in any time period (a special case of the problem considered in Gale [1959]). A conclusion of this research, however, is that if the number of arc changes is 'excessive' then it is better to work directly with the time expanded graph.

An important extension of these results is given by Orlin [1983, 1984a, b]. Orlin [1983] considers the extension of the Ford and Fulkerson result to infinite dynamic networks (with stationary data) where he maximizes the *throughput* of the network instead of the total flow (which would be negative). The throughput $v(t)$ of the network is defined as the total flow in transit in period t (that is, in a dynamic graph, this would be the total flow crossing a line between periods t and $t+1$). It is shown that for some period t^o, which can be viewed as the end of the initial transient period, that $v(t) = v(t^o) = v$ for all $t \geq t^o$.

Consider now the problem:

$$\max_{x_{ij}} \sum_{ij\in\mathcal{A}} x_{ij}\tau_{ij}$$

subject to

$$\sum_j (x_{ij} - x_{ji}) = 0 \qquad i \in \mathcal{N}$$

$$x_{ij} \geq l_{ij} \qquad i, j \in \mathcal{N}$$

$$x_{ij} \leq u_{ij} \qquad i, j \in \mathcal{N}$$

which is a problem defined over the static graph. The optimal static flows x can now be extended into stationary dynamic flows x_t using methods similar in spirit to those developed by Ford and Fulkerson. Decompose x into a set of (static) cycle flows, f_p, where τ_p is the transit time around a cycle. Starting with a cycle at time 1, we can now form an infinite path composed of successive repetitions of the same cycle. We can form τ_p node disjoint copies of this same path, starting at times $1, \ldots, \tau_p$ (the cycle starting at time $t = 1$ returns to the same node by time $\tau_p + 1$ and repeats itself). Let f^s_{pt} denote the stationary path flows created by pushing the static flows f_p along these infinite paths, and let x^s_t be the associated stationary link flows. The major result of Orlin [1983] can be stated as:

Theorem 22. *If the problem is bounded, then:*

(a) The optimum dynamic flow among all stationary flows equals the optimum dynamic flow among all dynamic flows.

(b) The optimum throughput equals the minimum upper capacity of a cut (which is monotone).

(c) If the upper and lower bounds are integral, then the optimum stationary flows may be taken to be integral.

Part (a) of the theorem is the infinite analog of the Ford and Fulkerson result. Part (b) is the dynamic analog of the Ford and Fulkerson max-flow, min-cut theorem.

An important corollary of these results is that they all apply to the *minimum* throughput problem, which is obtained by multiplying all flows and bounds by -1. The minimum throughput problem is applicable to the problem of minimizing the number of vehicles required to serve a predetermined set of schedules that must be repeated in each time period.

A related problem considered in Orlin [1984b] is the problem of minimizing long-run average costs in dynamic networks with convex costs. If $c_{ij}x_{ijt}$ is the cost in period t on link (ij), then we wish to solve

$$\min_{x_{ijt}} \lim_{P\to\infty} \frac{1}{P} \sum_{t=0}^{P} \sum_{i,j\in\mathcal{N}} c_{ij}x_{ijt} \tag{101}$$

An optimal solution can be found for this problem by solving the following static network problem with a side constraint:

$$\min_{x_{ij}} \sum_{i,j\in\mathcal{N}} c_{ij}x_{ij}$$

such that:

$$\sum_j (x_{ij} - x_{ji}) = 0 \quad i \in \mathcal{N} \tag{102}$$

$$\sum_{ij} \tau_{ij} x_{ij} = 0 \tag{103}$$

Equation (103) is called the *throughput constraint* (note that we do not require $x_{ij} \geq 0$). One effect of this constraint is that the optimal solution may be noninteger. Orlin presents an algorithm for rounding this solution to obtain an optimal integer dynamic flow (which is periodic).

6.2. Model ending techniques for infinite horizon networks

Stationary dynamic networks provide intuition into the structure of optimal solutions for infinite, dynamic networks. However, the actual problem being solved is highly specialized. A richer set of problems arise in the solution of rolling horizon problems, which typically exhibit transient data.

A general statement of an infinite dynamic network is given by:

$$\min_{x_t} \sum_{t=0}^{\infty} \alpha^t f_t(x_t) \tag{104}$$

subject to, for $t = 0, 1, 2, \ldots$

$$\begin{array}{llll} A_0 x_0 & & & = R_0 \\ B_{0,1}x_0 + A_1 x_1 & & & = R_1 \\ B_{0,2}x_0 + B_{1,2}x_1 + A_2 x_2 & & & = R_2 \\ B_{0,3}x_0 + B_{1,3}x_1 + B_{2,3}x_2 + A_3 x_3 & & & = R_3 \\ \vdots & \ddots & & \vdots \\ B_{0,t}x_0 + B_{1,t}x_1 + B_{2,t}x_2 + & \ldots & + A_t x_t & = R_t \end{array} \tag{105}$$

Here, $f_t(x_t)$ is the cost in period t and $\alpha < 1$ is a discount factor. The matrices B_{t_1,t_2}, $t_1 < t_2$, capture the impact of decisions made in period t_1 on period t_2 (if there is no impact, the coefficient is zero). If each constraint corresponds to the flow conservation constraint for a node, then the terms $B_{t_1,t_2}x_{t_1}$ captures the flow into a node in period t_2 from period t_1. The matrices A_t capture the flow out of each node to all future time periods (if some flow goes from period t to period t, then A_t must also capture movements within the time period). The vectors R_t give the surplus/deficit at each node i, where $R_{it} > 0$ represents flow entering the network.

Five methods have been suggested for approximating infinite horizon problems. These include:

1. salvage value;
2. boundary conditions;
3. planning horizon approaches [Morton, 1979];
4. dual equilibrium [Grinold, 1983a];
5. primal equilibrium [Grinold, 1983a].

Below we provide a brief summary of the first four approaches. The primal equilibrium uses concepts similar to the dual equilibrium [see Grinold, 1983a].

6.2.1. Salvage value methods

Salvage value methods involve solving a finite problem and then handling end effects by adding an ad hoc penalty for flow left in the system at the end of the planning horizon. Let S_t be the total flow entering each node in period t from periods $t' < t$, where

$$S_t = \sum_{t'=0}^{t-1} B_{t',t} x_{t'} \tag{106}$$

Let ν be a vector of salvage values, giving the value of a unit of flow in each node in time period $P + 1$. We can now replace equation (104) with:

$$\min_{x_t, S_{P+1}} \sum_{t=0}^{P} \alpha^t f_t(x_t) + \nu^T S_{P+1} \tag{107}$$

subject to constraints (105) for $t = 0, \ldots, P$.

The calculation of ν is ad hoc; it depends on the specific characteristics of the problem, and there is generally not a formal, mathematical basis for it. In inventory problems, this can be a measure of the cost of ending inventories. In dynamic fleet management problems, it can be an estimate of the value of a vehicle at a particular point in the system. In railroad applications, this is sometimes the problem of returning a freight car owned by another railroad (known as a foreign car) back to its home railroad. In any event, we use the term salvage value to refer to any ad hoc penalty that is used to reduce end effects.

6.2.2. Boundary condition methods

Boundary conditions attempt to handle end effects by actually specifying ending inventories or flows, as opposed to simply specifying a penalty. For example, management might specify ending inventories of goods or vehicles in each location.

$$\min_{x_t} \sum_{t=0}^{P} \alpha^t f_t(x_t)$$

subject to constraints (105) for $t = 0, \ldots, P$ and

$$\begin{aligned} S_{P+1} &\geq \bar{S}^l \\ S_{P+1} &\leq \bar{S}^u \end{aligned} \tag{108}$$

where $\bar{S}^l$ and $\bar{S}^u$ represent a specified vector of minimum and maximum final inventories. Equation (108) represents a boundary condition on the final flows. In this example, we are requiring that the supply to each node at the end of the planning horizon be at least $\bar{S}^e$. By constraining the total flow at the end of the planning horizon, we can try to force the model to avoid extreme end of horizon strategies. We could also specify maximum limits, or both minimum and maximum limits. In any event, like salvage values, these limits are generally chosen on an ad hoc basis.

6.2.3. Planning horizon methods

Planning horizon approaches represent a form of model truncation, but where the length of the planning horizon is chosen in such a way to ensure, at least empirically, a good quality solution. This is probably the most widely used approach in practice. Assume that we would like to perform planning over a horizon period of $0, \ldots, P$. To avoid distortions in period P, we solve the model over a longer horizon P', $P' \geq P$. The idea is to choose P' large enough so that the decisions in the interval $0, \ldots, P$ are of 'good' quality. The problem can be viewed as one of solving:

$$\min_{x_t} \sum_{t=0}^{P} c^T x_t + \sum_{t=P+1}^{P'} c^T x_t \tag{109}$$

subject to flow conservations constraints over $t = 0, \ldots, P, \ldots, P'$. In contrast with the salvage value approach, which uses a linear function to handle end effects, the planning horizon approach simply extends the model from period P to period P'. The extension can be thought of as a nonlinear function of x_P. To see this, let S_{P+1} be the ending flows given $x_0, x_1, \ldots, x_P$, and define $G(S_{P+1})$ as follows:

$$G(S_{P+1}) = \min_{x_{P+1},\ldots,x_{P'}} \sum_{t=P+1}^{P'} c^T x_t$$

subject to flow conservation constraints over periods $t = P, \ldots, P'$, and given the initial state S_{P+1}. Although the objective function is linear, as a result of the flow conservation constraints, $G(S)$ is a nonlinear (piecewise linear) function of the initial state vector S_{P+1}, which of course is a linear function of the initial flows $x_0, x_1, \ldots, x_P$.

Planning horizon methods are also referred to as *forward* methods for dynamic optimization [Morton, 1981]. The use of forward methods is reviewed in much greater depth in Morton [1979]. The planning horizon literature (see Section 3.4.3) is based on finding forecast horizons P^f such that longer planning horizons $P' > P^f$ will not change the decision in the first P^d periods, $P^d \geq 1$. P^d is then called a decision horizon. Exact forecast/decision horizons have only been found for very special problems, and hence practitioners focus on finding horizons P' that produce 'good' quality solutions for the first time period only.

6.2.4. Dual equilibrium method

The most comprehensive body of research on solving infinite horizon problems is that presented by Grinold [1977, 1983a, b], Hopkins [1971], Grinold & Hopkins [1973] for deterministic problems, and Flam & Wets [1987] for stochastic problems. Grinold [1983a] presents the *dual equilibrium* approach for replacing infinite horizon models with finite approximations. This work was applied by Hughes & Powell [1988] for dynamic networks arising in the dynamic vehicle allocation problem, and has been applied in other contexts as well [see, for example, Murphy & Soyster, 1986].

We start with the model presented by equations (104) and (105). Following Grinold [1983a], we designate the first τ^I time periods as the *transient phase*, representing data that reflects the initial state of the network. For example, the transient phase would capture realizations of forecasted data, which may extend for several days into the future. We let x_0 represent decisions in the first phase, which may represent several time periods. After τ^I, the data is assumed to follow a stationary, periodic pattern with period τ^p. In many applications, the natural period for the stationary phase is one week. If this is the case, then we can simplify the model in equations (104) and (105) by:

$$\begin{aligned} B_{t_1,t_2} &= B_{t_2-t_1} && t_1 \geq 1,\ t_2 > t_1 \\ A_{t,t} &= A && t \geq 1 \end{aligned}$$

With this simplification, equation (105) becomes:

$$\begin{array}{lllll} A_0 x_0 & & & & = R_0 \\ B_{0,1} x_0 + A x_1 & & & & = R_1 \\ B_{0,2} x_0 + B_1 x_1 & + A x_2 & & & = R_2 \\ B_{0,3} x_0 + B_2 x_1 & + B_1 x_2 & + A x_3 & & = R_3 \\ \vdots & & & \ddots & \vdots \\ B_{0,t} x_0 + B_{t-1} x_1 + B_{t-2} x_2 + & & \ldots & + A x_t & = R_t \end{array} \tag{110}$$

The dual equilibrium method works by aggregating decisions in the future periods. Costs in period t are weighted by $(1-\alpha)\alpha^{t-1}$ and summed for $t \geq 1$, where $\sum_{t=1}^{\infty}(1-\alpha)\alpha^{t-1} = 1$. The aggregate link flows for the stationary phase would be:

$$x(\alpha) = (1-\alpha)\sum_{t=1}^{\infty}\alpha^{t-1}x_t$$

The constraints (110) are aggregated by multiplying the equation for period t by $\alpha^{t-1}(1-\alpha)$ and summing, giving:

$$\begin{aligned} &(1-\alpha)\left[\sum_{t=1}^{\infty}\alpha^{t-1}B_{0,t}\right]x_0 + \\ &\quad + (1-\alpha)\sum_{t=1}^{\infty}\alpha^{t-1}x_t\left\{A + \left[\sum_{t=1}^{\infty}\alpha^{t-1}B_{t-1}\right]\right\} = (1-\alpha)\sum_{t=1}^{\infty}\alpha^{t-1}R_t \end{aligned} \tag{111}$$

Define

$$\begin{aligned} R(\alpha) &= (1-\alpha)\sum_{t=0}^{\infty}\alpha^{t-1}R_t \\ B(\alpha) &= (1-\alpha)\sum_{t=1}^{\infty}\alpha^{t-1}B_{0,t} \\ A(\alpha) &= A + \sum_{t=1}^{\infty}\alpha^{t-1}B_{t-1} \end{aligned}$$

Substituting these back into (111) gives:

$$B(\alpha)x_0 + A(\alpha)x(\alpha) = R(\alpha)$$

Furthermore, if $f_t(x_t)$ is convex then we can write

$$\begin{aligned}\sum_{t=1}^{\infty} \alpha^t f_t(x_t) &= \frac{\alpha}{(1-\alpha)}\left[\sum_{t=1}^{\infty}(1-\alpha)\alpha^{t-1} f_t(x_t)\right] \\ &\geq \frac{\alpha f(x(\alpha))}{(1-\alpha)} \qquad (112)\end{aligned}$$

where (112) holds with equality when $f_t(x_t)$ is linear (which is typically the case in transportation). If $f_t(x_t) = c^T x_t$, then

$$\begin{aligned}\sum_{t=1}^{\infty} \alpha^t c^T x_t &= c^T x_0 + \alpha \sum_{t=1}^{\infty} \alpha^{t-1} c^T x_t \\ &= c^T x_0 + \frac{\alpha c^T x(\alpha)}{(1-\alpha)} \qquad (113)\end{aligned}$$

The dual equilibrium finite approximation is now given by:

$$\min\ c^T x_0 + \frac{\alpha c^T x(\alpha)}{(1-\alpha)} \qquad (114)$$

subject to

$$\begin{aligned} A_0 x_0 &= R_0 && (115)\\ B(\alpha)x_0 + A(\alpha)x(\alpha) &= R(\alpha) && (116)\\ x_0, x(\alpha) \geq 0 & && (117)\end{aligned}$$

Hughes & Powell [1988] demonstrate this approach for dynamic networks, motivated by the dynamic fleet management problem. The assumption is made in this paper that the transient phase is long enough for all vehicles in transit at $t = 0$ to appear before the beginning of the first stationary phase, implying $R_t = 0, t \geq 1$. Furthermore, the longest travel time is assumed to be shorter than the length of the stationary phase, implying $B_{0,t} = 0, t \geq 2$, and $B_t = 0, t \geq 2$. The important result in Hughes & Powell [1988] is that for a dynamic network, equations (114)–(117) form a generalized network, depicted in Figure 16 for a two city problem with two time periods in each phase. The links that pass from the transient to the stationary phase carry a link multiplier of $1 - \alpha$. Zero cost arcs carry flow from the third set of nodes back to the first set, which feature an arc multiplier of α.

The intuitive foundation of the dual equilibrium method is the assumption that flows in the stationary phases will quickly reach a stationary pattern, implying

$$x_{\simeq} x_2 \simeq \ldots \simeq x \qquad (118)$$

Note that since each phase can consist of several time periods, our use of the term stationary is somewhat broader than Ford and Fulkerson. The research on maximal dynamic flows, in particular the work by Orlin [1983, 1984b], suggests that data will *tend* to follow a stationary pattern, supporting the intuition behind (118). At the same time, the presence of initial conditions from the transient phase, as well as the fact that the stationary phases are periodic (as opposed to strictly stationary), suggests that (118) will never be more than an approximation.

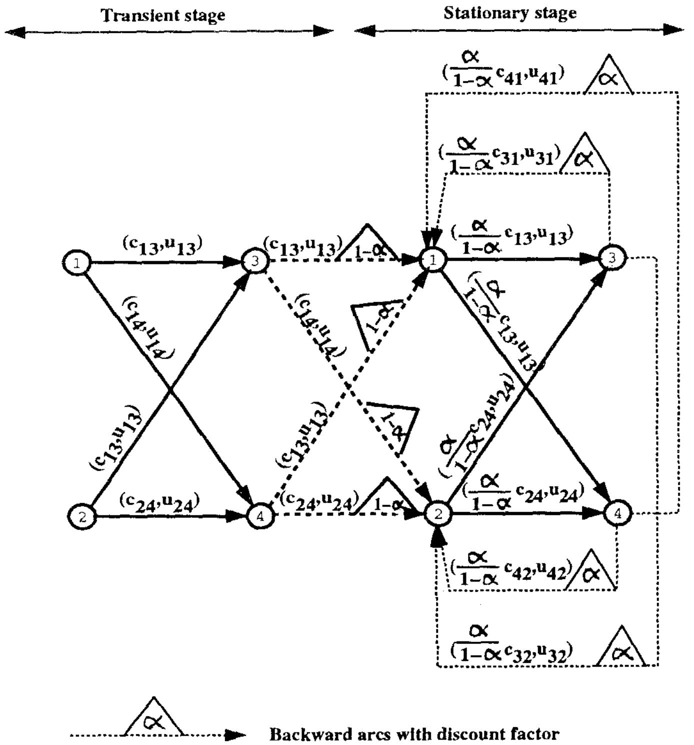


Fig. 16. Generalized network representation of dual equilibrium procedure.

The dual equilibrium method creates flows $x(\alpha)$ in the stationary phase with the same throughput as at the end of the transient phase. As a result, the upper bounds in the aggregate stationary phase are the same as in an individual stationary phase. This occurs since if $x_1 = x_2 = \ldots = x$, then $x(\alpha) = x$. However, the cost coefficients in the stationary phase are all factored by $\alpha/(1-\alpha)$.

Hughes & Powell [1988] present a somewhat more intuitive approach for aggregating future time periods, referred to as the generalized summation method. In this approach, a more straightforward net present value is used to represent future activities. Thus:

$$x(\alpha) = \sum_{t=1}^{\infty} \alpha^t x_t$$

subject to constraints (105), with the definitions of $R(\alpha)$, $B(\alpha)$ and $A(\alpha)$ modified to use a weight of α^t for period $t, t \geq 1$. This problem is also a generalized network, with the same form as that given in Figure 16. The only changes are (a) the multipliers $(1-\alpha)$ on links from the transient to stationary phase are replaced with α, (b) cost coefficients in the stationary phase are unfactored, and (c) the upper bounds on links in the stationary phase are factored by $\alpha/(1-\alpha)$. The dual equilibrium and generalized summation methods are effectively equivalent and in numerical experiments give comparable results.

A related result by Grinold & Hopkins [1973] gives conditions under which an infinite dynamic network can be solved exactly as a finite linear program. The main restriction is that the matrices A, B_t and $A(\alpha)$ are *Leontief*. A matrix A is Leontief if and only if there is at most one positive element in each column, and there exists $x \geq 0$ such that $Ax > 0$. These conditions are satisfied if the network is uncapacitated (a condition that would never be satisfied for dynamic fleet management problems). Nonetheless, if this condition were satisfied, we could solve the infinite horizon problem as follows:

First solve:

$$\min \; c^T y$$

such that

$$\begin{aligned} A(\alpha)y &= e \\ y &\geq 0 \end{aligned} \tag{119}$$

where e is a vector of ones. Let π^* be the optimal dual variables associated with constraint (119). This problem can be viewed as a kind of discounted infinite shortest path problem for a vehicle entering each node in the aggregate stationary phase. Grinold & Hopkins [1973] give some technical conditions for the existence of π^*. The product $B_1 x_0$ gives the flows into each node of the first stationary phase. The following problem:

$$\min_{x_0} \left[c^T + (\pi^*)^T B_1\right] x_0$$

will then give the same optimal first phase solution as the original discounted infinite network. Not surprisingly, numerical work reported in Hughes & Powell [1988] indicates that a Leontief model is a poor approximation of the future periods. This result also suggests that linear approximations of future activities will not, in general, be very effective.

There are some significant weaknesses in this approach to the solution of infinite horizon network problems. First is the fact that the dual equilibrium/generalized summation method will return a fractional solution, whereas the solution of a truncated model with any planning horizon [equations (70)–(71)] will return an integer solution (even when a discount factor is used). The second weakness lies in the heuristic basis of a discount factor in a deterministic model. In practice, the discount factor α is not used to determine the time value of money, but rather is an ad hoc approach for capturing uncertainty in the data. As a result, if the length of a stationary phase is one week, a weekly discount factor as low as $\alpha = 0.3$ might be used to reflect forecasting uncertainties.

7. Stochastic programming for networks

The most natural framework for writing stochastic formulations of dynamic fleet management problems is stochastic programming. Since the original articles of Beale [1955] and Dantzig [1955], a substantial literature has developed in the area of solving linear programs under uncertainty. The bulk of this research has been in the context of general linear programs and therefore can in principle be applied to stochastic formulations of all the models described in Section 4. However, as we demonstrate below, classical applications of stochastic programming methods can be extremely difficult to apply to stochastic networks with many random variables.

In this section, we summarize basic results from stochastic programming as a foundation for developing approximations which can be effectively applied to large stochastic networks. The problem is extremely rich, and as a result the presentation here is necessarily brief. Section 7.1 reviews some elementary concepts and notation. Section 7.2 surveys basic solution approaches that are used in stochastic programming, which are divided between methods for approximating the recourse function, scenario aggregation techniques, and finally stochastic gradient techniques. These are then covered in Sections 7.3, 7.4 and 7.5, respectively.

Throughout the presentation, we focus on results that have been specifically applied to, or are especially well adapted to, dynamic networks. This narrow tour through stochastic optimization, then, is intended to highlight emerging lines investigation into stochastic, dynamic networks.

7.1. Basic concepts and notation

Deterministic dynamic network models attempt to simultaneously find an optimal set of decisions x_t over all periods t in a planning horizon, illustrated by the simultaneous optimization problem:

$$\min_{x \in \mathcal{X}} \sum_{t=0}^{P} c^T x_t$$

In stochastic models, a simultaneous optimization formulation does not recognize the sequencing of decisions and realizations. It is important to identify 'what you knew and when you knew it.' To do this, we must define our random variables much more carefully. For the presentation here, we assume the only source of randomness is in the right-hand-side constraints:

ξ_{ijt} = market demand for loads moving from i to j in period t.

ξ_t = $\{\ldots, \xi_{ijt}, \ldots\}$.

We denote by $(\Omega, \mathcal{F}, \mathcal{P})$ the underlying probability space for ξ_t where Ω is a set of elementary outcomes, $\mathcal{F}$ is a set of events, and $\mathcal{P}$ is a probability measure defined over Ω. Let $\omega \in \Omega$ denote an elementary outcome, where $\omega = (\omega_1, \omega_2, \ldots, \omega_t, \ldots, \omega_P)$, $\omega_t \in \Re^m$, denotes the set of elementary outcomes associated with period t. For example, ω_t might be a vector representing the individual decisions of each shipper. We then let $\xi_t(\omega_t)$ be a *realization* of the market demands in period t. If there are n markets, then

$$\xi_t(\omega_t) : \Re^m \to \Re^n$$

represents the aggregation of individual shipper demands to market demands.

For a given outcome ω_t, we need to determine an optimal set of actions $x_t(\omega_t)$. These actions depend on the *history* of the system, which is summarized by all previous outcomes and actions. Let H_t be the history of the process up to and including period t:

$$H_t = \left[(x_0), (\xi_1(\omega_1), x_1(\omega_1)), (\xi_2(\omega_2), x_2(\omega_2)), \ldots, (\xi_t(\omega_t), x_t(\omega_t))\right]$$

The problem is to choose a set of actions x_t given the history H_{t-1} and the outcome ω_t. For this reason, we should write the decision vector as $x_t(H_{t-1}, \omega_t)$ to communicate this dependence on past history and the current set of outcomes. For compactness, we use the notation $x_t(\omega_t)$, to signal the dependence on the current outcome, with the dependence on past history implicit. For additional simplicity, we may use $x_t(\omega)$, where we assume $\omega = \omega_t$ in this context. In practice, with suitable independence assumptions, we can capture past history with a single state vector S_t of greatly reduced dimension. In this case, we assume $x_t(\omega_t)$ is implicitly a function of S_t. The notation $x_t(\omega_t)$ also communicates the property that actions in period t must be made before we know the outcomes $\omega_{t+1}, \omega_{t+2}, \ldots$.

A period t where all data is known, and where decisions must be made prior to realizations of future activities, is referred to as a *stage*. The vast majority of the stochastic programming literature addresses two-stage problems, where decisions must be made in the first, deterministic stage which account for the impacts of future events, which are uncertain. Further, we assume that different actions may be taken in the future once the outcomes of random events become known. This class of problems is referred to as two-stage stochastic programs with recourse. The general two-stage stochastic linear program with recourse can be stated as:

$$\min_{x \in \mathcal{X}_0} c^T x_0 + E_{\omega_t}\left[Q_1(x_0, \omega_1)\right] \tag{120}$$

where

$$\mathcal{X}_0 = \{x \mid A_0 x = R_0, x \geq 0\}.$$

We define

$$Q_1(x_0, \omega_1) = \inf_{y \in \mathcal{X}_1(x_0, \omega_1)} c_1(\omega_1)^T y$$

where

$$\mathcal{X}_1(x_0, \omega_1) = \{y \mid A_1(\omega_1) y = R_1(\omega_1) - B_1(\omega_1) x_0\}$$

In this formulation, we allow $c(\omega)$, $A(\omega)$, $R(\omega)$ and $B(\omega)$ to be stochastic. B is generally referred to as the *technology* matrix while A is the *recourse* matrix. For example, B describes the network which determines how the flows x_0 in period 0 enter period 1 (thus the network is the technology of the problem). $A_1(\omega)$

then determines the set of permissible actions in period 1 (the recourse). If A_1 is deterministic, the problem is said to have *fixed recourse*. If A_1 represents a node-arc incidence matrix, then the problem has *network recourse*. Finally, if we can express A_1 as

$$A_1 = [I, -I]$$

where $y = (y^+, y^-)^T$, $y^+, y^- \in \Re^m$, and $I \in \Re^{m \times m}$ is the identity matrix, then the problem has *simple recourse*. If the problem is feasible for all $x_0 \in \Re^n$, then the problem has *complete recourse*. If $Q_1(x_0, \omega_1) < \infty$ for all $x_0 \in \mathcal{X}_0$ and $\omega \in \Omega$, then the problem is said to have *relatively complete* recourse. For network problems that arise in practice, $\mathcal{X}_t(\omega)$ is nonempty and *compact*, which implies that we can replace the infimum with a simple minimization. $Q_1(x_0, \omega)$ is called the *recourse function*, with the *expected recourse function* given by

$$E\big[Q_1(x_0)\big] = \int\limits_{\omega \in \Omega} Q_1(x_0, \omega)\, \mathrm{d}P(\omega) \tag{121}$$

The integration of (121) is computationally intractable for most problems of interest. This implies that not only can we not solve (121) using standard methods, we cannot even calculate the objective function. Thus, the difficulty of working with equation (121) is the heart of what makes stochastic programming a difficult problem.

Dynamic network problems often require looking several time periods into the future. These problems must be formulated as multi-stage stochastic programs to capture the imbedded structure of realizations and decisions. The generic structure of a multistage stochastic program is:

$$\min_{x_0} c^T x_0 + E_{\omega_1}\Big\{ \min_{x_1} c^T x_1(\omega_1) + E_{\omega_2}\Big\{ \min_{x_2} c^T x_2(\omega_2) + E_{\omega_3}\{\ldots\}\Big\}\Big\}$$

This formulation exhibits the imbedded minimizations and expectations that are characteristic of multistage stochastic programs. Needless to say, the size and complexity of this problem grows explosively with the number of stages.

7.2. *Solution approaches in stochastic programming*

Methods for solving stochastic programs can be organized into three groups which progress from easy to hard problems:

1. approximating the expected recourse function;
2. scenario methods;
3. stochastic gradient procedures.

The first two methods are based on the seminal papers by Wets on the equivalent convex program [Wets, 1966] and the equivalent deterministic program [Wets, 1974]. The equivalent convex program is the mathematical foundation of the first class of techniques, which seek to replace the expected recourse function with an analytical approximation which allows the first stage problem to be solved using standard optimization techniques. These techniques are only tractable when the underlying problem and the event space exhibit special structure. The second approach, scenario methods, is based on the equivalent deterministic program. This involves solving a large, combined optimization problem for both stages and all possible future scenarios. Finally, the third approach, stochastic gradient methods, uses sampling methods to calculate gradients used in search procedures.

A complete discussion of these approaches is well beyond the scope of this chapter. Excellent reviews can be found in Wets [1982, 1988], Birge & Wets [1986] (approximation schemes for stochastic programs), Vladimirou [1991] (scenario methods), Frantzeskakis [1990] (bounding procedures) and Chong [1991] (stochastic approximation methods). A very small percentage of this work is directed at dynamic networks. At the same time, some of the concepts are easily and directly applied to dynamic networks and should not be ignored simply because the literature has not addressed this special case. Just the same, the scope of our presentation is limited to dynamic networks and related results that readily lend themselves to dynamic networks. Section 7.3 outlines basic results that are used to develop approximations of recourse functions. These ideas are then applied in Section 9 for dynamic networks with random arc capacities. Section 7.4 gives a summary of basic results for scenario methods and their application to stochastic networks. Finally, Section 7.5 introduces stochastic gradient procedures and describes how they can be used to solve more complex problems.

7.3. Approximating the recourse function

Consider again the basic stochastic program (120) with recourse function $Q_1(x_0, \omega)$ and expected recourse function:

$$\bar{Q}_1(x_0) = \int_{\omega\in\Omega} Q_1(x_0, \omega)\,\mathrm{d}P(\omega) \tag{122}$$

Further, as we defined earlier, let $\mathcal{X}_0$ be the feasible region for the first stage and let $\mathcal{X}_1(\omega)$ be the feasible region for the second stage given x_0 and outcome ω. In general stochastic programs, the issue of feasibility of the second stage is an important problem. For dynamic fleet management problems, however the presence of unbounded inventory arcs guarantees that the second stage exhibits *complete recourse*. Van Slyke & Wets [1969] show that $Q_1(x_0, \xi(\omega))$ is a continuous, convex function in x_0 over $\mathcal{X}_1$ and that $\bar{Q}_1(x_0)$ is also convex in x_0 Wets [1966] shows that (120) can be written as the following *equivalent convex* program:

$$\min c^T x_0 + \bar{Q}_1(x_0) \tag{123}$$

subject to

$$x_0 \in \mathcal{X}_0$$

$$W x_1 \le \alpha \tag{124}$$

where the matrix W and vector α are developed from geometric properties of the feasible region $\mathcal{X}_0 = \bigcap_{\omega\in\Omega} \mathcal{X}_0(\omega)$. For dynamic networks which exhib complete recourse (the second stage is feasible for any x_0), (124) is not needed.

The equivalent convex problem establishes a framework that the recourse function can be viewed as a convex function in x_0, as opposed to a large embedded set of optimization problems given in (161). In practice, $\bar{Q}_1(x_1)$ is impossible to find explicitly. However, we can view (123) as a basis for developing approximations. Birge & Wets [1986] provide an extensive review of approximation schemes for stochastic programs. Some basic strategies that can be used to approximate the recourse function include:

1. primal decomposition methods (such as restricted recourse strategies);
2. dual decomposition methods (Lagrangian relaxation);
3. cutting plane techniques;
4. response surface methods.

Primal decomposition methods impose restrictions on the set of feasible strategies that can be used in the optimization in the second stage [see Frantzeskakis & Powell, 1989a]. The best example of this is simple recourse, where the optimization in the second stage is replaced by a trivial optimization problem whose expectation can be found analytically (see Sections 9.1.2 and 9.2). Dual decomposition methods seek to simplify the calculation of the expected recourse function by relaxing complicating constraints (see Section 9.4). These methods are illustrated in some depth in Section 9 in the context of dynamic networks with random arc capacities.

Response surface methods take samples to estimate a nonlinear approximation of the expected recourse function. This idea was first introduced in Beale, Forest & Taylor [1980] and further investigated in Beale, Dantzig & Watson [1986]. However, there is very little research using this technique.

7.4. Scenario methods

Assume the set Ω consists of the finite set of outcomes $\omega_1, \omega_2, \ldots, \omega_s, \ldots, \omega_L$ with probabilities p_s. Then (120) can be written:

$$\min\ c_0^T x_0 + \sum_{s=1}^{L} c_1^T x_1(\omega_s) p_s \tag{125}$$

subject to

$$\begin{array}{lllll}
A_0 x_0 & & & & = R_0 \\
B_0 x_0 + A_1 x_1(\omega_1) & & & & = R_1 \\
B_0 x_0 & + A_1 x_1(\omega_2) & & & = R_1 \\
\vdots & & \ddots & & \vdots \\
B_0 x_0 & & & + A_1 x_1(\omega_L) & = R_1 \\
\quad x_1(\omega_1) & & & & \le \xi_1(\omega_1) \\
& x_1(\omega_2) & & & \le \xi_1(\omega_2) \\
& & \ddots & & \vdots \\
& & & x_1(\omega_L) & \le \xi_1(\omega_L)
\end{array} \tag{126}$$

An alternative formulation of the same problem is the *full variable split* form of the problem. In this version, we introduce the variable

$x_0(\omega_s)$ = actions in the first period given scenario s in the second stage.

We then replace x_0 with $x_0(\omega_s)$ in equation (126). Of course, we cannot have a set of actions in the first stage that depends on the outcome of the second stage. The goal of stochastic programming is to choose a single set of actions for the first stage that minimizes expected costs over both stages for all possible scenarios. As a result we need to introduce the *nonanticipativity* constraint:

$$x_0(\omega_s) - x_0 = 0 \tag{127}$$

This form of the nonanticipativity constraint has the same form as (126) with the variable x_0 in each constraint.

Problem (125) is a large optimization problem, where both the first and second stage problems are networks. However, while there is a different solution $x_1(\omega_s)$ for each scenario ω_s in the second stage, there is only a single solution x_0 in the first stage. Constraints (126) represent the flow conservation constraints for each scenario. Since x_0 appears in the flow conservation constraint for each scenario, the problem is no longer a network, and the resulting linear program is very large.

Much more problematically, the number of scenarios will in practice be exponentially large. In typical applications, $|\Omega|$ may easily be on the order of 10^{10} to 10^{1000} (assuming discrete random variables). The standard approach to solving (125) is to replace the set of outcomes Ω with a much smaller sample that is randomly chosen from the original population. In practice, a sample with as few as 100 observations, or smaller, can still be computationally difficult. A number of strategies have been developed to solve this problem, drawing from the field of large scale optimization:

1. techniques for staircase linear programs [see, for example, Wets, 1966];
2. L-shaped decomposition [Van Slyke & Wets, 1969];
3. interior point algorithms for linear programs [see Lustig, Mulvey & Carpenter, 1991];
4. the progressive hedging algorithm [Rockafellar & Wets, 1987].

For the most part, these methods do little to take advantage of the underlying network structure of the problem. An exception is the progressive hedging algorithm [Rockafellar & Wets, 1987] which works by relaxing the nonanticipativity constraint and solving an augmented Lagrangian problem. If the original problem is a linear network, the result of this decomposition is an algorithm where at each iteration it is necessary to solve a sequence of nonlinear network problems, thus retaining at least some of the original structure.

The progressive hedging algorithm has been applied to stochastic networks with random arc capacities by Kornhauser & Maslanka [1989]. Rolling horizon experiments are reported and compared to the results given in Frantzeskakis & Powell [1990] on the same dataset. Although the scenario approach did not perform as well as the SLAP algorithm of Frantzeskakis and Powell, the results were generally good. Execution times, however, were substantially higher.

Kornhauser & Chen [1990] further experimented with stochastic, capacitated multicommodity flow problems. Scenario methods have been most widely applied to stochastic dynamic networks for financial planning [Mulvey & Vladimirou, 1989; Vladimirou, 1991]. Financial problems characteristically exhibit complex correlations in the random variables which give the return from a particular investment. Scenario methods allow these correlations to be captured without requiring strong independence assumptions.

7.5. Stochastic gradient methods for complex systems

Scenario methods are effective for two-stage stochastic programs with two characteristics: a) the recourse problem $Q_1(x_0, \omega_s)$ can be formulated as an optimization problem for a particular outcome $\omega_s \in \Omega$, and b) the sample space Ω can be reasonably approximated by a small set of outcomes (scenarios). It is not uncommon for one or both of these qualifications to be violated in the context of large, complex problems. For example, it might be relatively easy to calculate the future consequences of decisions made now using a *simulation* model, but may be virtually impossible to formulate future activities using an *optimization* model. Thus, we can find a statistical estimate of $\bar{Q}_1(x_0)$, but we cannot find $\bar{Q}_1(x_0)$ exactly and, more importantly, we cannot calculate the gradient $\nabla_x \bar{Q}_1(x_0)$ exactly. The second qualification is when we are able to formulate $Q_1(x_0, \omega_1)$ as an optimization problem, but we cannot accurately approximate the sample space using a 'reasonable' number of outcomes. This problem is more common in stochastic, dynamic networks where the number of stochastic elements in each period can be quite large. Instances where $|\Omega|$ is on the order of 10^{100} are quite common, implying the sample space is effectively infinite. We can still take a small sample from this population, but even 10 or 20 scenarios can create an extremely large deterministic equivalent problem.

Stochastic approximation procedures handle these problems by successively sampling from the set of outcomes and then obtaining an *estimate* of the derivative of the objective function. Under fairly mild assumptions, convergent algorithms can be developed which are quite easy to implement. The limitation of the approach is that the rate of convergence can be quite slow.

The first stochastic approximation procedure was given in Robbins & Monroe [1951] which sought to solve a problem in regression analysis. Let $\bar{F}(x) = E_\omega F(x, \omega)$ be a monotone function of x, and assume $\bar{F}(x^*) = \theta$ for a given value of θ. The problem is to find x^*, given that we can find $F(x, \omega)$ but cannot exactly calculate $\bar{F}(x)$. Kiefer & Wolfowitz [1952] consider the related problem:

$$\min_x E_\omega F(x, \omega) \tag{128}$$

for $x \in \Re^1$ and show how the method of Robbins and Monroe could be used to find a sequence x_n that converges stochastically to x^*. Blum [1954] extends this to problems where $x \in \Re^n$ using a sequence of the form

$$x^{k+1} = x^k + \alpha_k g^k(x^k, \omega^k) \tag{129}$$

where $g^k(x^k, \omega^k)$ is a *statistical estimate* of the gradient of $\bar{F}(x)$ which satisfies

$$E_\omega g(x, \omega)^T \nabla_x \bar{F}(x) < 0 \tag{130}$$

which is to say that the expectation of the gradient must be a valid descent vector (we assume that $F(x, \omega)$ is differentiable). Blum shows that $\lim_{k\to\infty} x^k = x^*$ almost surely under the following conditions:

$$\sum_{k=1}^{\infty} \alpha_k \quad = \infty \tag{131}$$

$$\sum_{k=1}^{\infty} \alpha_k^2 \quad < \infty \tag{132}$$

$$|\nabla \bar{F}^2(x)| \; < \infty \tag{133}$$

Conditions (131) and (132) are very standard in this literature. One sequence that satisfies them is $\alpha_n = 1/n$. Condition (131) can be viewed as ensuring that the algorithm will not stall out. Condition (132), combined with certain limits on the function itself, guarantees that the *variance* of the solution x^k goes to zero as $k \to \infty$. An important characteristic of this method is that while the *expectation* of $g^k(x, \omega)$ must be a valid descent vector as required by (130) a specific *realization* $g(x, \omega)$, used in (129), may not be a descent vector. As a result, the objective function may actually increase from one iteration to the next.

Stochastic approximation methods have been studied by a number of authors. A thorough review of the field is given in Wasan [1969]. However, these methods apply only to unconstrained problems. For constrained programs (and problems where $F(x, \omega)$ is not differentiable), Ermoliev [1969] introduced the notion of stochastic quasigradient methods, drawing on the concepts behind stochastic approximation procedures and nondifferentiable optimization problems [see Shor, 1979]. One of the earliest applications of these methods to stochastic network problems is given in Powell & Sheffi [1982] and Sheffi & Powell [1982], where the approach was dubbed the method of successive averages (when $\alpha_k = 1/k$, equation (129) implies that x^k is an average of all previous gradients). An excellent review of stochastic gradient methods is given in Ermoliev [1988].

Consider the stochastic programming problem (with recourse):

$$\min_{x \in \mathcal{X}} \bar{F}(x) \tag{134}$$

where

$$\bar{F}(x) = E_\omega \Big[c^T x + \min_{y \in \mathcal{Y}(x,\omega)} q^T y \Big] \tag{135}$$

where $\mathcal{X}$ and $\mathcal{Y}(x, \omega)$ are compact, convex feasible regions for the first and second stage problems, where:

$$\mathcal{Y}(x, \omega) = \big\{ y \mid A(\omega) y = R(\omega) - B(\omega) x, \; y \geq 0 \big\} \tag{136}$$

for given matrices and vectors $B(\omega)$, $R(\omega)$ and $A(\omega)$. Let $g_x^k(x^k)$ be the subgradient of $\bar{F}(x)$ with respect to the first-stage decisions x. A stochastic quasi-gradient

(SQG) $g_x^k(x^k, \omega)$ is a subgradient of $\bar{F}(x)$ for a particular realization ω which satisfies:

$$E_\omega \left[g_x^k(x^k, \omega)\right] = \partial \bar{F}(x^k) + \beta^k \tag{137}$$

where $\partial \bar{F}(x^k)$ is a subgradient of $\bar{F}(x^k)$ and $\beta^k \in \Re^n$ is a *bias vector* which must satisfy:

$$\lim_{k \to \infty} \beta^k = 0$$

The bias vector simply generalizes the original condition (130) that the expectation of the stochastic gradient be a strict descent vector. The theory also allows $\bar{F}(x)$ to be nondifferentiable in x, requiring the use of the subgradient term in (137).

In the case of dynamic networks, it is quite easy to find $g_x^k(x^k, \omega^k)$ for a given realization ω^k. In this case, the constraint set (136) is just the flow conservation constraint for the second stage. Let x^{k-1} be the first stage flows from the previous iteration, and solve, for a given realization ω^k:

$$\min c^T y(\omega^k) \tag{138}$$

subject to

$$\sum_m y_{jm}(\omega^k) = R_j^1 + \sum_i y_{ij}^{k-1} \qquad j \in \mathcal{C} \tag{139}$$

$$y_{ij}(\omega^k) \leq \xi_{ij}(\omega^k) \qquad i, j \in \mathcal{C} \tag{140}$$

Let $\pi^k(\omega^k)$ be the dual of constraint (139). Note that for a particular outcome ω, we have only to solve a single stage *deterministic* network (using standard network flow algorithms). The dual $\pi_j^k(\omega^k)$ is then the standard dual for the flow conservation constraint. Our stochastic gradient is then:

$$g_x^k(x^k, \omega) = c + \pi^k(\omega^k)^T B(\omega) \tag{141}$$

where the matrix $B(\omega)$ is the node-arc incidence matrix for nodes j in the second stage and arcs in the first stage [in effect, the coefficients of x in (139)].

The stochastic gradient $g_x^k(\omega)$ is not a *feasible* descent vector. Two methods can be used to obtain a feasible descent vector. Section 7.5.1 discusses using a projection operation; Section 7.5.2 describes a stochastic linearization approach. Finally, Section 7.5.3 briefly reviews methods for using simulation techniques to estimate gradients.

7.5.1. Stochastic projection

The first method of obtaining a feasible direction is to use a simple projection operator $\Pi_{\mathcal{X}}(\ldots)$ to map points back onto the feasible region:

$$x^{k+1} = \Pi_{\mathcal{X}}\left[x^k - \alpha_k g_x^k(x^k, \omega)\right] \tag{142}$$

where α_k is the step size at iteration k. If the feasible region $\mathcal{X}$ is defined by flow conservation constraints, then the projection step is impractical. On the other hand, if we are using the flow splitting methods of Section 5.4, the projection operation is quite easy. While this method has not yet been tried, it may be quite promising.

The stepsize α_k plays an important role in smoothing the fluctuations between iterations. The only condition the stepsize sequence must satisfy is (131). Not surprisingly, the rate of convergence can be extremely poor. One step that has been found to accelerate convergence is to smooth the estimate of the gradient vector. Let

$$g_x^k(x^k, \omega) = c + \pi^k(\omega^k)^T B(\omega) \tag{143}$$

represent a sample of the gradient of iteration k. Now let

$$\bar{g}_x^{k+1}(\omega) = \beta_k \bar{g}_x^k(\omega) + (1 - \beta_k) g_x^k(x^k, \omega) \tag{144}$$

where $0 \leq \beta_k \leq 1$. Equation (144) performs a successive averaging of sample gradients. We then use the averaged gradient $\bar{g}_x^k(x^k, \omega)$ in (129) instead of $g_x^k(x^k, \omega)$. This approach is sometimes referred to as a mixed stochastic gradient method.

7.5.2. Stochastic linearization

As an alternative, we could solve the stochastic linearization problem (this can be thought of as a kind of 'stochastic Frank–Wolfe'), first introduced by Gupal & Bajenov [1972]. Thus we would solve:

$$\min_{y \in \mathcal{X}} \bar{g}_y^k(\omega^k)^T y \tag{145}$$

where $\bar{g}^k(\omega)$ is calculated using (144). The feasible set $\mathcal{X}$ captures the network constraints in the first period. The attraction of (145) is that it is a linear network flow problem. The second stage problem, which is captured by the vector $\pi^k(\omega^k)^T B(\omega)$, adds no complexity to solving the first stage problem. Also, obtaining the duals $\pi_k(\omega_k)$ involves solving a deterministic network. Thus, this approach is extremely amenable to solving stochastic networks. However, considerable research is needed to evaluate and improve the rate of convergence, which is known to be quite slow for most problems.

Unlike the projection operation, the stochastic linearization method requires $0 < \beta_k < 1$ in equation (144) [see Ermoliev, 1988]. Let $\hat{y}^k(\omega)$ be the optimal solution of this subproblem at iteration k. Since $\hat{x}^k(\omega)$ is feasible, we can update the current solution using:

$$x^{k+1} = x^k + \alpha_k(\hat{y}^k(\omega) - x^k) \tag{146}$$

where again, α_k is a given stepsize.

7.5.3. Simulation methods for gradient estimation

Stochastic gradient procedures have the attractive feature that they accommodate complex models which cannot be formulated as optimization problems. Assume, for example, that given a state $S_1 = A_1 x_1$ for the second stage, we can only *simulate* future activities using Monte Carlo methods, yielding a statistical estimate of the recourse function, $Q_1(S_1)$. We now need to obtain an estimate of the gradient $h_1(S_1) = \nabla_s Q_1(S_1)$. When $Q_1(S_1)$ can be formulated as an optimization problem, this estimate is obtained from the dual of a realization of the recourse function. If we are forced to simulate the recourse function, then we can obtain an estimate of $h_i = \partial Q(S)/\partial S_i$ using finite difference methods [Kiefer & Wolfowitz, 1952]

$$h_i(S, \omega) = \left(\frac{1}{\Delta}\right)[Q(S + \Delta e_i, \omega) - Q(S, \omega)] \tag{147}$$

where Δ is a stepsize, e_i is a vector of zeroes with a 1 in the ith element, and ω is a sample realization. Given a sample gradient $h(S, \omega)$, the gradient $g^k(x^k, \omega)$ is found from

$$g^k(x^k, \omega) = c + h(S^k, \omega) \cdot B(\omega) \tag{148}$$

Kiefer & Wolfowitz [1952] and Ermoliev [1969] provide the conditions required to guarantee that the sequence x^k converges to the optimum when using (147) for unconstrained and constrained problems, respectively.

Another approach for estimating gradients is to use *infinitesimal perturbation analysis* (IPA). IPA was originally introduced by Ho, Eyler & Chien [1979] and is reviewed in depth in Glasserman [1991]. It has since been studied by a number of authors, primarily in the context of queueing networks (see Chong [1991] and the references cited there). The motivation behind IPA is to obtain an estimate of the gradient $h(s)$ by using a single simulation rather than the $n + 1$ simulations (assuming $S \in \Re^n$) implicit in the finite difference equations (147). The basic concept of IPA is to perform a single simulation while keeping track of events which would trace the impact of small changes in the vector S.

7.5.4. Remarks

An attraction of stochastic gradient procedures is that they easily handle complex stochastic networks. Each iteration involves solving a single realization of a stochastic network. We can accommodate uncertainty in arc capacities, travel times, costs, and even external supplies and demands.

Stochastic gradient methods provide a rigorous foundation for solving very complex stochastic networks that can arise in practice. In addition, they are particularly well suited to networks since the linear approximations of the recourse function produce network subproblems. At this time, stochastic approximation procedures have not been applied to stochastic, dynamic networks. As a result, there is no experimental evidence supporting the overall effectiveness of the approach. Stochastic approximation procedures in general are known to be quite slow in convergence, reflecting the weak assumptions required to guarantee convergence. Thus, these methods must be viewed as an algorithm of last resort.

8. Stochastic programming models in networks and routing

The basic concepts and framework presented in Section 7 can now be applied to two problems in logistics. Section 8.1 presents several models that have been used to formulate vehicle routing problems with uncertain demands. Then, Section 8.2 shows how the dynamic fleet management problem with random demands can be formulated as a multistage dynamic network with random arc capacities. Solution methods for this special type of stochastic program are described in Section 9.

8.1. Stochastic programming for vehicle routing

Consider a formulation of the standard vehicle routing problem with deterministic demands. We might define:

$$x_{ij}^k = \begin{cases} 1 & \text{if vehicle } k \text{ goes from } i \text{ to } j, \\ 0 & \text{otherwise.} \end{cases}$$

d_i = demand at customer i.

$$(VRP) \ \min_{x,y} F(x, y) = \sum_{i \in C} \sum_{j \in C} x_{ij}^k c_{ij} \tag{149}$$

subject to

$$\begin{aligned} \sum_{j \in C} x_{ij}^k - \sum_{l \in C} x_{jl}^k &= 0 \qquad \forall i \in C \\ \sum_k \sum_{j \in C} x_{ij}^k &= 1 \qquad \forall i \in C \\ \sum_{j \in C} x_{ij}^k - y_i^k &= 0 \qquad \forall i, k \\ \sum_{i \in C} y_i^k d_i &\le V \end{aligned} \tag{150}$$

Problem (*VRP*) is a standard formulation of the vehicle routing problem with deterministic demands. In this context, we first assume demands are known, and then determine the routing.

As described in Section 7, there are several strategies which can be used when demands are uncertain:

1. *Myopic model*. In this case, we ignore forecasted demands. The model would be run again as new demands were called in.

2. *Deterministic model*. Here, we simply forecast demands and then treat these forecasts deterministically within the model. This model would be identical to (VRP) above, except that some of the demands would represent realizations, and others would be expectations of forecasts. The model would normally be run again as each forecast became a realization.

3. *Chance constrained programming*. In this model, the vehicle capacity constraint in (150) would be replaced with a probabilistic inequality and solved as a nonlinear programming problem.

4. *Stochastic programming*. Stochastic programming is the most rigorous solution framework, but also the most complex. In this approach, we optimize the problem over the known demands, taking account the expected costs of responding in an optimal way to various demand outcomes (scenarios). Various

approximations of stochastic programs can be developed by imposing restrictions on how we allow the system to respond to specific demand outcomes.

Myopic and deterministic models, and chance constrained formulations, are all effectively deterministic models. Myopic models ignore forecasted demands; deterministic models typically use the expected demand, and chance constrained formulations, as we show below, are equivalent to solving the model with deterministic demands set at a particular percentile from the demand distribution. Stochastic programming formulations offer a much richer framework in terms of accounting for more complex behavior.

A nice review of chance constrained and stochastic programming formulations is given in Dror, Laporte & Trudeau [1989]. We first present a basic chance constrained model, and then discuss stochastic programming formulations.

8.1.1. Chance constrained models

In a chance constrained formulation [see Stewart & Golden, 1983], we replace the vehicle capacity constraint with:

$$P\Big(\sum_{i\in C} y_i^k d_i(\omega) \le V\Big) \ge (1-\alpha) \tag{151}$$

Equation (151) can be replaced with an analytical expression by making suitable assumptions about the distribution of $d_k(\omega)$. For example, if the demand is normally distributed with mean μ_i and variance σ_i^2, then (151) becomes (Stewart & Golden, 1983]:

$$\sum_{i\in C} y_i^k \mu_i + \gamma\Big(\sum_{i\in C}\sigma_i^2 (y_i^k)^2\Big)^{1/2} \le V \tag{152}$$

where γ is a constant chosen so that:

$$\begin{aligned} &P\left[\frac{\sum_i d_i^k y_i^k - M_k}{S_k} \le \gamma\right] = 1-\alpha \\ &M_k = \sum_l \mu_l y_l^k \\ &S_k = \sum_l \sigma_l^2 (y_l^k)^2 \end{aligned} \tag{153}$$

Stewart and Golden show that if (i) the demands are independent, (ii) the terms $(\sum_i d_i^k y_i^k - M_k)/S_k$ and $(d_i(\omega)-\mu_i)/\sigma_i$ have the same distribution for all i, and (iii) σ_i^2/μ_i is equal to the same constant θ for all i, then constraint (153) is equivalent to:

$$\sum_i^k \mu_i \le \bar{V} \tag{154}$$

where

$$\bar{V} = \frac{2V + \gamma^2\theta - (\gamma^4\theta^2 + 4V\gamma^2\theta)^{1/2}}{2} \tag{155}$$

Thus the stochastic vehicle routing problem with chance constraints reduces to a deterministic vehicle routing problem with demands μ_i and vehicle capacity $\bar{V}$.

8.1.2. *Stochastic programming with recourse*

Models of stochastic, dynamic problems can be differentiated based on what do we know, when do we know it, and what can we do about it once it is known. Most tractable models are characterized by strong assumptions that allow us to do very little once demands become known. Assume that, using some type of assumption about future demands, we make an initial decision about a tour. As demands become known, we then have to decide what assumptions are we going to make regarding our response to those demands. Note that we are not talking about the possibility of rerunning the model. Rather, we are making assumptions about our response so that we will make a better decision about what to do right now (before new data arrives).

We discuss three basic recourse strategies. The first is simple recourse, where constraint violations are assessed with a penalty. In this case, no recourse action is allowed in response to random demands, and infeasible problems may result. The second is a class of strategies we call *depot return recourse* where routes are decided in advance, and are not modified as demands become known. Finally, the third assumes full reoptimization at each step in the journey.

Simple recourse. Consider the basic VRP with random demands, denoted by $d_i(\omega)$. If we substitute random demands into the vehicle capacity constraint, we obtain:

$$\sum_{i \in C} y_i^k d_i(\omega) \leq V \tag{156}$$

which may of course be infeasible for a given realization $d_i(\omega)$. Now consider the addition of a recourse variable $y_i^+(\omega)$ to produce

$$\sum_{i \in C} y_i^k d_i(\omega) - y_k^+(\omega) \leq V \tag{157}$$

We can now formulate the vehicle routing problem with simple recourse as:

$$\min_{x,y} F(x, y) + E_\omega \left\{ \min_{y^+(\omega)} \sum_k c_k^r y_k^+(\omega) \right\} \tag{158}$$

where c_i^r is the recourse penalty cost for violating the capacity of the kth vehicle. The minimization inside the expectation is a trivial one, since for any realization ω, $y_k^+ = \max\{0, (\sum_{i \in C} y_i^k d_i(\omega) - V)\}$. When analytically tractable assumptions are made about the distribution of $d_i(\omega)$, it is normally possible to rewrite (158) in the following form:

$$\min_{x,y} F(x, y) + \bar{Q}(x, y) \tag{159}$$

where $\bar{Q}(x, y)$ is a nonlinear, convex function of x and y (as well as being a function of the recourse penalty c^r). For most local search techniques, this problem is no harder than one with linear costs. The drawback, of course, is that the penalty for violating the vehicle capacity will vary depending on the distance of the customer to other customers and the depot.

A restricted recourse strategy. Simple recourse strategies assume that random outcomes are handled by assessing penalties. As a next step, we can consider strategies where we allow a limited response. Consider the event where the total demands on a tour happen to exceed the capacity of the vehicle. Ideally, we would like to completely reoptimize the vehicle tours. A step in this direction is to consider restricted recourse strategies, as suggested in Frantzeskakis & Powell [1989a], is to *anticipate* only a limited set of responses to random demands.

For example, assume that vehicle routes are fixed in advance (based, for example, on forecasted demands). Then, as the vehicle completes each tour, instances where total demands exceed vehicle capacity are handled by inserting breaks in the tour. Thus, if a vehicle is picking up goods and reaches capacity before reaching the end of the tour, the vehicle returns to the depot, drops off its load, and then returns to the next customer on the original tour. Laporte & Louveaux [1990] first introduced this model, and suggested a second variation where customer demands are determined before the vehicle actually starts on the tour (but after the tour has been designed). The sequence of customers is not changed, but the point at which the vehicle stops and returns to the depot is optimized to determine the best break in the tour.

Dror & Trudeau [1986] propose an even more extreme model. In the event of a break in the route, they assume that the remainder of the route consists of trips back and forth to the depot for each customer. Thus, a potential route break carriers a very high penalty.

Full VRP recourse. It is useful to consider a more complete model to illustrate the complexities of modeling dynamic demands explicitly in the context of a multistage stochastic program. First, we need to characterize demands arriving to a central facility as a stochastic process. Assume we have up to N customers and let (d_n, τ_n) be the (random) demand, and the time τ that the demand is called in. Let a specific realization of this be denoted by $(d_n(\omega), \tau_n(\omega))$ and let $N_t \in N$ be the set of customers whose demands are known prior to time t. Assume time is indexed so that $t = 0$ refers to now, $t < 0$ refers to past events and $t > 0$ index future events. Assume t is discrete, and for simplicity, assume that all travel times between customers require exactly one time period.

Now let $H_t = \{(d_n(\omega), \tau_n(\omega)) | n \in N_t\}$ be the *history* of the process up to time t. We can then define $x_t(H_t)$ as the set of routes given data known up to time t and with forecasted information about customers $n \notin N_t$. Dropping the dependence on the history (which is implicit), x_t, $t < 0$, refers to the part of the tour that has not yet been completed, x_0 represents the decision we need to make now, and x_t, $t > 0$ represents the part of the tour that is being planned but will not be implemented until later, and therefore may be changed (given proper communication with the driver).

A multistage stochastic program would take the basic form:

$$\min_{x_0}\{c^T x_0 + E_{\omega_1}\{\min\{c^T x_1 + E_{\omega_2}\{c^T x_2 + \ldots\}\}\}\} \tag{160}$$

where in each stage we would constrain the problem to satisfy basic vehicle routing constraints. A tractable multistage model has not been proposed, and remains an active area of research.

8.2. A stochastic programming formulation of dynamic fleet management

We are now ready to write out the dynamic fleet management problem as a multistage stochastic program. We begin with the basic, single commodity formulation of DVA, where the only source of uncertainty is in the forecasts of future demands. We let $\xi_t(\omega_t)$ represent a particular outcome of ξ_t. Further, we let $\hat{\xi}_0$ denote the actual market demands in period 0, which are known. The stochastic DVA with random demands can be written:

$$\begin{aligned}\max_{x_0, y_0 \in \mathcal{X}_0} r^T x_0 - c^T y_0 \; + \; E_{\omega_1}\Bigg\{ & \max_{x_1(\omega_1), y_1(\omega_1) \in \mathcal{X}_1(\omega)} r^T x_1(\omega_1) - c^T y_1(\omega_1) \\ & + E_{\omega_2}\Bigg\{ \max_{x_2(\omega_2), y_2(\omega_2) \in \mathcal{X}_2(\omega)} r^T x_2(\omega_2) - c^T y_2(\omega_2) \\ & + E_{\omega_3}\Bigg\{ \ldots \Bigg\}\Bigg\}\Bigg\}\end{aligned} \tag{161}$$

where $\mathcal{X}_t(\omega)$ is the set of feasible x_t, y_t defined by, for $t = 0, \ldots, P$:

$$\begin{aligned} &\sum_j (x_{ijt}(\omega) + y_{ijt}(\omega)) - \sum_k \left[x_{ki,(t-\tau_{ki}(\omega))} + y_{ki,(t-\tau_{ki}(\omega))}\right] = R_{it} && i \in \mathcal{C} \\ &x_{ijt}(\omega_t) \qquad\qquad \leq \xi_{ijt}(\omega_t) \qquad i, j \in \mathcal{C} \\ &x_{ijt}(\omega_t), y_{ijt}(\omega_t) \geq 0 \qquad\qquad i, j \in \mathcal{C} \end{aligned} \tag{162}$$

where

x_{ijt} = flow of loaded vehicles from i to j in time period t.

y_{ijt} = flow of empty vehicles from i to j in time period t.

$\tau_{ij}(\omega)$ = realization of a travel time from i to j for a journey initiated in time period t.

If $t = 0$, the right hand side of (162) is replaced with $\hat{\xi}_{ij0}$. The set $\mathcal{X}_t(\omega_t)$ is a function of the history of the process H_{t-1} and the current outcome ω_t.

Problem (161) formulates the stochastic DVA as a multistage dynamic network with random arc capacities. This model was first given in Powell [1988]. Random travel times are captured by using $\tau_{ki}(\omega)$, which represents an outcome of a trip terminating in period t (this is not the most general possible model, but it serves our purposes here). Random travel times are equivalent to random node-arc incidence matrices with uncertainty in the location of the (-1) in each column.

Many of the extensions to the basic DVA discussed in Section 4.5 can be easily handled in the context of this framework. This includes multiple equipment types and transportation constraints that might represent rail or ship capacities. Traffic assignment problems, where smaller shipments must be consolidated into vehicles, are also easily accommodated.

Problem (161) is of course only a formulation, and does not readily lend itself to any existing solution procedures for problems of realistic size. It does, however, provide a framework and a starting point for the development of approximations.

As a first step, we can simplify (161) by introducing a state variable S_t defined by

$$\begin{aligned} S_{it} &= \text{total supply of vehicles in region } i \text{ at the beginning of stage } t. \\ &= \sum_{k \in \mathcal{C}} \left(x_{ki,(t-\tau_{ki})} + y_{ki,(t-\tau_{ki})} \right) \end{aligned}$$

We assume that S_t completely summarizes the state of the system in period t. This is technically true for this model only if $\tau_{ij} = 1,\ i, j \in \mathcal{C}$, but this problem can be handled by extending the state vector to account for vehicles in transit. Using this state vector, we can restate (161) as follows:

$$\max_{x_0, y_0 \in \mathcal{X}_0} \left[r^T x_0 - c^T y_0 + \bar{Q}_1(S_1) \right] \tag{163}$$

where $\bar{Q}_1(S_1)$ is the expected recourse function for stage 1, defined by:

$$\bar{Q}_1(S_1) = E_{\omega_1}[Q_1(S_1, \omega_1)] \tag{164}$$

$$Q_1(S_1, \omega_1) = \max_{x_1, y_1 \in \mathcal{X}_1(\omega_1)} \left[r^T x_1 - c^T y_1 + \bar{Q}_2(S_2) \right] \tag{165}$$

The expected recourse function $\bar{Q}_t(S_t)$ is found recursively, where $\bar{Q}_{P+1}(S_{P+1}) = 0$.

For the purposes of our presentation, we refer to formulations (161)–(162) as the *simultaneous* form of the stochastic program and (163)–(165) as the *recursive*, or *state variable*, form. While mathematically equivalent, the two formulations lead to very different lines of investigation for solving these problems.

8.3. Remarks

This section has presented stochastic models of two basic problems, vehicle routing and fleet management. At this point in time, explicit, dynamic models for vehicle routing problems are intractable, and research to date has considered only chance constrained and simple recourse models. There is not enough experimental evidence at this point to say conclusively whether these simple models are performing effectively. Furthermore, there is some anecdotal evidence that supports the use of myopic models that ignore forecasted demands, with possibly limited use (in a deterministic fashion) of forecasted data. It is quite likely, at this point, that myopic models, or dynamic models with deterministic forecasts, will provide considerable benefits, with additional incremental benefits coming from the development of models which handle stochastic demands in a more rigorous manner.

The model for dynamic fleet management is, of course, much simpler because it is linear and avoids the combinatoric properties of vehicle routing problems. As a result, we have been able to make considerably more progress in this area. These results are described in the next section.

9. Approximations for networks with random arc capacities

Section 7 reviews general results for stochastic programs that can be applied to stochastic networks. For the most part, these methods tend to do little to take advantage of the basic structure of the problem. In some cases, algorithms can be designed which require solving sequences of network subproblems. Beyond this, there are few simplifying assumptions, allowing these methods to be applied to fairly general problems. The price of this generality is algorithms that can be quite slow and cumbersome to use. For some problems, this generality is not required, and other techniques can be used to produce accurate models that can be easily solved using standard algorithms. In this section, we review specialized results for dynamic networks with random arc capacities, a framework that is well suited to modeling uncertainty in demand forecasts. In contrast with the results of Section 7, which focuses on two stage problems with recourse, we treat both two-stage and n-stage models in this section. Section 9.1 illustrates the use of several simple approximations, including simple recourse, chance constrained programming and deterministic approximations. Section 9.2 introduces the concept of null recourse, which arises in inventory-type problems. Section 9.3 reviews a method for approximating the recourse function for multistage networks using linear approximations of the recourse function. Finally, Section 9.4 introduces the use of decomposition methods for approximating the recourse functions. Two models are used to illustrate these methods. The first is the classical stochastic transportation problem. The second is the n-stage dynamic network with random arc capacities.

9.1. *Simple recourse and related methods*

There are several simple methods that can be used to solve stochastic networks with random arc capacities. We illustrate their use in the context of the stochastic transportation problem with random demands. Let:

$\mathcal{M}$ = set of markets, $m = |\mathcal{M}|$.

$\mathcal{N}$ = set of suppliers, $n = |\mathcal{N}|$.

x_{ij0} = flow from supplier i to market j in the first stage $x_0 \in \Re^{nxm}$.

S_{j1} = total flow supplied to market j in the second stage $S_1 \in \Re^m$.

ξ_{j1} = random variable giving the market demand in the second stage.

c = vector of transportation costs, $c \in \Re^m$.

r = vector of market revenues, $r \in \Re^n$.

R_{i0} = initial inventory in market i.

We wish to solve the following demand-constrained version of the stochastic transportation problem:

$$\min c^T x_0 + \bar{Q}_1(S_1) \tag{166}$$

subject to

$$\begin{aligned} A_0 x_0 \quad &= R_0 \\ B_0 x_0 - S_1 &= 0 \\ x_0,\ S_1 \quad &\geq 0 \end{aligned}$$

The expected recourse function captures the 'profits' to be earned from supplying amount S_{j1} to market j. The form of the recourse function depends on how we wish to model second stage activities. For a particular outcome ω, our basic model is:

$$Q_1(S_1, \omega) = \sum_{j \in M} (-r)^T \max(S_{j1}, \xi_{j1}(\omega)) \tag{167}$$

However, other models may be used which yield different solution procedures. Below, we briefly outline three simple methods.

9.1.1. Deterministic approximations

The most widely used model in practice replaces the random demands $\xi_1(\omega)$ with deterministic approximations, d_1. We may use $d_1 = E_\omega[\xi_1(\omega)]$, or choose a value d_1 that satisfies Prob $[\xi_1 \leq d_1] = \alpha$, for a given service level α. Let:

y_{j1} = amount of satisfied demand.

Then we may write

$$\begin{aligned} S_{j1} - y_{j1} &\geq 0 \\ y_{j1} &\leq d_1 \\ y_{j1} &\geq 0 \end{aligned} \tag{168}$$

Now the recourse function is given simply by $\bar{Q}_1(S_1) = -r^T y_1$, and (166) reduces to a transportation problem with two demand arcs from each market node to a super sink, one with upper bound d_1 and cost $-r$ and the other serving as an overflow arc.

9.1.2. Simple recourse

If we did not use deterministic demands above, equation (168) would become

$$y_{j1} \leq \xi_{j1}(\omega) \tag{169}$$

which is not well-defined for all outcomes ω (unless we require the constraint to be defined *almost surely*, which implies that it must hold for *all* outcomes ω). As an alternative, we may introduce the following *recourse variables*:

$$y_{j1}^+(\omega) = \max[0, \xi_{j1}(\omega) - y_{j1}] \tag{170}$$

$$y_{j1}^-(\omega) = \max[0, y_{j1} - \xi_{j1}(\omega)] \tag{171}$$

$y_1^+(\omega)$ is a random variable giving the unsatisfied demand (the underage) and $y_1^-(\omega)$ gives the excess supply (the overage). We may now write the demand constraint as:

$$y_{j1} + y_{j1}^+(\omega) - y_{j1}^-(\omega) = \xi_{j1}(\omega)$$

and the flow conservation constraint as:

$$B_0 x_0 - y_1 = 0$$

Assume that excess supply costs r^- and unsatisfied demand costs r^+. Note that the satisfied demand is given by $y_{j1} - y_{j1}^-$, which produces revenue r. The conditional recourse function is now given by:

$$Q(y_1, \omega) = \sum_j -r^T(y_{j1} - y_{j1}^-(\omega)) + (r^-)^T y_1^-(\omega) + (r^+)^T y_1^+(\omega)$$

The recourse function is separable in the recourse variables $y_{j1}^+(\omega)$ and $y_{j1}^-(\omega)$, which allows the expected recourse function to be found very simply. Depending on the distribution function for $\xi_1(\omega)$, the expectations

$$\bar{y}_{j1}^+(s) = \int_\omega \max[0, \xi_{j1}(\omega) - s]\,\mathrm{d}P(\omega)$$

$$\bar{y}_{j1}^-(s) = \int_\omega \max[0, s - \xi_{j1}(\omega)]\,\mathrm{d}P(\omega)$$

may be found as analytical functions of the supply $s = y_{j1} = S_{j1}$, or numerically as piecewise linear functions. As a result, we may write

$$\begin{aligned}\bar{Q}(y_1) &= E_w Q_1(y_1, \omega)\\ &= -r^T y_1 + (r + r^+)^T \bar{y}_1^+(y_1) + (r^-)^T \bar{y}_1^-(y_1)\end{aligned}$$

which is separable in the vector y_1. Now the original problem is a simple nonlinear network, with a linear transportation problem augmented by nonlinear arcs from each market to the supersink. The new network is illustrated in Figure 17.

The stochastic transportation problem is a multidimensional analog of the original newsboy problem and has been studied by a number of authors [Williams, 1963; Szwarc, 1964; Cooper & Leblanc, 1977; Qi, 1985; among others]. Williams [1963] exploited this feature to develop a specialized algorithm which relaxed the *supply* constraints, and then used the result from inventory theory that the optimum amount to be supplied to each market is $(r_j^- - c_{ij})/(r_j^- - r_j)$, for $r_j^- \geq c_{ij} > r$. Williams [1963] shows that there exists a *certainty equivalent* ξ^* that gives a deterministic problem whose solution x_1^* is the same as the stochastic problem. Cooper & Leblanc [1977] demonstrate how the Frank–Wolfe algorithm can be used to solve the nonlinear network. Qi [1985] presents the forest iteration method which further exploits the structure of the problem to develop an efficient solution algorithm. Wallace [1986] presents an algorithm which groups realizations of the demand vector which share the same optimal basis. Finally, Zipkin [1982] presents results for aggregating market nodes for very large stochastic transportation problems.

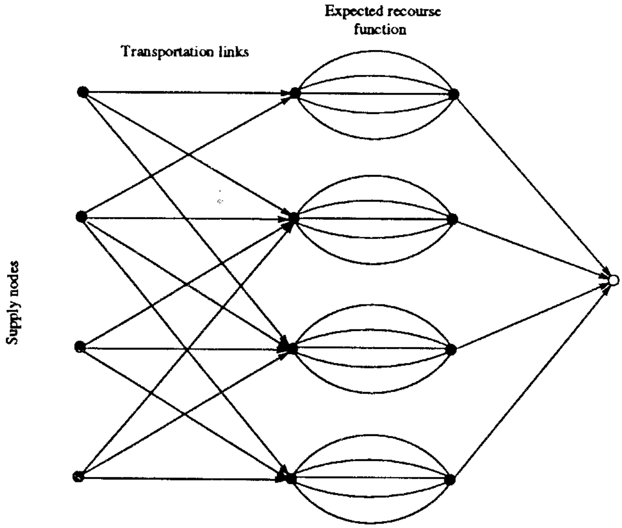


Fig. 17. Network representation of stochastic transportation problem with simple recourse.

9.1.3. Chance constrained programming

A third approach for dealing with random bounds is to use the concept of *chance constraints*, introduced by Charnes & Cooper [1959]. Here, the basic demand constraint (169) is replaced with:

$$\text{Prob}\,[y_{j1} \geq \xi_{j1}] \geq \alpha \tag{172}$$

where α is a specified *service level.* Chance constraints are useful when managers are more comfortable specifying the fraction of the market demand that should be satisfied. This is often more natural than specifying overage and underage costs.

9.1.4. Extension to multistage problems

These three solution methods can be applied to multistage stochastic networks, although their interpretation is not as natural. Consider the n-stage dynamic network with random arc capacities, given in Section 8.2. The complicating constraint in this problem is the random upper bounds on each link. In stage t, this constraint is given by:

$$x_{ijt} \leq \xi_{ijt}(\omega_t) \tag{173}$$

In a deterministic formulation, constraint (173) is replaced by a deterministic estimate d_{ijt}:

$$x_{ijt} \leq d_{ijt} \tag{174}$$

which transforms the problem into a n-period deterministic dynamic network. If $d_{ijt} = \bar{\xi}_{ijt} = E_{\omega_t}[\xi_{ijt}(\omega_t)]$, then it can be shown using Jensen's inequality that the optimal solution is a lower bound on the original n-stage recourse problem

[Frantzeskakis & Powell, 1990], but the bound can be quite loose. If the expected upper bounds are small, fractional solutions can become a problem. Frantzeskakis and Powell use the rounding procedure:

$$d_{ijt} = \lfloor \bar{\xi}_{ijt} + \gamma_{it} \rfloor$$

where γ_{it}, $0 \leq \gamma_{it} \leq 1$, is chosen so that

$$\sum_j d_{ijt} \cong \sum_j \bar{\xi}_{ijt}$$

Alternatively, we may use simple recourse to handle the random upper bounds. As before, we introduce recourse variables $x^+_{ijt}(\omega_t)$ and $x^-_{ijt}(\omega_t)$ in a manner analogous to the stochastic transportation problem. The random arc capacity is now stated as:

$$x_{ijt} + x^+_{ijt}(\omega_t) - x^-_{ijt}(\omega_t) = \xi_{ijt}(\omega_t)$$

In the context of the dynamic vehicle allocation problem, $x^+_{ijt}(\omega_t)$ gives the unsatisfied demand, and $x^-_{ijt}(\omega_t)$ is the flow of excess vehicles, which must therefore move empty. x_{ijt} is the total flow of vehicles (loaded and empty), and $x_{ijt} - x^-_{ijt}(\omega_t)$ is the number of vehicles moving loaded. Flow conservation constraints are written entirely as a function of x_{ijt}. As before, if the distribution of $\xi_{ijt}(\omega_t)$ follows a simple form, the expected overage and underage $\bar{x}^+_t$ and $\bar{x}^-_t$ can be found as functions of x_{ijt}. This model was first introduced by Powell, Sheffi & Thiriez [1984] and Powell [1988]. The use of simple recourse produces an n-period nonlinear dynamic network, which can be solved using the methods described in Section 5.4.

Finally, we can enforce (173) using a chance constraint, producing a dynamic network with nonlinear arc constraints. An effective solution procedure can be developed by relaxing the chance constraint and adding it to the objective function. The difficulty with both simple recourse and chance constrained programming in the context of the DVA is the physical interpretation of the constraints. For example, simple recourse assumes that any of the x_{ijt} vehicles that cannot move loaded will move empty anyway. This is a very restrictive assumption that is likely to be very inaccurate if the expected upper bounds are small.

9.2. Null recourse

Simple recourse is a strategy where responses to random arc capacities are handled by using a parallel overflow link. As a result, while the flow on the random arc from node i to node j may be a random variable, the total flow on both the random arc and the overflow arc joining i and j is deterministic, greatly simplifying the problem. For dynamic fleet management problems, this is equivalent to assuming that if a vehicle does not move loaded between i and j, it will move empty.

A different model allocates a certain amount of flow to move over a link with a random arc capacity. If this flow is greater than the capacity of the arc for a specific realization, the excess flow is assumed to spill to the inventory link to be used in the subsequent time period. An important consequence of this policy is that future supplies of vehicles are random.

This problem can be approached by extending the basic approach developed in Section 5.4 for nonlinear dynamic networks. Although their presentation is quite different, Jordan & Turnquist [1983] effectively use this approach to solve a stochastic version of the empty freight car allocation problem. Their problem is to allocate a fleet of empty freight cars around the system to meet forecasted demands. They assume that once a car is moved empty from i to j, it remains in inventory at j until the spatially distributed, car is consumed. Thus, the problem becomes a dynamic inventory planning problem. Their model allows for uncertainty in demands and travel times. Each node has a stochastic *inventory* of cars S_{it} which is to be used to satisfy demands. Unused inventories of cars at a location are held in place until the next time period. The inventories of cars S_{it} are stochastic due to uncertainty in the travel times. For this problem, we could define the decision variables as:

θ_{ijt} = fraction of the *supply* of cars S_{it} to be moved empty from i to j at time t.

Reverting back to our original notation for the dynamic vehicle allocation problem, let y_{ijt} represent flows of empty cars where $y_{ijt} = S_{it}\theta_{ijt}$. Jordan and Turnquist assume S_{it} is normally distributed with known mean and variance, which implies y_{ijt} is also normally distributed with moments that are easily calculated. Let:

D_{it} = random variable describing the demand for cars at node i at time t.

and assume D_{it} is also normally distributed with known mean and variance. If S_{it} is the available inventory in period t, then $\max\{0, S_{it} - D_{it}\}$ is the remaining inventory to be held over to the next period. The total inventory in period $t+1$ is then

$$S_{j,t+1} = \max\left\{0, S_{jt} - D_{jt}\right\} + \sum_i S_{i,t-\tau_{ij}}\theta_{ij,(t-\tau_{ij})}$$

Jordan and Turnquist develop approximations for the mean and variance of $S_{j,t+1}$, using normal approximations everywhere. Derivatives of the objective function with respect to the decision variables θ_{imt} are developed using backward recursions similar to that developed for deterministic flows.

Powell [1986] extends this approach to the full dynamic vehicle allocation problem with loaded and empty moves. In this model, the demands are given by

D_{ijt} = number of loads available to be moved from i to j in period t (no backlogging allowed).

The decision variables are:

α_{ijt} = fraction of the supply of available vehicles in node i at time t *allocated* to be moved loaded from i to j at time t.

β_{ijt} = fraction of the supply to be moved empty.

Let:

x_{ijt} = *actual* number of vehicles that are moved loaded from i to j in period t.

R_{it} is used as the *exogenous* supply of vehicles while S_{it} is the endogenous flow of vehicles through node i at time t. D_{ijt}, S_{it}, y_{ijt} and x_{ijt} are all random variables that are approximated using an Erlang distribution. The flow of loaded vehicles is given by:

$$\begin{aligned} x_{ijt} &= \min\left\{D_{ijt}, S_{it}\theta_{ijt}\right\} \qquad i, j \in \mathcal{C},\ i \neq j && (175)\\ x_{iit} &= \sum_{j\in\mathcal{C}}\left[S_{it}\theta_{ijt} - x_{ijt}\right] \qquad i \in \mathcal{C} && (176)\end{aligned}$$

Equation (176) expresses the assumption that if $S_{it}\theta_{ijt} > D_{ijt}$, then the excess vehicles are assumed to be held in inventory. Thus, if the flow allocated to move over the link exceeds the random upper bound, we assume that we do nothing with the excess flow (that is, we do not move capacity 'empty' as we did in simple recourse), hence the term null recourse.

It is possible to formulate this problem as a convex, nonseparable, nonlinear programming problem to maximize total expected profits over the planning horizon. Derivatives are again calculated using backward recursions, leveraging the same basic structure given in (96). The Frank–Wolfe algorithm works quite well for this problem, although it is very easy to implement other algorithms with better convergence [Bertsekas, 1979; Bertsekas, Gafni & Gallagher, 1984; Powell, Berkkam & Lustig, 1992].

The real power of this approach is its ability to exploit the dynamic structure of the problem and to handle forecasting uncertainties in a simple way. Its major limitation is that it is exceptionally difficult to develop a nonlinear cost function for strategies that are much more general than fixed flow allocations. For example, the 'move or hold' strategy implicit in the inventory equations (175)–(176) produces a fairly cumbersome set of equations. Even small generalizations can be intractable in this framework. One example is a strategy where a vehicle that cannot be moved loaded over link (i, j) is used to satisfy a demand on another link (i, k).

9.3. Successive linear approximation procedure

A different approach to solving the n-stage (161) is suggested by Powell [1987] and studied more formally by Frantzeskakis [1990] and Frantzeskakis & Powell [1990] (see also the discussion in Assad, Wasil & Lilien [1992, pp. 124–129]). This approach seeks to develop a separable, nonlinear approximation of the recourse function $Q_1(S_1)$ which would allow the first stage problem to be solved using standard optimization methods. We begin by restating the basic problem:

$$\min c^T x_0 + \bar{Q}_1(S_1) \tag{177}$$

subject to

$$\begin{aligned} A_0x_0 \quad &= R_0\\ B_0x_0 - S_1 &= R_1\\ x_0 \leq \hat{\xi}_1&\\ x_0 \geq 0&\end{aligned}$$

where $\bar{Q}_t(S_t)$ is defined recursively as:

$$\begin{aligned} \bar{Q}_t(S_t) &= E_{\omega_t}[Q_t(S_t, \omega_t)] \\ \bar{Q}_t(S_t, \omega_t) &= \min_{x_t, S_{t+1}} c_T x_t + \bar{Q}_{t+1}(S_{t+1}) \end{aligned}$$

subject to

$$A_t x_t = R_t + S_t \tag{178}$$

$$B_t x_t - S_{t+1} = 0$$

$$x_t \leq \xi_t(\omega_t) \tag{179}$$

$$x_t \geq 0$$

We now introduce the restriction that the network problem in each stage has a bipartite structure, where every link moves from a node in stage t to a node in stage $t+1$. Thus, each stage is like a one-sided transportation problem with supply constraints (178), random arc capacities (179), and nonlinear market cost functions $\bar{Q}_{t+1}(S_{t+1})$.

The successive linear approximation procedure (SLAP) develops an approximation of $\bar{Q}_1(S_1)$ by starting at period $t = P$, and building a sequence of linear approximations. The approach is quickly illustrated in Figure 18. First, we use the structure of the last time period to calculate exactly the expected recourse function, which is a piecewise linear, convex function. Then, we replace the nonlinear recourse functions with linear approximations (these linear approximations can also be formed to create both upper and lower bounds). Finally, we add the slope of the linear approximation to the cost of each arc coming into that node. Now we

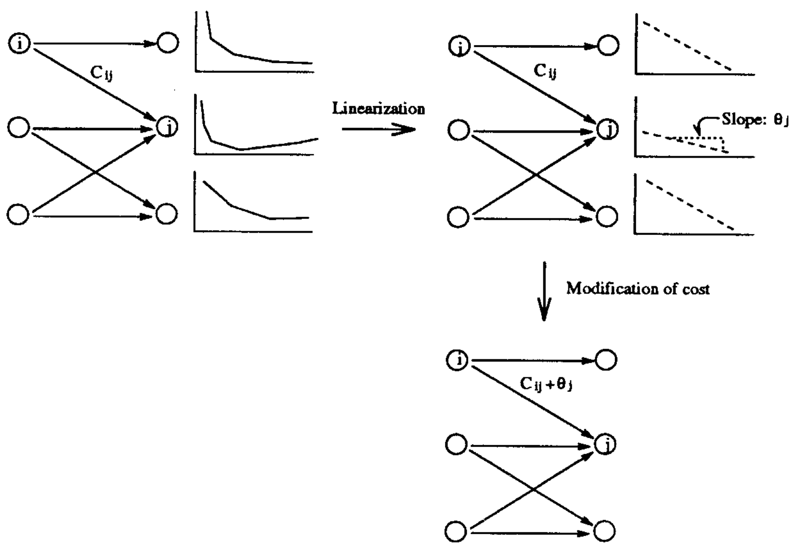


Fig. 18. The successive linear approximation procedure.

have a problem for time period $P-1$ that looks just like time period P, and the process can be repeated.

We next demonstrate these steps in more detail. Consider the case of $t = P$, where $\bar{Q}_{P+1}(S_{P+1}) \equiv 0$. Time period P becomes a one-sided transportation problem with random arc capacities. The recourse problem for stage P is then separable in S_P allowing us to write:

$$Q_P(S_P, \omega_P) = \sum_i Q_{iP}(S_{iP}, \omega_P)$$

where

$$Q_{iP}(S_{iP}, \omega_{iP}) = \min \sum_j c_{ij} x_{ijP}$$

subject to

$$\sum_j x_{ijP} = S_{iP}$$

$$x_{ijP} \le \xi_{ij}(\omega_P)$$

For a given realization ω_P the optimal solution is to greedily assign flow to the lowest cost links. Assume the links are ordered:

$$c_{i1} \le c_{i2} \le \ldots \le c_{iL}$$

where L denotes the number of links emanating from node i. Let $s = S_{iP}$ be the total flow that must be moved over these L links (we assume that the Lth link is unbounded to ensure feasibility). Let

$$\begin{aligned} I_{ijt}(k) &= \begin{cases} 1 & \text{if the } k\text{th unit of flow is assigned to link } (i,j) \text{ in period } t \\ 0 & \text{otherwise} \end{cases} \\ d_{ijt}(k) &= \text{Prob}\,[I_{ijt}(k) = 1] \end{aligned}$$

$d_{ijt}(k)$ is referred to as a *routing probability*. Let

$$U_{imt} = \sum_{j=1}^{m} \xi_{ijt}$$

be the cumulative capacity of the first m links. Then it is easy to verify that the event $[I_{ijt}(k) = 1]$ is equivalent to the joint event $(U_{i,j-1,t} < k \cap U_{ijt} \ge k)$. Thus

$$\begin{aligned} d_{ijt}(k) &= \text{Prob}\,[U_{i,j-1,t} < k \cap Ut_{ijt} \ge k] \\ &= \text{Prob}\,[U_{i,j-1,t} < k] + \text{Prob}\,[U_{ijt} \ge k] - \\ &\quad - \text{Prob}\,[U_{i,j-1,t} < k \cup U_{ijt} \ge k] \end{aligned} \tag{180}$$

The events $(U_{i,j-1,t} < k)$ and $(U_{ijt} \ge k)$ are collectively exhaustive (since $U_{i,j-1,t} \ge k$ implies $U_{ijt} \ge k$), and hence the last term of (180) is equal to 1.0:

$$\begin{aligned} d_{ijt}(k) &= \text{Prob}\,[U_{i,j-1,t} < k] + \text{Prob}\,[U_{ijt} \ge k] - 1 \\ &= \text{Prob}\,[U_{i,j-1,t} < k] - \text{Prob}\,[U_{ijt} < k] \end{aligned} \tag{181}$$

If the random variables ξ_{ijt} are independent, then the distribution of U_{ijt} is easy to compute (this is especially true if ξ_{ijt} follows a Poisson distribution). Thus the routing probabilities can be calculated relatively easily, even for large networks.

Now let

$$\begin{aligned} v_{iP}(k) &= \text{expected value of the } k\text{th unit of flow} \\ &= \sum_j d_{ijP}(k)c_{ij} \end{aligned}$$

For integer s we can express the recourse function $Q_{iP}(S_{iP})$ as:

$$Q_{iP}(s = S_{iP}) = \sum_{k=1}^{s} v_{iP}(k)$$

The importance of this result is that we have found the expected recourse function exactly. The challenge is to exploit this result for the remaining stages. Consider next the problem encountered with stage $t = P - 1$.

Now the recourse function $\bar{Q}_{P-1}(S_{P-1})$ is no longer separable in the vector $S(N-1)$. The nonlinear recourse function $\bar{Q}_P(S_P)$ couples the routing of flow out of different nodes in stage $P-1$. To circumvent this problem, replace $\bar{Q}_{iP}(s)$ with a linear approximation of the form:

$$\hat{Q}_{iP}(s) = a_{iP} + b_{iP}s$$

Since $\bar{Q}_{iP}(s)$ is convex, an appropriate choice of a_{iP} and b_{iP} gives a linear *support* of $\bar{Q}_{iP}(s)$, implying that $\hat{Q}_{iP}(s)$ can be constructed as a lower bound. However, the constant a_{iP} does not affect the optimal solution of stage $P-1$. Substituting $\hat{Q}_{iP}(s)$ into the expression for $\bar{Q}_{P-1}(S_{P-1})$ gives

$$\begin{aligned} \bar{Q}_{P-1}(S_{P-1}, \omega_{P-1}) &= \min \sum_i \sum_j c_{ij} x_{ij,P-1} + \sum_j [a_{jP} + b_{jP} S_{jP}] \\ &= \min \sum_i \sum_j c_{ij}\ x_{ij,P-1} + \sum_j \left[a_{jP} + b_{jP} \sum_i x_{ij,N-1}\right] \\ &= \min \sum_i \sum_j (c_{ij} + b_{jP}) x_{ij,N-1} + \sum_j a_{jP} \end{aligned}$$

This problem has the same structure as $Q_P(S_P, \omega_P)$ with the exception that the link costs c_{ij} are replaced by $c_{ij} + b_{jP}$. Thus, this recourse function is also separable, and can be calculated in the same manner we used to find $\bar{Q}_P(S_P)$.

This procedure can be applied recursively back to time period 1, with the last linearization occurring in time period 2. This leaves us with a piecewise linear, separable, convex approximation of $\bar{Q}_1(S_1)$. If the first stage is a network problem, then we are able to solve the original stochastic program (approximately) as a pure network with the form identical to that given in Figure 17.

The SLAP methodology easily handles P-stage stochastic transportation problems. Its novelty lies in the way it approximates the recourse function, thereby avoiding the explosion that occurs with dynamic programming or the conditioning inherent in multi-stage scenario methods. The successive linearizations, however, do introduce errors, and informal experiments have suggested that it may actually produce worse results as the number of stages is increased past a certain limit. On the other hand, it significantly outperforms the best competing heuristics (see Section 10.1).

9.4. The successive convex approximation method

The SLAP methodology linearizes the recourse function for period $t + 1$ to induce separability of the recourse function for period t. The method then takes advantage of the special structure of the optimization problem for a single node to find the expected recourse problem exactly. As a result, the methodology is highly tuned to a very specific type of stochastic network.

In this section, we use decomposition methods to develop a more accurate approximation of the expected recourse function. The approach is presented in three steps. The first result we need is the ability to find the expected recourse function over a tree with random arc capacities, given in Section 9.4.1. The method generalizes the technique used in Section 9.3 for single level trees. Next, Section 9.4.2 shows how dynamic, acyclic networks can be decomposed into stochastic trees using Lagrangian relaxation. This result allows us to approximate the recourse function for two-stage transshipment networks. Finally, Section 9.4.3 describes a backward recursion similar to SLAP which provides an approximation to multistage networks with random arc capacities.

9.4.1. Trees with random arc capacities

Powell & Cheung [1991] introduce an efficient method for finding the expected recourse function for a single stage optimization problem that has the structure of an n-level tree with random arc capacities, depicted in Figure 19. Let ξ

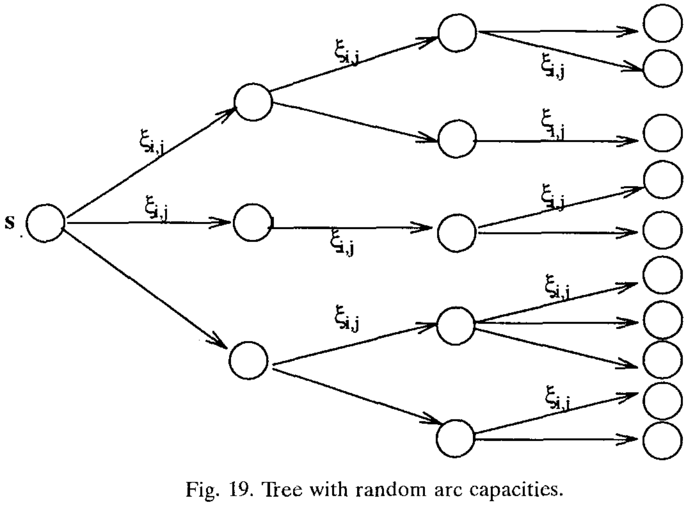


Fig. 19. Tree with random arc capacities.

represent the vector of (independent) arc capacities with a single realization $\xi(\omega)$. Let

r = index of the root node.

s = supply of flow entering the tree at node r.

$x(\omega)$ = vector of link flows conditioned on outcome ω.

$\mathcal{N}_e$ = set of ending nodes (where flow may leave the network).

$\mathcal{N}_t$ = set of pure transshipment nodes in the network ($\mathcal{N}_e \cap \mathcal{N}_t = \phi$).

The expected recourse function is given by

$$\bar{Q}(s) = E_\omega[Q(s,\omega)]$$

where the conditional recourse function is defined by :

$$Q(s,\omega) = \min_{x(\omega)} c^T x(\omega) \tag{182}$$

subject to

$$\sum_{j\in\mathcal{N}} x_{rj}(\omega) = s \tag{183}$$

$$\sum_{i\in\mathcal{N}} x_{ij}(\omega) - \sum_{k\in\mathcal{N}} x_{jk}(\omega) \begin{cases} = 0 \ \forall\, j \in N_t \\ \geq 0 \ \forall\, j \in N_e \end{cases} \tag{184}$$

$$x_{ij}(\omega) \leq \xi_{ij}(\omega) \tag{185}$$

$$x_{ij}(\omega) \geq 0 \tag{186}$$

Equation (184) allows flow to exit from any node in the set $\mathcal{N}_e$. Constraint (183) forces s units of flow to enter the network. Constraint (186) allows a random capacity on any link of the network, but we assume there exists at least one unbounded path from r to some node $j \in \mathcal{N}_e$ to ensure feasibility. There is no a priori restriction on the number of links in a path through the tree.

The solution of (182) is discussed in depth in Powell & Cheung [1991]. Here we present only the algorithm, which involves a single backward pass through the tree. To set up the algorithm, let:

n = path index.

c_n = cost of the nth path.

$l(n)$ = last node on path n.

$p(j)$ = predecessor of node j in the tree.

T^n = subtree consisting of all the links in the first n paths

$\mathcal{N}^n$ = set of nodes in T^n.

$\hat{\xi}^n_{ij}(\omega)$ = maximum possible flow on link $(i, j) \in T_n$.

$\zeta^n_i(\omega)$ = maximum possible flow that can move through node i in the graph T_n.

$Z^n(\omega)$ = total capacity of the first n paths.

The random variables $\hat{\xi}_{ij}^n(\omega)$, $\zeta_i^n(\omega)$ and $Z^n(n)$ are defined in the algorithm below.

Expected tree recourse algorithm

Step 1. Set $n = 0$, $\hat{\xi}_{ij}^0 = 0$, $S_i = 0$.
Step 2. Set $n = n + 1$, $j = l(n)$. Here, we augment the tree T^n with the $n + 1^{st}$ path to create the subtree T^{n+1}.
Step 3. Set $i = p(j)$.
Compute the distribution of $\hat{\xi}_{ij}^n$, defined by (for $j \in \mathcal{N}^n$):

$$\hat{\xi}_{ij}^n = \begin{cases} \xi_{ij} & \text{if } j = l(n) \\ \min\{\hat{\xi}_{ij}^n, \zeta_j^n\} & \text{otherwise} \end{cases}$$

Compute the distribution of ζ_i^n by:

$$\zeta_i^n = \sum_{k \in \mathcal{N}^n} \hat{\xi}_{ik}^n$$

Step 4. Set $j = i$ and return to step 3 until $i = r$.
Step 5. Let $Z^n = \zeta_r^n$, and go to step 2 until all paths are exhausted or we reach an uncapacitated path.

Steps 2 and 3 are the main computational steps, since the distributions of $\hat{\xi}_{ij}^n$ and ζ_i^n must be calculated numerically. Note that the random variables $\hat{\xi}_{ik}^n$ are independent for $k \in \mathcal{N}$, and ζ_j^n and $\hat{\xi}_{ij}^n$ are independent. When the algorithm is completed, we assume we have computed the probability mass function of the residual path capacities Z^n, from which we can obtain the cumulative distribution. Let

$d_{in}(k)$ = probability the kth unit of flow is routed over the nth path.

Clearly, $d_{in}(k)$ is similar to the routing probabilities we introduced for the SLAP algorithm, so it should not be too surprising that we calculate them using:

$$d_{in}(k) = \text{Prob}\,[Z^n < k] - \text{Prob}\,[Z^{n-1} < k]$$

which is analogous to equation (181). The distributions of Z^{n-1} and Z^n are calculated numerically using the tree recourse algorithm. Note that even if the arc capacities are independent Poisson random variables, the random variables ξ_i^n, and hence Z^n, do not have any special form (as a result of the minimum operator in step 3).

An interesting side benefit of the tree recourse algorithm is that we are equally capable of solving a single stage, n-level tree or an n-stage stochastic program composed of single level trees. In fact, Powell & Cheung [1992] show that the n-stage formulation is actually easier to solve for large problems. To see this, consider the tth stage of an n-stage tree. For $t > 1$, this stage will consist of a *forest* of two level trees. The links in stage t form a single level network with random arc capacities. The links for stage $t + 1$ represent the piecewise linear

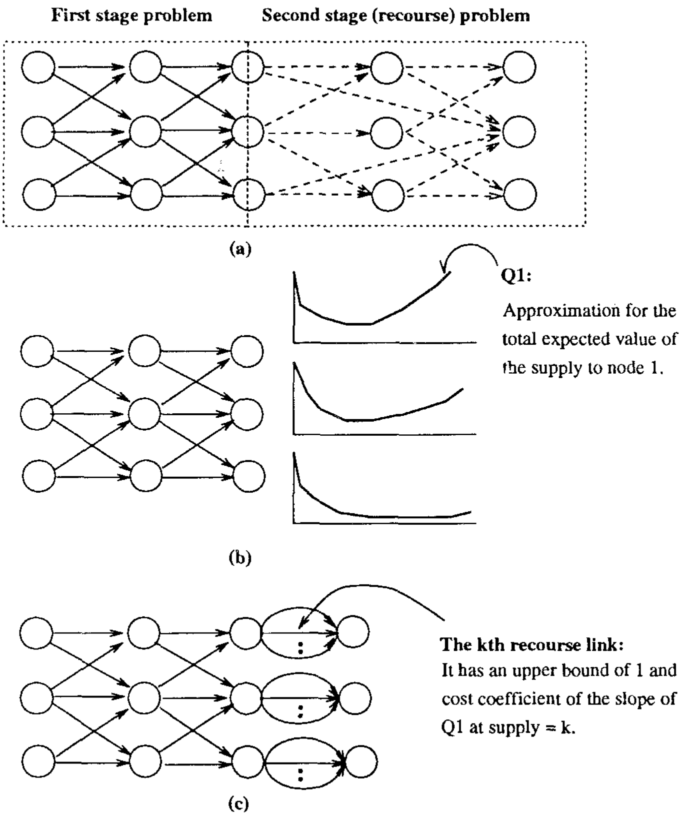


Fig. 20. Second stage stochastic network.

recourse function $\bar{Q}^{t+1}(s^{t+1})$ which is separable as a result of the tree structure (these links will each have an upper bound of 1 or ∞). We now have to apply the tree recourse algorithm to each subtree to find expected recourse functions for period t, after which we repeat the process for stage $t-1$. At each point, we are working with relatively small trees.

9.4.2. Network recourse decomposition

Powell & Cheung [1992] demonstrate how the tree recourse concept can be used to approximate recourse functions for general (multistage) dynamic networks with random arc capacities. Consider the second stage of a two-stage stochastic dynamic network as illustrated in Figure 20. Assume flow enters the network through nodes 1,2 and 3. Treat the flow on link (i, j) that originated at node k as a separate commodity. If we denote this flow as x_{ij}^k (suppressing the stage index), then of course

$$x_{ij} = \sum_k x_{ij}^k \tag{187}$$

This problem must be solved subject to random upper bounds:

$$x_{ij} \leq \xi_{ij} \tag{188}$$

Using the multicommodity formulation, (188) becomes a (random) bundle constraint, which destroys the tree structure. Assume instead that we relax (188) and add it to the objective function with penalty $\lambda_{ij}(\omega)$:

$$\min \sum_k \sum_{ij} c_{ij} x_{ij}^k(\omega) + \sum_{ij} \lambda_{ij}(\omega)\Big(\sum_k x_{ij}^k(\omega) - \xi_{ij}(\omega)\Big)$$

and instead enforce the constraint:

$$x_{ij}^k(\omega) \leq \xi_{ij}(\omega)$$

Now the problem decomposes by commodity k, but this does not guarantee that each subproblem forms a tree, since the flows that originate at a given node k can still form a cycle (since the network itself is not a tree). Powell and Cheung [1992] introduce the concept of the extended graph which forms a tree out of a subset of the paths that originate at node k. When two paths form a cycle, the joining link is split into two links, which is effectively another level of decomposition. The conditional Lagrange multiplier $\lambda(\omega)$ can be approximated by a constant λ, which can be optimized using subgradient methods. Since we are interested only in an approximation of the expected recourse function, and not in the second stage solution itself, the multipliers λ do not have to be very precise, and it is not that important if the bundle constraints in the second stage are violated. Numerical work in Powell and Cheung [1992] shows that the Lagrangian relaxation closely approximates the exact expected recourse function (calculated using Monte Carlo simulation) after ten iterations of the subgradient optimization algorithm.

9.4.3. The SCAM algorithm for multistage networks

The network recourse decomposition method can be applied to multistage networks in the following way. Consider the three-stage network illustrated in Figure 21, with stages $t-1, t$ and $t+1$. We first use the network recourse decomposition method to develop an approximate, expected recourse function for stage $t+1$, producing the functions in Figure 21b. (If the network is the simple transportation problem shown in the figure, then stage $t+1$ is separable, and the resulting expected recourse function is exact). These convex, piecewise linear functions can be represented as a series of links (the recourse links) out of each node in stage $t+1$, as shown in Figure 21c.

The next step is to find the expected recourse function for stage t. This is a transshipment network, where flows from different nodes in stage t must move over the same recourse links representing stage $t+1$. We apply the network recourse decomposition method to these links, effectively unbundling the flows from different nodes within stage t, as illustrated in Figure 21d. The result is a set of trees, one for each node in stage 1, which exhibit stochastic upper bounds on the links in stage t, plus the deterministic recourse links which represent stage $t+1$. Lagrange multipliers are introduced for each recourse link, and these

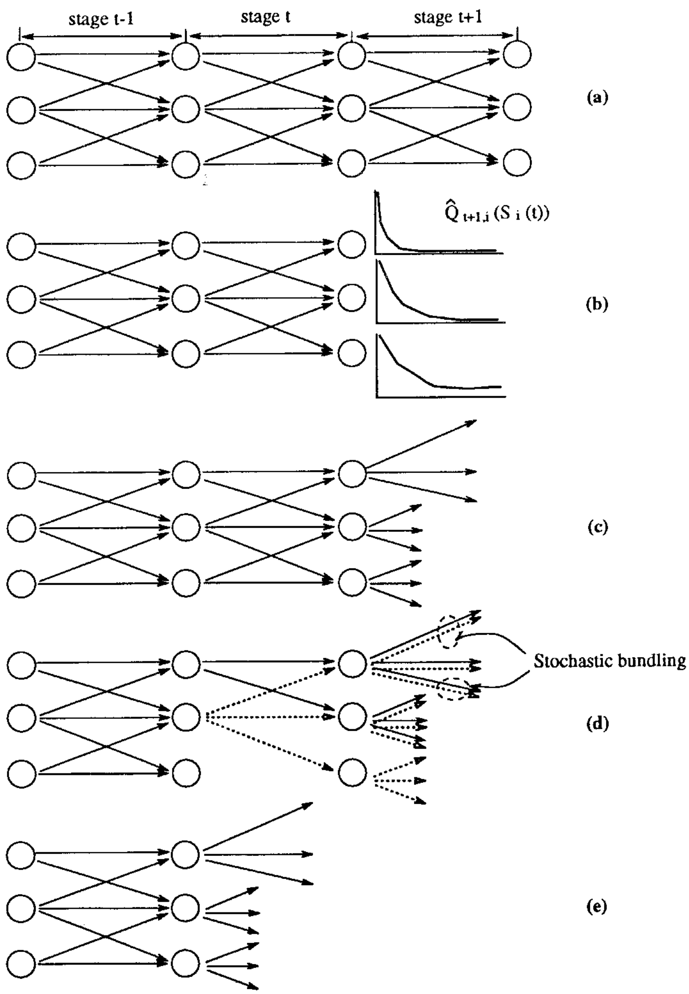


Fig. 21. The successive convex approximation methodology.

are optimized using subgradient optimization. This in turn results in a separable approximation of the recourse function for stage t, which is again modeled as a set of recourse links for each node, as shown in Figure 21e.

Thus, the SCAM methodology uses a backward recursion where for each stage, we find an approximation of the expected recourse function for a particular stage t which includes the recourse links for stage $t+1$. In contrast with the SLAP methodology, which linearizes the recourse function for stage $t+1$, we are able to use a convex approximation. Although the computational costs for SCAM are somewhat higher, it consistently outperforms the SLAP approach in rolling horizon experiments by one to five percent. Both of these methods, in turn, consistently outperform deterministic models.

10. Evaluating dynamic models

As this chapter has illustrated, there are often a number of ways to formulate and solve dynamic models. A rich set of approximations have evolved to handle forecasting uncertainties and model truncation in a tractable way. However, this leaves the problem of evaluating the quality of the solution that is provided. Static models, regardless of their complexity, typically enjoy the feature of a single, well-defined cost function that can be evaluated even if it is extremely difficult to find the minimum. Two heuristics can be compared by simply comparing the objective function values produced by each method. By contrast, the objective function of a dynamic model is often of little intrinsic interest, since the more relevant measure of performance is the costs produced over time through repeated applications of the model.

Dynamic models are generally applied on a rolling horizon basis, producing a stream of costs that must be evaluated over time. Two questions arise. First, how well are we optimizing the dynamic model over the chosen planning horizon? And second, how well do the recommendations of the dynamic model perform when evaluated over time? Four broad areas of research provide insight into these issues, although overall, relatively little progress has been made for evaluating the results of dynamic network models of realistic size. The first area of research addresses the first question by deriving upper and lower bounds on the expected recourse function, which helps answer the question of how well we are optimizing the dynamic model. The next three lines of investigation consider the quality of the recommendations when evaluated over time. These include the use of simulation experiments; the use of posterior bounds; and rigorous bounds for rolling horizon results. In this section, we briefly review the basic ideas behind each approach, and summarize the state of the art of research in each area.

10.1. Rolling horizon simulations

Consider a rolling horizon procedure $\mathcal{P}$, which might consist of an n-stage stochastic program, or an n-period deterministic dynamic network model. When applied in time period τ, the procedure will span time periods $t = \tau$ up to $t = \tau + P$. The procedure will operate on a *realization* of the actual data in time period τ, denoted by outcome ω_τ, and *forecasted* data for periods $\tau+1$ up to $\tau+n$. Let the *actions* recommended by $\mathcal{P}$ in time period τ be denoted by $x_\tau(\mathcal{P}, \omega_\tau, S_\tau)$ where S_τ is the state of the system at time τ. S_τ is assumed to capture the history of the process up to time τ. A *simulation* is comprised of the sequence of outcomes and actions $\omega_1, x_1(\mathcal{P}, \omega_1, S_1), \omega_2, x_2(\mathcal{P}, \omega_2, S_2), \ldots, \omega_N, x_N(\mathcal{P}, \omega_n, S_N)$. This simulation reflects a single realization of the complete set of outcomes $\omega = \{\omega_1, \ldots, \omega_N\}$, and yields total costs:

$$C(\mathcal{P}, \omega, N) = \sum_{\tau=1}^{N} c^T x_\tau(\mathcal{P}, \omega_\tau, S_\tau)$$

Of course, $C(\mathcal{P}, \omega, N)$ is a random variable. We would like to compare two rolling horizon procedures $\mathcal{P}_1$ and $\mathcal{P}_2$ on the basis of these costs, but this requires developing an understanding of the distribution of $C(\mathcal{P}, \omega, N)$. We can reduce the variance of $C(\mathcal{P}, \omega, N)$ by increasing the simulation length or by taking a sample of n simulations, and calculating a sample average cost:

$$\bar{C}(\mathcal{P}, N) = \frac{1}{n} \sum_{i=1}^{n} C(\mathcal{P}, \omega_i, N)$$

This reduces, but does not eliminate uncertainty in the estimate of the costs.

Testing a procedure $\mathcal{P}$ in a rolling horizon experiment is the most direct method for evaluating a model. Many dynamic models are fairly robust and yield good results despite fairly coarse approximations of future events. However, they can be computationally very burdensome, and while they can provide an indication if one procedure is better than another, they provide little absolute measure of the effectiveness of either procedure (perhaps they are both poor). Finally, if a procedure yields poor results, rolling horizon simulations yield few insights with respect to how the procedure should be modified.

10.2. Bounds for stochastic programs with network recourse

In sharp contrast with the other lines of investigation, the literature on bounds for stochastic programs is extensive. The basic program considered is as follows. Consider the two-stage stochastic program:

$$\min_{x} F(x) = c^T x + \bar{Q}(x) \tag{189}$$

where $\bar{Q}(x)$ is the usual expected recourse function given by

$$\bar{Q}(x) = E_{\omega}\{Q(x, \omega)\} \tag{190}$$

where $Q(x, \omega)$ is an imbedded optimization problem. In Section 7, we considered methods for optimizing (189) which recognizes the fact that we cannot write $\bar{Q}(x)$ analytically. However, for practical problems, we cannot even calculate $\bar{Q}(x)$ for a given value of x, since the expectation in (190) is generally too difficult. As a result, given two solutions $x^{(1)}$ and $x^{(2)}$, that may be produced by two different heuristics, we are unable to say, precisely, whether $F(x^{(1)}) > F(x^{(2)})$ or $F(x^{(1)}) < F(x^{(2)})$. Note that we are not trying to optimize $F(x)$ at this point, but rather are simply trying to estimate the objective function of the dynamic model.

To address this problem, a large literature has evolved over the years on the problem of developing rigorous bounds on the expected recourse function. An excellent review of this work is given in Kall & Stoyan [1982] and Birge & Wets [1986]. The basic approach to finding a lower bound of the recourse function has been to apply Jensen's inequality which states that, for a convex function $F(\xi)$ of a random variable ξ,

$$E_{\xi}[F(\xi)] \geq F(\bar{\xi})$$

Kall & Stoyan [1982] show that Jensen's inequality can be applied to stochastic programs with recourse as long as a) the recourse function is convex in the second stage actions x_1 and the random variable ξ, for a given first stage action x_0, and b) the random variable ξ does not appear in the objective function. The appeal of Jensen's inequality is that it is so easy to apply, since it involves solving the recourse function for a single realization of the random variables. Furthermore, it is relatively straightforward to apply Jensen's inequality to multistage problems. For example, Frantzeskakis and Powell [1989b] show that when applied to an N-stage network with random arc capacities, Jensen's inequality involves solving an N-period dynamic network with upper bounds equal to the expected arc capacities.

A novel approach for bounding the recourse function for networks is proposed by Wallace [1987]. In contrast with methods that approximate the probability measure (such as the Edmundson–Madansky bound) or the recourse function (the ray approximation of Birge and Wets), Wallace's approach approximates the problem itself. Consider the second stage of a two stage stochastic network with recourse with random supplies α_i, random demands β_j, and random arc capacities ξ_{ij}. Wallace's bound takes the following form:

$$Q(\alpha, \beta, \xi) \leq P(\alpha, \beta, \xi) = \rho + \sum_i f_i(\alpha_i) + \sum_j g_j(\beta_j) + h(\xi)$$

These calculations assume a fixed first stage vector x_0. $P(\alpha, \beta, \xi)$ is an upper bound on the recourse function which is constructed to be separable in the random supplies, demands and arc capacities. ρ is a constant found using $\rho = Q(\alpha^0, \beta^0, \xi^0)$ where $(\bar{\alpha}, \bar{\beta}, \bar{\xi})$ represents a base point. We might use, for example, $(\bar{\alpha}, \bar{\beta}, \bar{\xi}) = (0, 0, 0)$ or $(E\{\alpha\}, E\{\beta\}, E\{\xi\})$. Let $\bar{x}^1$ be the flows resulting from this base point (thus, $\rho = c^T \bar{x}^1$). Now let z^1_α, z^1_β and z^1_ξ be random perturbations from $\bar{x}^1$ resulting from specific realizations of α, β and ξ. Assume $(\bar{\alpha}, \bar{\beta}, \bar{\xi}) = (0, 0, 0)$. Increasing realizations of α_i are captured by placing flow on successively longer (more costly) paths from node i to the root (or slack) node of the graph. Realizations of the arc capacities are handled by putting flow on a predetermined set of cycles which are ordered from least to most cost. As much flow as possible is put on the highest ranked/least cost cycle. Flow that cannot fit on the highest ranked cycle spills to lower ranked cycles. To simplify taking expectations, Wallace describes a method for splitting links that are shared by more than one cycle (since the amount of flow that can fit on one cycle is random, the amount of unused capacity that might be available for another cycle that shares the same link is also random with a potentially complex probabilistic structure). This splitting process produces an upper bound since it restricts flow on one cycle from using capacity that may have been allocated to a higher ranked cycle (but which was not fully utilized). This concept is extended in Frantzeskakis and Powell [1989a] which presents much tighter bounds for the same class of problems.

Frantzeskakis and Powell [1989b] use the SLAP concept of Section 9.3 to develop rigorous upper and lower bounds for large-scale dynamic networks with random arc capacities. Lower bounds are constructed by finding a linear support of the recourse function around some point $\hat{x}$. An upper bound is established by drawing a line between the origin and the recourse function evaluated at the maximum feasible value of x. Both of these upper and lower bounds represent linear approximations that can be used within the SLAP methodology. Experimental testing demonstrated that the SLAP upper and lower bound, as well as Jensen's bound, can be calculated very efficiently. However, the bounds are not very close. In one experiment, the upper and lower bounds were found to be as much as 50% higher and lower than the best available estimate of the recourse function (obtained using Monte Carlo methods). More promising is the network recourse decomposition method in Section 9.4.2 where subgradient optimization is used to refine the Lagrange multipliers, but this has not been tested.

10.3. Posterior bounds for rolling horizon procedures

One approach for obtaining an absolute measure of the effectiveness of a RHP is to use a device known as a posterior bound. A RHP must choose a set of actions x_τ at time τ based on the state of the system S_τ and the outcome ω_τ, but without knowledge of future events. Assume now that we know the entire set of outcomes $\omega = \{\omega_1, \ldots, \omega_N\}$ over the entire simulation. Rather than solving a sequence of N minimization problems, we could solve a single minimization problem over all N periods. Let

$$x^{(p)}(\omega, N) = \left\{x_1^{(p)}(\omega, N), x_2^{(p)}(\omega, N), \ldots, x_N^{(p)}(\omega, N)\right\} \tag{191}$$

be the set of activities resulting from this global, posterior optimization with cost:

$$C^{(p)}(\omega, N) = \sum_{t=1}^{N} c^T x_t^{(p)}(\omega, N) \tag{192}$$

Since this posterior optimization can anticipate future events, it will produce lower overall costs, and thus:

$$C^{(p)}(\omega, N) \leq C(\mathcal{P}, \omega, N) \tag{193}$$

giving us our posterior bound on the costs for a simulation. Let $C(\mathcal{P}^*, \omega, N)$ be the optimal achievable (nonanticipatory) costs found from using the best possible rolling horizon procedure $\mathcal{P}^*$. By definition

$$C^{(p)}(\omega, N) \leq C(\mathcal{P}^*, \omega, N) \leq C(\mathcal{P}, \omega, N) \tag{194}$$

Of course, we will never know $C(\mathcal{P}^*, \omega, N)$. However, we conjecture that for large, real-world problems, $C^{(p)}(\omega, N)$ and $C(\mathcal{P}^*, \omega, N)$ may be surprisingly close (at least within 10%). If this is the case, the gap between the posterior bound $C^{\mathcal{P}}(\omega, N)$ and the results of a given rolling horizon procedure $\mathcal{P}$ may provide an indication of the absolute effectiveness of a particular dynamic model.

10.4. Error bounds for rolling horizon procedures

It would be especially desirable to have a rigorous bound on the error produced by a finite horizon RHP over the results produced by an optimal, infinite horizon procedure. Alden [1987] and Alden & Smith [1987] derive error bounds for RHP's and demonstrate their use in the context of a particular version of the dynamic vehicle allocation problem. Let $S_i(t)$ denote the number of vehicles in city i at time t, and let $S_t = \{\ldots, S_{it}, \ldots\}$represent the vector of vehicle supplies. S_t represents the state of the system at time t. Each day, a certain number of vehicles will move loaded from one region to the next, after which the remaining vehicles may be repositioned empty to handle excess supplies of vehicles in other regions. Let π_s denote a *policy* for repositioning vehicles given that the system is in state s, and let $\mathcal{P}(\pi)$ be the one-step transition matrix using policy π.

Alden & Smith [1987] derive error bounds for a finite-stage RHP in terms of the *Doeblin coefficient* of the one-step transition matrix. If Q is a stochastic matrix with element Q_{ij}, the Doeblin coefficient β_{D} is defined as [Seneta, 1981]:

$$\beta_{\mathrm{D}} = 1 - \sum_{j=1}^{n} \min_i Q_{ij} \tag{195}$$

If Q has at least one zero in each column, then $\beta_{\mathrm{D}} = 1$. Alternatively, if Q is a stable matrix (all rows equal) then $\beta_{\mathrm{D}} = 0$. β_{D} is a measure of the ergodicity of the transition matrix. Now define the *coupling coefficient* $\beta_{\mathcal{C}}$ as:

$$\beta_{\mathcal{C}} = 1 - \min_{\pi} \sum_{j=1}^{n} \min_i Q_{ij}(\pi)$$

$\beta_{\mathcal{C}}$ is the largest possible Doeblin coefficient over all policies π.

Finally, let:

α = discount factor per period.
P = length of the planning horizon in the RHP.
N = number of periods being simulated.
$\mathcal{C}$ = maximum cost of any state transition.
$\bar{\mathcal{C}}(\mathcal{P}^*, N) = E_\omega[\mathcal{C}(\mathcal{P}^*, \omega, N)]$.
$\bar{\mathcal{C}}(\mathcal{P}, N) = E_\omega[\mathcal{C}(\mathcal{P}, \omega, N)]$

The bound derived by Alden & Smith [1987] is given by:

$$\bar{\mathcal{C}}(\mathcal{P}, N) - \bar{\mathcal{C}}(\mathcal{P}^*, N) \leq \frac{(\alpha\beta_c)^T}{1-\alpha\beta_c}\left[1 + \frac{\alpha(1-\beta_c)(1-\alpha^{N-T-1})}{1-\alpha}\right]\mathcal{C} \tag{196}$$

Note that as long as $\beta_c > 0$, we have a valid bound even if there is no discounting ($\alpha = 1$).

The Alden–Smith bound is significant primarily because it is the only bound for rolling horizon procedures (actually it extends an earlier result by Bean & Smith [1984]) and because it captures the effect of ergodicity on the choice of planning horizons. The concept has also been applied to the dynamic vehicle allocation problem, and in fact presents a fresh perspective on this problem. The limitation of the result is that the bound is likely to be very weak. For large problems, the coupling coefficient β_c is likely to be quite large, since it will often be the case that some states cannot be reached from at least one other state (however, steps can be taken to reduce this problem). Also, we have the classic curse of dimensionality which produces an exponentially large number of states, making the calculation of β_c difficult. These issues notwithstanding, the result is an important milestone in this line of research.

Acknowledgement

This research was supported in part by grant DDM-9102134 from the National Science Foundation, and by grant AFOSR-F49620-93-1-0098 from the Air Force Office of Scientific Research.

References

Agnihothri, U.K.S., and P. Kubat (1982), Stochastic allocation rules, *Oper. Res.* 30(3), 545–555.

Ahuja, R., T. Magnanti, and J. Orlin (1992), *Network Flows: Theory, Algorithms and Applications*, Prentice Hall, New York.

Alden, J. (1987), *Error Bounds for Rolling Horizon Procedures*, Ph.D. dissertation, Dept. of Industrial and Operations Engineering, University of Michigan.

Alden, J., and R. Smith (1992), Rolling horizon procedures in nonhomogeneous Markov decision processes. *Oper. Res.* 40(Supp. 2), 183–194.

Andreatta, G. (1987), Shortest path models in stochastic networks, in: G. Andreatta, F. Mason, and P. Serafini, eds., *Stochastics in Combinatorial Optimization*, World Scientific, Singapore.

Andreatta, G., F. Ricaldone, and L. Romeo (1985), Exploring stochastic shortest path problems, in: *ATTI Giornate di Lavoro*, Tecnoprint, Bologna.

Andreatta, G., and G. Romanin-Jacur (1987), The flow management problem, *Transp. Sci.* 21(4), 249–253.

Andreatta, G., and L. Romeo (1988), Stochastic shortest paths with recourse, *Networks* 18, 193–204.

Arabeyre, J., J. Fearnley, F. Steiger, and W. Teather (1969), The airline crew scheduling problem: A survey, *Transp. Sci.* 3, 140–163.

Aronson, J. (1980), *Forward Linear Programming*, Ph.D. dissertation, Grad. School of Industrial Administration, Carnegie–Mellon University.

Aronson, J. (1989), A survey of dynamic network flows, in: P. Hammer, ed., *Ann. Oper. Res.*, pp. 1–66. J.C. Baltzer AG.

Aronson, J. (1990), Private communication.

Aronson, J., and B. Chen (1986), A forward network simplex algorithm for solving multiperiod network flow problems, *Nav. Res. Logistics Q.* 33(3), 445–467.

Aronson, J., and B. Chen (1989), A computational study of emperical decision horizons in infinite horizon multiperiod network flow models, Working paper 89-275, Dept. of Mgmt. Sci. and Information Tech., University of Georgia.

Aronson, J., and G.L. Thompson (1984), A survey on forward methods in mathematical programming. *Large Scale Systems* 7, 1–16.

Assad, A. (1987), Multicommodity network flows: A survey, *Networks* 8(1), 37–92.

Assad, A., E. Wasil, and G. Lilien (1992), *Excellence in Management Science: A Readings Book*, Prentice Hall, New York.

Avram, J., and D. Bertsimas (1992), The minimum spanning tree constant in geometrical probability and under the independent model; an unified approach, *Ann. Appl. Probab.* 2, 113–130.

Bagchi, B., and B. Nag (1991), Dynamic vehicle scheduling: An expert systems approach, *Int. J. Phys. Distrib. Logist. Manage.* 21(2), 10–18.

Bahl, H., L. Ritzman, and J. Gupta (1987), Determining lot sizes and resource requirements: A review, *Oper. Res.* 35(3), 329–345.

Ball, M., L. Bodin, and R. Dial (1983), A matching based heuristic for scheduling mass transit crews and vehicles, *Trans. Sci.* 17, 4–31.

Ball, M., and A. Roberts (1985), A graph partitioning approach to airline crew scheduling, *Trans. Sci* 19, 107–126.

Bartholdi, J., and L. Platzman (1982), An $O(n \log n)$ planar traveling salesman heuristic based on spacefilling curves, *Oper. Res. Lett.* 1, 121–125.

Bartholdi, J., L. Platzman, R. Collins, Lee, and W. Warden (1983), A minimal technology routing system for meals on wheels, *Interfaces* 13, 1–8.

Batta, R., O. Berman, S. Chiu, R. Larson, and A. Odoni (1990), Discrete location theory, in: P. Mirchandani and R. Francis, eds., *Discrete Location Theory*, Wiley, New York.

Beale, E. (1955), On minimizing a convex function subject to linear inequalities, *J. R. Stat. Soc.* B17, 173–184.

Beale, E., G. Dantzig, and R. Watson (1986), A first order approach to a class of multi-time period stochastic programming problems, *Math. Program. Study* 27, 103–117.

Beale, E., J. Forest, and C. Taylor (1980), Multi-time period stochastic programming, in: M. Dempster, ed., *Stochastic Programming*, Academic Press.

Bean, J.C., and R.L. Smith (1985), Optimal capacity expansion over an infinite horizon, *Manage. Sci.* 31(12), 1523–1532.

Bean, J.C., and R.L. Smith (1987), Conditions for the existence of planning horizons, *Math. Oper. Res.* 9(12), 391–401.

Beardwood, J., J. Halton, and J. Hammersley (1959), The shortest path through many points, *Proc. Camb. Phil. Soc.* 55, 299–327.

Bell, W., L. Dalberto, M. Fisher, A. Greenfield, R. Jaikumar, P. Kedia, R. Mack, and P. Prutzman (1983), Improving the distribution of industrial gases with an online computerized routing and scheduling optimizer, *Interfaces* 13, 4–23.

Bellmore, M., W. Ecklof, and G. Nemhauser (1969), A decomposable transshipment algorithm for a multiperiod transportation problem, *Nav. Res. Logist. Q.* 16, 517–524.

Bellmore, M., and R. Vemuganti (1973), On multicommodity maximal dynamic flows, *Oper. Res.* 21, 10–21.

Bensoussan, A., and M. Crouhy, and J. Proth (1983), *Mathematical Theory of Production Planning*, North-Holland, Amsterdam.

Berman, O. (1981), Dynamic repositioning of indistinguishable service units on transportation networks, *Transp. Sci.* 15(2), 115–136.

Berman, O., and B. LeBlanc (1984), Location–relocation of mobile facilities on a stochastic network, *Transp. Sci.* 18(4), 315–330.

Berman, O., and D. Simchi-Levi (1986), Minimum location of a traveling salesman, *Networks* 16, 329–354.

Berman, O., and D. Simchi-Levi (1988), Finding the optimal a priori tour and location of a traveling salesman with non-homogeneous customers, *Transp. Sci.* 22, 148–154.

Berman, O., and D. Simchi-Levi (1988), A heuristic algorithm for the traveling salesman location problem on networks, *Oper. Res.* 36, 478–484.

Bertsekas, D. (1979), Algorithms for nonlinear multicommodity network flow problems, in: A. Bensoussan and J. Lions, eds., *International Symposium on Systems Optimization and Analysis*, pp. 210–224, Springer-Verlag, Boston.

Bertsekas, D. (1987), *Dynamic Programming: Deterministic and Stochastic Models*, Prentice Hall.

Bertsekas, D., E. Gafni, and R. Gallagher (1984), Second derivative algorithms for minimum delay distributed routing in networks, *IEEE Trans. Commun.* COM-32, 911–919.

Bertsekas, D., and R. Gallager (1987), *Data Networks*, Prentice Hall.

Bertsekas, D., and J. Tsitsiklis (1991), An analysis of stochastic shortest path problems, *Math. Oper. Res.* 16, 580–595.

Bertsimas, D. (1988), *Probabilistic combinatorial optimization problems*, Technical Report 194, Operations Research Center, MIT.

Bertsimas, D. (1989), On probabilistic traveling salesman facility location problems, *Transp. Sci.* 23, 184–191.

Bertsimas, D. (1992), A vehicle routing problem with stochastic demand, *Oper. Res.* 40, pp. 574–585.

Bertsimas, D., and M. Grigni (1989), Worst-case examples for the spacefilling curve heuristic for the Euclidean traveling salesman problem, *Oper. Res. Lett.* 5, 241–244.

Bertsimas, D.J., P. Jaillet, and A.R. Odoni, (1990), A priori optimization, *Oper. Res.* 38, 1019–1033.

Bertsimas, D.J., and G. van Ryzin (1991), A stochastic and dynamic vehicle routing problem in the Euclidean plane, *Oper. Res.* 39, 601–615.

Bertsimas, D.J., and G. van Ryzin (1993a), Stochastic and dynamic vehicle routing problem in the Eucilidean plane with multiple capacitated vehicles, *Oper. Res.* 41, pp. 60–76.

Bertsimas, D.J., and G. van Ryzin (1993b), Stochastic and dynamic vehicle routing with general demand and interarrival time distributions, *Adv. Appl. Prob.* 25(4), 947–978.

Bes, C., and S. Sethi (1988), Concepts of forecast and decision horizons: Applications to dynamic stochastic optimization problems, *Math. Oper. Res.* 13(2), 295–310.

Bhaskaran, S., and S. Sethi (1988), Decision and forecast horizons in a stochastic environment: A survey, *Optimal Control Applic. Meth.* 8(1), 61–67.

Bhat, U. (1984), *Elements of Applied Stochastic Processes*, Wiley, New York.

Bielli, M., G. Calicchio, B. Micoletti, and S. Ricciardelli (1982), The air traffic flow control problem as an application of network theory, *Comput. Oper. Res.* 9(4), 265–278.

Birge, J., and R. Wets (1986), Designing approximation schemes for stochastic optimization problems, in particular for stochastic programs with recourse, *Math. Program. Study* 27, 131–149.

Birge, J.R., and J.K. Ho (1987), The stochastic dynamic traffic assignment problem, Technical Report 87-16, Dept. of Industrial and Operational Engineering, University of Michigan.

Blum, J. (1954), Approximation methods which converge with probability one, *Ann. Math. Stat.* 25, 382–386.

Bodin, L., B. Golden, A. Assad, and M. Ball (1983), Routing and scheduling of vehicles and crews, the state of the art, *Comput. Oper. Res.* 10, 69–211.

Bowman, E. (1956), Production scheduling by the transportation method of linear programming, *Oper. Res.* 4, 100–103.

Bradley, G. (1975), Survey of deterministic networks, *AIIE Transactions* 7(3), 222–234.

Brown, G., and G. Graves (1981), Real-time dispatch of petroleum tank trucks, *Manage. Sci.* 27, 19–32.

Brown, G., G. Graves, and D. Ronen (1987), Scheduling ocean transportation of crude oil, *Manage. Sci.* 33, 335–346.

Burness, R., and J. White (1976), The traveling salesman location problem, *Transp. Sci.* 10, 348–360.

Busacker, R., and P. Gowen, (1961), A procedure for determining a family of minimal cost network flow patterns, Oper. Res. Off., Tech. paper 15.

Carey, M. (1986), A constraint qualification for a dynamic traffic assignment model, *Transp. Sci.* 20(1), 55–58.

Carey, M. (1987), Optimal time-varying flows on congested networks, *Oper. Res.* 35(1), 58–69.

Chalmet, L., R. Francis, and P. Saunders (1982), Network models for building evacuation, *Manage. Sci.* 28, 86–105.

Charnes, A., and W. Cooper (1959), Chance constrained programming, *Manage. Sci.* 5(1), 73–79.

Chih, K. (1986), A real time dynamic optimal freight car management simulation model of the multiple railroad, multicommodity, temporal spatial network flow broblem, Ph.D. Dissertation, Dept. of Civil Engng. and Oper. Res., Princeton University.

Chih, K., M. Hornung, M. Rothenberg, and A. Kornhauser (1990), Implementation of a real time locomotive distribution system, in: T. Murthy, R. Rivier, G. List, and J. Mikolaj, eds., *Computer Applications in Railway Planning and Management*, pp. 39–50, Computational Mechanics Publications.

Choi, W., H. Hamacher, and S. Tufekci (1988), Modeling of building evacuation problems by network flows with side constraints, *Eur. J. Oper. Res.* 35, 98–110.

Chong, E. (1991), *On-line Stochastic Optimization of Queueing Systems*, Ph.D. dissertation, Dept. of Electrical Engng., Princeton University.

Christofides, N., and P. Brooker (1967), Optimal expansion of an existing network, *Math. Program.* 6, 197–211.

Colbourn, C. (1987), *The Combinatorics of Network Reliability*, Oxford University Press, New York.

Cooke, K., and E. Halsey (1966), The shortest route through a network with time-dependent internodal transit times, *J. Math. Anal. Appl.* 14, 493–498.

Cooper, L., and L. Leblanc (1977), Stochastic transportation problems and other network related convex problems, *Nav. Res. Logist. Q.* 24, 327–337.

Crainic, T., J. Ferland, and J. Rousseau (1984), A tactical planning model for rail freight transportation, *Transp. Sci.* 18, 165–184.

Crainic, T., M. Gendreau, and P. Dejax (1993), Dynamic stochastic models for the allocation of empty containers, *Oper. Res.* 41, 102–126.

Crainic, T., and J. Rousseau (1987), The column generation principle and the airline crew scheduling, *INFOR* 25(2), 136–151.

Crainic, T., and J. Roy (1988), Or tools for the tactical planning of freight transportation, *Eur. J. Oper. Res.* 33, pp. 290–297.

D'Ans, G., and D. Gazis (1976), Optimal control of oversaturated store and forward transportation networks, *Transp. Sci.* 10, 1–19.

Dantzig, G. (1955), Linear programming under uncertainty, *Manage. Sci.* 1, 197–206.

Dantzig, G., and A. Ferguson (1956), The allocation of aircrafts to routes: An example of linear programming under uncertain demand, *Manage. Sci.* 3, 45–73.

Dantzig, G., and D. Fulkerson (1954), Minimizing the number of tankers to meet a fixed schedule, *Nav. Res. Logist. Q.* 1, 217–222.

Dempster, M., M. Fisher, L. Jansen, B. Lageweg, J. Lenstra, and A.R. Kan (1981), Analytical evaluation of hierarchical planning systems, *Oper. Res.* 29, 707–716.

Derman, C. (1970), *Finite State Markovian Decision Processes*, Academic Press, New York.

Desrosiers., J. (1990), private communication.

Desrosiers, J., F. Soumis, and M. Desrochers (1984), Routing with time windows by column generation, *Networks* 14, 545–565.

Doulliez, P., and M.R. Rao (1984), Optimal network capacity planning: A shortest path scheme, *Oper. Res.* 23, 811–818.

Dreyfus, S. (1969), An appraisal of some shortest-path algorithms, *Oper. Res.* 17, 395–412.

Driscoll, J., H. Gabow, R. Shraiman, and R. Tarjan (1988), Relaxed heaps: An alternative to Fibonacci heaps with applications to parallel computation, *Commun. ACM* 31, 1343–1354.

Dror, M., M. Ball, and B. Golden (1985), A computational comparison of algorithms for the inventory routing problem, *Ann. Oper. Res.* 4, 3–23.

Dror, M., and P. Trudeau, (1986), Stochastic vehicle routing with modified savings algorithm, *Eur. J. Oper. Res.* 23, 228–235.

Dror, M., G. Laporte, and P. Trudeau (1989), Vehicle routing with stochastic demands: Properties and solution frameworks, *Transp. Sci.* 23, 166–176.

Eilon, B., C. Watson-Gandy, and N. Christofides (1971), *Distribution Management*, Hafner.

Ellis, R., and R. Rishel (1974), An application of stochastic optimal control theory to optimal rescheduling of airplanes, *IEEE Trans. Autom. Control* 139–147.

Erickson. R.E., C.L. Monma and A.F. Veinott (1987), Send-and-split method for minimum-concave-cost network flows, *Math. Oper. Res.*, 12(4), 634–664.

Erlenkotter, D. (1969), *Preinvestment Planning for Capacity Expansion: A Multi-Location Dynamic Model*, Ph.D. Dissertation, Graduate School of Business, Stanford University, Stanford, CA.

Erlenkotter, D. (1975), Capacity planning for large multi-location systems: Approximate and incomplete dynamic programming approaches, *Manage. Sci.* 22, 274–285.

Erlenkotter, D. (1981), A comparative study of approaches to dynamic location problems, *Eur. J. Oper. Res.* 6, 133–143.

Ermoliev, Y. (1969), On the stochastic quasigradient method and stochastic quasi-feyer sequences, *Kibernetika* 2.

Ermoliev, Y. (1988), Stochastic quasigradient methods, in: Y. Ermoliev and R. Wets, eds., *Numerical Methods in Stochastic Programming*, Springer-Verlag.

Ermoliev, Y., T. Krivets, and V. Petukhov (1976), Planning of shipping empty seaborne containers, *Cybernetics* 12, 664.

Farvolden, J., and W. Powell, (1991), Subgradient methods for the service network design problem, Dept. of Industrial Engng., University of Toronto.

Farvolden, J., W. Powell and I. Lustig (1993), A primal partitioning solution for the arc-chain formulation of a multicommodity network flow problem, *Oper. Res.* 41(4), 669–693.

Feeney, G. (1957), Controlling the distribution of empty cars, in: *Proc. 10th Nat. Meeting, Oper. Res. Soc. Am.*

Fisher, M.L., and M.B. Rosenwein (1989), An interactive optimization system for bulk-cargo ship scheduling, *Nav. Res. Logist.* 36, 27–42.

Flam, S., and R.J.-B. Wets (1987), Existence results and finite horizon approximates for infinite horizon optimization problems, *Econometrica* 55(5), 1187–1209.

Florez, H. (1986), *Empty-Container Repositioning and Leasing: An Optimization Model*, Ph.D. dissertation, Polytechnic Institute of New York.

Fong, C. (1974), *The Multi-Region Dynamic Capacity Expansion Problem: Exact and Heuristic Approaches*, Ph.D. Dissertation, Graduate School of Management, University of Rochester.

Fong, C., and V. Srinivasan (1981a), The multiregion dynamic capacity expansion problem. part II, *Oper. Res.* 29, 800–816.

Fong, C., and V. Srinivasan (1981b), The multiregion dynamic capacity expansion problem. part I, *Oper. Res.* 29, 787–799.

Fong, C., and V. Srinivasan (1986), The multiregion dynamic capacity expansion problem: An improved heuristic, *Manage. Sci.* 32(9), 1140–1152.

Fong, C., and V. Srinivisan (1964), Multiperiod capacity expansion and shipment planning with linear costs, *Nav. Res. Logist. Q.* 23(1), 255–260.

Ford, L., and D. Fulkerson (1956), Maximal flow through a network, *Can. J. Math.* 8, 399–404.

Ford, L., and D. Fulkerson (1958a), A suggested computation for maximal multicommodity networks flows, *Manage. Sci.* 5, 97–101.

Ford Jr., L., and D. Fulkerson (1958b), Constructing maximal dynamic flows from static flows, *Oper. Res.* 6, 419–433.

Ford, L., and D. Fulkerson (1962), *Flows in Networks*, Princeton University Press, Princeton.

Fox, K., B. Gavish, and S. Graves (1980), An n-constraint formulation of the (time dependent) traveling salesman problem, *Oper. Res.* 28, 1018–1021.

Frank, H. (1967), Dynamic communication networks, *IEEE Trans. Commun. Tech.* COM-15(2), 156–163.

Frank, H. (1969), Shortest paths in probabilistic graphs, *Oper. Res.* 17, 583–599.

Frank, H., and M. El-Bardai (1969), Dynamic communication networks with capacity constraints, *IEEE Trans. Commun. Tech.* COM-17(4), 432–437.

Frantzeskakis., L, (1990), *Dynamic networks with random arc capacities, with application to the stochastic dynamic vehicle allocation problem*, Ph.D. dissertation, Dept. of Civil Engng., Princeton University.

Frantzeskakis, L., and W. Powell (1989a), *An Improved Polynomial Bound on the Expected Recourse Function for Stochastic Networks*, Working paper–SOR-89-xx, Dept. of Civil Engng. and Oper. Res., Princeton University.

Frantzeskakis, L., and W. Powell (1989b), *Bounding procedures for dynamical networks with random link capacities, with application to stochastic programming*, Report sor-89-12, Princeton University.

Frantzeskakis, L., and W. Powell (1990), A successive linear approximation procedure for stochastic dynamic vehicle allocation problems, *Transp. Sci.* 24(1), 40–57.

Fredman, M., and R. Tarjan (1987), Fibonacci heaps and their uses in improved network optimization algorithms, *J. ACM* 34, 596–615.

Friesz, T., D. Bernstein, T. Smith, R. Tobin, and B. Wie (1993), A variational inequality formulation of the dynmamic network user equilibrium problem, *Oper. Res.* 41, 179–191.

Friesz, T., J. Luque, R. Tobin, and B. Wie, (1995), Dynamic network traffic assignment considered as a continuous time optimal control problem *Oper. Res.*, to appear.

Frieze, A., and G. Grimmet (1985), The shortest path problem for graphs with random arc lengths, *Discr. Appl. Math.* 10, 57–77.

Gale, D. (1959), Transient flows in networks, *Michigan Math. J.* 6, 59–63

Gallagher, R. (1978), A minimum delay routing algorithm using distributed computation, *IEEE Trans. Commun.* COM-25, 73–85.

Garey, M., and D. Johnson (1979), *Computers and Intractibility: A Guide to the Theory of NP-completeness*, Freeman, San Francisco.

Gavish, B. (1981), A decision support system for managing the transportation needs of a large corporation, *AIIE Trans.* 61–85.

Glasserman, P. (1991), *Gradient Estimation via Perturbation Analysis*, Kluwer, Norwell, Massachusetts.

Glickman, T., and H. Sherali, Large-scale network distribution of pooled empty freight cars over time, with limited substitution and equitable benefits, *Trans. Res.*, 19, 85–94.

Glover, F., G. Jones, D. Karney, D. Klingman, and J. Mote (1979), An integrated production, distribution and inventory planning system, *Interfaces* 9(5), 72–86.

Golden, B., and A. Assad, eds. (1988), *Vehicle Routing: Methods and Studies*, Studies in Management Science and Systems, Vol. 16, North-Holland, Amsterdam.

Golden, B., and J. Yee (1979), A framework for probabilistic vehicle routing, *AIIE Trans.* 11, 109–112.

Goldman, A. (1974), Optimal centre location in simple networks, *Transp. Sci.* 5, 212–221.

Gorenstein, S., S. Poley, and W. White (1971), *On the Scheduling of the Railroad Freight Operations*, Tech. Report 320-2999, IBM Philidelphia Scientific Center, IBM.

Graves, S.C., and J.B. Orlin (1985), A minimum concave-cost dynamic network flow problem with an application to lot-sizing, *Networks* 15, 59–71.

Grinold, R. (1977), Finite horizon approximations of infinite horizon linear programs, *Math. Program.* 12, 1–17.

Grinold, R. (1983a), Convex infinite horizon programs, *Math. Program.* 25, 64–82.

Grinold, R. (1983b), Model building techniques for the correction of end effects in multistage convex programs, *Oper. Res.* 31(3), 407–431.

Grinold, R., and D. Hopkins (1973), Computing optimal solutions for infinite-horizon mathematical programs with a transient stage, *Oper. Res.* 21, 179–187.

Gupal, A., and L. Bajenov (1972), Stochastic linearization, *Kibernetika*, 1.

Haghani, A. (1989), Formulation and solution of a combined train routing and makeup, and empty car distribution model, *Transp. Res.* 23B(6), 433–452.

Haimovich, M., A.R. Kan, and L. Stougie (1988), Analysis of heuristics for vehicle routing problems, in: B. Golden and A. Assad, eds., *Vehicle Routing: Methods and Studies*, North-Holland, Amsterdam.

Hakimi, S. (1964), Optimum location of switching centres and the absolute centres and medians of a graph, *Oper. Res.* 12, 450–459.

Halpern, J. (1977), Shortest path with time dependent length of edges and limited delay possibilities in nodes, *Z. Oper. Res.* 21, 117–124.

Hamacher, H., and S. Tufekci (1987), On the use of lexicographic-minimum cost flows in evacuation modeling, *Nav. Res. Logist.* 34, 487–503.

Hammer, P. (1969), Time-minimizing transportation problems, *Nav. Res. Logist. Q.* 16, 345–357.

Hammer, P. (1971), Communication on 'the bottleneck problem' and 'some remarks on the time transportation problem', *Nav. Res. Logist. Q.* 18, 487–490.

Hassin, R., and E. Zemel (1985), On shortest paths in graphs with random weights, *Math. Oper. Res.* 10, 557–564.

Hausman, W., and P. Gilmour (1967), A multiperiod truck delivery problem, *Transp. Res.* 1(4), 349–357.

Herren, H. (1973), The distribution of empty wagons by means of computer: An analytical model for the Swiss Federal Railways (SSB), *Rail Int.* 4(1), 1005–1010.

Herren, H. (1977), Computer controlled empty wagon distribution on the SSB, *Rail Int.* 8(1), 25–32.

Heyman, D., and M. Sobel (1984), *Stochastic Models in Operations Research, Volume II: Stochastic Optimization*, McGraw Hill, New York.

Higle, J.L., J. Bean, and R.L. Smith (1984), *Capacity Expansion Under Stochastic Demands*, Tech. Report 84-28, Dept. of Systems and Industrial Engng., University of Arizona.

Ho, J. (1980), A successive linear optimization approach to the dynamic traffic assignment problem, *Transp. Sci.* 14(4), 295–305.

Ho, J., and E. Loute (1983), Computational experience with advanced implementation of decomposition algorithms, *Math. Prog.* 27(3), 283–290.

Ho, J.K., and A.S. Manne (1974), Nested decomposition for dynamic models, in: E.A. Ho, ed., *Mathematical Programming 6*, North-Holland, Amsterdam.

Ho, Y.-C., M. Eyler, and T. Chien (1979), A gradient technique for general buffer storage design in a serial production line, *Int. J. Prod. Res.* 7(6), 557–580.

Hopkins, D. (1971), Infinite horizon optimality in an equipment replacement capacity expansion model, *Manage. Sci.* 18, 145–156.

Howard, R. (1971), *Dynamic Probabilistic Systems, Volume II: Semimarkov and Decision Processes*, Wiley, New York.

Hughes, R., and W. Powell (1988), Mitigating end effects in the dynamic vehicle allocation model, *Manage. Sci.* 34(7), 859–879.

Jaillet, P. (1985), *The Probabilistic Traveling Salesman Problems*, Tech. Report 185, Operations Research Center, MIT.

Jaillet, P. (1987), Stochastic routing problems, in: G. Andreatta, F. Mason, and P. Serafini, eds., *Stochastics in Combinatorial Optimization*, World Scientific, Singapore.

Jaillet, P. (1988), A priori solution of a traveling salesman problem in which a random subset of the customers are visited, *Oper. Res.* 36, 929–936.

Jaillet, P. (1991a), On some probabilistic combinatorial optimization problems defined on graphs, in: A. Odoni, L. Bianco, and G.S. "o, eds., *Flow Control of Congested Network*, Springer Verlag, Berlin.

Jaillet, P. (1991b), Probabilistic routing problems in the plane, in: H. Bradley, ed., *Operational Research 90*, Pergamon Press, London.

Jaillet, P. (1992a), Rates of convergence for quasi-additive smooth Euclidean functionals and application to combinatorial optimization problems, *Math. Oper. Res.* 17, 965–980.

Jaillet, P. (1992b), Shortest path problems with nodes failures, *Networks* 22, 589–605.

Jaillet, P. (1992c), Analysis of combinatorial optimization problems in Euclidean spaces, *Math. of Oper. Res.* 17, pp. 965–980.

Jaillet, P. (1993), Cube versus torus models for combinatorial optimization problems and the Euclidean minimum spanning tree constant, *Ann. Appl. Probab.* 3, pp. 582–592.

Jaillet, P., and A. Odoni (1988), Probabilistic vehicle routing problems, in: B. Golden and A. Assad, eds., *Vehicle Routing: Methods and Studies*, North-Holland, Amsterdam.

Jarvis, J., and H. Ratliff (1982), Some equivalent objectives for dynamic network flow problems, *Manage. Sci.* 28, 106–109.

Bookbinder, J.H., and S. Sethi (1980), The dynamic transportation problem: A survey, *Nav. Res. Logist. Q.* 27, 447–452.

Jones., K, (1992), *Multicommodity Network Flows: The Impact of Formulation on Decomposition*, Ph.D. dissertation, Dept of Civil Engng. and Oper. Res., Princeton University.

Jones, K., I. Lustig, J. Farvolden, and W. Powell (1993), Multicommodity network flows: The impact of formulation on decomposition, *Math. Program.* 62, pp. 65–117.

Jordan, W., and M. Turnquist (1983), A stochastic dynamic network model for railroad car distribution, *Transp. Sci.* 17, 123–145.

Kall, P., and D. Stoyan (1982), Solving stochastic programming problems with recourse including error bounds, *Math. Operationsforsch. Stat., Ser. Optimization*, 13(3), 431–447.

Karbowicz, C., and J. MacGregorSmith, (1984), A k-shortest path routing heuristic for stochastic evacuation networks, *Engng. Optimization*, 7, pp. 253–280.

Karmarkar, U. (1979), Convex/stochastic programming and multilocal inventory problems, *Nav. Res. Logist. Q.* 26(1), 1–19.

Karmarkar, U. (1981a), The multiperiod, multilocation invertory problem, *Oper. Res.* 29(2), 215–228.

Karmarkar, U. (1981b), Policy structure in multistage production/inventory problems: An application of convex analysis, *TIMS Stud. Manage. Sci.* 16, 331–352.

Karmarkar, U. (1987), The multilocation, multiperiod inventory problem: Bounds and approximations. *Manage. Sci.* 33(1), 86–94.

Karp, R. (1977), Probabilistic analysis of partitioning algorithms for the traveling salesman in the plane, *Math. Oper. Res.* 2, 209–224.

Karp, R. (1979), A patching algorithm for the non-symmetric traveling salesman problem, *SIAM J. Comput.* 8, 561–573.

Kaufman, D., and R.L. Smith (1993), Fastest paths in time-dependent networks for intelligent vehicle/highway systems application, *IVHS J.* 1, 1–11.

Kennington, J., and R. Helgason (1980), *Algorithms for Network Programming*, Wiley, New York.

Kiefer, J., and J. Wolfowitz (1952), Stochastic estimation of the maximum of a regression function, *Ann. Math. Stat.*, 23, 462–466.

Klingman, D., and J. Mote (1982), A multi-period production, distribution and inventory planning model, *Adv. Manage. Stud.* 1(2), 56–76.

Kornhauser, A., and V. Maslanka (1989), *A Scenario Aggregation Approach to the Dynamic Vehicle Allocation Problem Under Uncertainty*, Dept. of Civil Engng. and Oper. Res., Princeton University.

Kornhauser, A.L., and C. Chen (1990), *A Stochastic Multicommodity Network Optimization Model for the Dynamic Vehicle Allocation Problem*, Tech. Report, Dept. of Civil Engng. and Oper. Res., Princeton University.

Kraay, D., P. Harker, and B. Chen (1991), Optimal pacing of trains in freight railroads: Model formulation and solution, *Oper. Res.* 39(1), 82–99.

Laporte, G. (1988), Location-routing problems, in: B. Golden and A. Assad, eds., *Vehicle Routing: Methods and Studies*, North-Holland, Amsterdam.

Laporte, G., and P.J. Dejax (1989), Dynamic location-routing problems, *J. Opl. Res. Soc* 40(5), 471–482.

Laporte, G., and F. Louveaux (1990), Formulations and bounds for the stochastic capacitated vehicle routing problem with uncertain supplies, in: J. Gabzewicz, J. Richard, and L. Wolsey, eds., *Economic Decision-Making: Games, Econometrics and Optimization*, North-Holland, Amsterdam.

Laporte, G., F. Louveaux, and H. Mercure (1989), Models and exact solutions for a class of stochastic location-routing problems, *Eur. J. Oper. Res.* 39, 71–78.

Laporte, G., F. Louveaux, and H. Mercure (1992), The vehicle routing problem with stochastic travel times, *Transp. Sci.* 26(3), 161–170.

Larson, R., and A. Odoni (1981), *Urban Operations Research*, Prentice Hall, New Jersey.

Lasdon, L.S., and R. Terjung (1971), An efficient algorithm for multi-item scheduling, *Oper. Res.* 19(4), 946–969.

Lawler, E., A.R.K.J. Lenstra, and D. Shmoys, eds. (1985), *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, Wiley, Chichester.

Leddon, C., and E. Wrathall (1967), Scheduling empty freight car fleets on the louisville and nashville railroad, in: *Second International Symposium on the Use of Cybernetics on the Railways, October*, pp. 1–6, Montreal, Canada.

Lin, S. (1965), Computer solutions of the traveling salesman problem, *Bell System Tech. J.* 2245–2269.

Lundin, R.A., and T.E. Morton (1975), Planning horizons for the dynamic lot size model: Zabel vs. protective procedures and computational results, *Oper. Res.* 23(4), 711–734.

Lustig, I., J. Mulvey, and T. Carpenter, (1991), Formulating stochastic programs for interior point methods, *Oper. Res.*, 39(5), 757–770.

MacGregor Smith, J., and D. Towsley, (1981), The use of queueing networks in the evaluation of egress from buildings, *Environ. Planning*, Vol. B8, pp. 125–139.

MacGregor Smith, J. (1993), Multi-objective routing in stochastic evacuation networks, in: D.-Z. Du and P. Pardalos, eds., *Network Optimization Problems*, pp. 263–281, World Scientific.

Magnanti, T., and R. Simpson (1978), *Transportation Network Analysis and Decomposition Methods*, Report No. DOT-TSC-RSPD-78-6, U.S. Dept. of Transportation.

Malandrak, C., and M.S. Daskin (1992a), Time dependent vehicle routing problems: Formulations, properties and heuristic algorithms, *Transp. Sci.* 26(3), 185–200.

Malandrak, C., and R.B. Dial (1992b), *A Dynamic Programming Heuristic Algorithm for the Time Dependent Traveling Salesman Problem*, Unpublished tech. report, Roadnet Technologies, Inc.

Malandraki, C. (1989), *Time Dependent Vehicle Routing Problems: Formulations, Solution Algorithms and Computational Experiments*, Ph.D. Dissertation, Department of Civil Engineering, Northwestern University.

Marsten, R., and F. Shepardson (1981), Exact solution of crew scheduling problems using the set partitioning model: Recent successful applications, *Networks* 11, 167–177.

Mendiratta., V, (1981), *A Dynamic Optimization Model of the Empty Car Distribution Process*, Ph.D. Dissertation, Dept. of Civil Engng., Northwestern University.

Mendiratta, V., and M. Turnquist (1982), A model for the management of empty freight cars, *Trans. Res. Rec.* 838, 50–55.

Merchant, D., and G. Nemhauser (1978a), A model and an algorithm for the dynamic traffic assignment problem, *Transp. Sci.* 12(3), 200–207.

Merchant, D., and G. Nemhauser (1978b), Optimality conditions for a dynamic traffic assignment model, *Transp. Sci.* 12(3), 183–199.

Minieka, E. (1973), Maximal, lexicographic and dynamic network flows, *Oper. Res.* 12(2), 517–527.

Minieka, E. (1974), Dynamic network flows with arc changes, *Networks* 4, 255–265.

Minoux, M. (1987), Network synthesis and dynamic network optimization, *Ann. Discr. Math.* 31, 283–324.

Mirchandani, P., and R. Francis, eds. (1990), *Discrete Location Theory*, Wiley, New York.

Misra, S. (1972), Linear programming of empty wagon disposition, *Rail Int.* 3, 151–158.

Morton, T. (1979), Infinite horizon dynamic programming models: A planning horizon formulation, *Oper. Res.* 27(4), 730–742.

Morton, T. (1981), Forward algorithms for forward thinking managers, in: R. Schultz, ed., *Applications of Management Science*.

Morton, T., and W. Wecker (1977), Discounting, ergodicity, and convergence for Markov decision processes, *Manage. Sci.* 23, 890–900.

Mulvey, J., and H. Vladimirou (1989), Solving Multistage Stochastic Networks: An Application of Scenario Aggregation, SOR-88-1, *Statistics and Operations Research Series*.

Mulvey, J., and S. Zenios (1987), *GENOS 1.0 User's Guide: A Generalized Network Optimization System*, Tech. Rep. 87-12-03, Princeton University.

Mulvey, J., and S. Zenios (1987), Real-time operational planning for the U.S. air traffic system, *Appl. Num. Math.* 3, 427–441.

Murphy, F., and A. Soyster (1986), End effects in capacity expansion models with finited horizons, *Nav. Res. Logist. Q.* 33, 373–383.

Odoni, A. (1986), The flow management problem in air traffic control, in: L.B.A.R. Odoni and G. Szego, eds., *Flow Control of Congested Networks*, pp. 269–288, Springer-Verlag, New York.

Odoni, A. (1987), Stochastic facility location problems, in: G. Andreatta, F. Mason, and P. Serafini, eds., *Stochastics in Combinatorial Optimization*, World Scientific, Singapore.

Orda, A., and R. Rom (1990), Shortest-path and minimum-delay algorithms in networks with time-dependent edge-length, *J. Assoc. Comput. Machin.* 37, 607–625.

Orlin, J. (1983), Maximum-throughput dynamic network flows, *Math. Program.* 27(2), 214–231.

Orlin, J. (1984a), Minimum convex cost dynamic network flows, *Math. Oper. Res.* 9(2), 190–207.

Orlin, J. (1984b), Some problems on dynamic/periodic graphs, *Prog. Combinat. Optim.* pp. 273–293.

Papadimitriou, C., and K. Steiglitz (1982), *Combinatorial Optimization: Algorithms and Complexity*, Prentice Hall, New Jersey.

Perakis, A., and N. Papadakis (1989), Minimal time vessel routing in a time-dependent environment, *Transp. Sci.* 23(4), 266–276.

Picard, J., and M. Queyranne (1978), The time dependent travelling salesman problem and its application to the tardiness problem in one-machine scheduling, *Oper. Res.* 26(2), 86–110.

Platzman, L., and J. Bartholdi (1989), Spacefilling curves and the planar traveling salesman problem, *J. ACM* 36, 719–737.

Powell, W. (1986), A stochastic model of the dynamic vehicle allocation problem, *Transp. Sci.* 20, 117–129.

Powell, W. (1987), An operational planning model for the dynamic vehicle allocation problem with uncertain demands, *Transp. Res.* 21B, 217–232.

Powell, W. (1988), A comparative review of alternative algorithms for the dynamic vehicle allocation problem with uncertain demands, *Vehicle Routing: Methods and Studies*, pp. 249–292.

Powell, W. (1991), Optimization models and algorithms: An emerging technology for the motor carrier industry, *IEEE Trans. Vehicular Technol.* 40, 68–80.

Powell, W., E. Berkkam, and I. Lustig (1993), On algorithms for nonlinear dynamic networks, in: D. Du and P. Pardalos, eds., *Network Optimization Problems: Algorithmis, Complexity and Applications*, pp. 203–231, World Scientific, New Jersey.

Powell, W., and R. Cheung (1994a), Stochastic programs over trees with random arc capacities, *Networks* 24, 161–175.

Powell, W., and R. Cheung (1994b), A network recourse decomposition method for dynamic networks with random arc capacities, *Networks* 24, 369–384.

Powell, W., and L. Frantzeskakis (1992), Restricted recourse strategies for stochastic, dynamic networks, *Transp. Sci.* (to appear).

Powell, W., and Y. Sheffi (1982), The convergence of equilibrium algorithms with predetermined step sizes, *Transp. Sci.* 16, 45–55.

Powell, W., and Y. Sheffi (1989), Design and implementation of an interactive optimization system for network design in the motor carrier industry, *Oper. Res.* 37(1), 12–29.

Powell, W., Y. Sheffi, K. Nickerson, K. Butterbaugh, and S. Atherton (1988), Maximizing profits for North American van lines' truckload division: A new framework for pricing and operations, *Interfaces* 18, 21–41.

Powell, W., Y. Sheffi, and S. Thiriez (1984), The dynamic vehicle allocation problem with uncertain demands, in: *Ninth Int. Symp. on Transportation and Traffic Theory*, pp. 357–374.

Psaraftis, H. (1988), Dynamic vehicle routing problems, in: B. Golden and A. Assad, eds., *Vehicle Routing: Methods and Studies*, pp. 223–248, North-Holland, Amsterdam.

Psaraftis, H., J. Orlin, D. Bienstock, and P. Thompson (1985), *Analysis and Solution Algorithms of Sealift Routing and Scheduling Problems*, Sloan Working Paper 1700-85, Sloan School of Management, MIT.

Psaraftis, H., and J. Tsitsiklis, Dynamic shortest paths with Markovian arc costs (1990), preprint.

Qi, L. (1985), Forest iteration method for stochastic transportation problem, *Math. Program. Study* 25, 142–163.

Ran, B., D. Boyce, and L. LeBlanc (1992), Toward a new class of instantaneous dynamic user-optimal traffic assignment models, *Oper. Res.* (to appear).

Ratcliffe, L., B. Vinod, and F. Sparrow (1984), Optimal prepositioning of empty freight cars, *Simulation*, 269–275.

Rhee, W., and M. Talagrand (1988), A sharp deviation inequality for the stochastic traveling salesman problem, *Ann. Probab.* 17, 1–8.

Robbins, H., and S. Monro (1951), A stochastic approximation method, *Ann. Math. Stat.* 22, 400–407.

Rockafellar, T., and R. Wets (1991), Scenarios and policy aggregation in optimization under uncertainty, *Math. Oper. Res.* 16(1), 119–147.

Russell, R., and D. Gribbin (1991), A multiphase approach to the period routing problem, *Networks* 21, 747–765.

Seneta, E. (1981), *Nonnegative Matrices and Markov Chains*, Springer-Verlag, New York.

Sethi, S., and S. Bhaskaran (1985), Conditions for the existence of decision horizons for discounted problems in a stochastic environment, *Oper. Res. Lett.* 4(2), 61–64.

Shan, Y. (1985), *A Dynamic Multicommodity Network Flow Model for Real-Time Optimal Rail Freight Car Management*, Ph.D. Dissertation, Princeton University.

Shapiro, J. (1993), Mathematical programming models and methods for production planning and scheduling, in: A.R.K.S.C. Graves and P. Zipkin, eds., *Logistics of Production and Inventory*, pp. 371–444.

Sheffi, Y., and W. Powell (1982), An algorithm for the equilibrium assignment problem with random link times, *Networks* 12, 191–207.

Shor, N. (1979), *The Methods of Nondifferentiable Op[timization and their Applications*, Naukova Dumka, Kiev.

Simchi-Levi, D., and O. Berman (1988), Heuristics and bounds for the traveling salesman location problem on the plane, *Oper. Res. Lett.* 5, 243–248.

Smith, S., and Y. Sheffi (1992), *Railroad Car Distribution: A Fast Network Model with Multiple Car Types and Delays*, CTS working paper, M.I.T.

Spaccamela, A.M., A.R. Kan, and L. Stougie (1984), Hierarchical vehicle routing problems, *Networks* 14, 571–586.

Stanley, J. (1987), A forward convex-simplex method, *Eur. J. Oper. Res.* 29, 328–335.

Steele, J. (1981), Complete convergence of short paths and karp's algorithm for the tsp, *Mathematics of Operations Research*, 6, 374–378.

Steele, J. (1981), Subadditive Euclidean functionals and nonlinear growth in geometric probability, *Ann. Probab.* 9, 365–376.

Stewart, W., and B. Golden (1983), Stochastic vehicle routing: A comprehensive approach, *Eur. J. Oper. Res.* 14(3), 371–385.

Szwarc, W. (1964), The transportation problem with stochastic demand, *Manage. Sci.* 11, 35–50.

Szwarc, W. (1966), The time transportation problem, *Zastowania Matematyki, Warsaw* 8, 231–242.

Szwarc, W. (1971a), The dynamic transportation problem, *Nav. Res. Logist. Q.* 13, 335–345.

Szwarc, W. (1971b), Some remarks on the time transportation problem, *Nav. Res. Logist. Q.* 18, 473–485.

Tapiero, C. (1971), Transportation-location-allocation problems over time, *J. Regional Sci.* 11, 377–384.

Tapiero, C. (1975), Transportation over time: A system theory approach, in: *Proceedings – XX International TIMS Meeting, Tel-Aviv, Israel, Vol 1*, pp. 239–243, Jerusalem.

Tapiero, C., and M. Soliman (1972), Multicommodities transportation schedules over time, *Networks* 2, 311–327.

Tillman, F. (1969), The multiple terminal delivery problem with probabilistic demands, *Transp. Sci.* 3, 192–204.

Trudeau, P., and M. Dror (1992), Stochastic inventory routing: Route design with stockouts and route failures, *Transp. Sci.* 26(3), 171–184.

Turnquist., M, Mov-em: A network optimization model for empty freight car distribution (1986), School of Civil and Environmental Engineering, Cornell University.

Turnquist, M.A., and B.P. Markowicz (1989), *An Interactive Microcomputer-Based Planning Model for Railroad Car Distribution*, For the CORS/ORSA/TIMS National Meeting, Cornell University.

Van Roy, T., and D. Erlenkotter (1980), *A Dual-Based Procedure for Dynamic-Facility Location*, Working paper 80-31, International Institute for Applied Systems Analysis, Laxenburg, Austria.

Van Slyke, R., and R. Wets (1969), L-shaped linear programs with applications to optimal control and stochastic programming, *SIAM J. Appl. Math.* 17, 638–663.

Veinott Jr, A. (1969), Minimum concave-cost solution of leontief substitution models of multi-facility inventory systems, *Oper. Res.* 17, 262–291.

Vemuganti, R., M. Oblak, and A. Aggarwal (1989), Network models for fleet management, *Decision Sci.* 20, 182–197.

Vladimirou, H. (1991), *Stochastic Networks: Solution Methods and Applications in Financial Planning*, Ph.D. dissertation, Dept. of Civil Engng. and Oper. Res., Princeton University.

Wagner, H., and Whithin (1958), Dynamic version of the economic lot size model, *Manage. Sci.* 5, 89–96.

Wallace, S. (1986), Decomposing the requirement space of a transportation problem, *Math. Prog. Study*, 28, 29–47.

Wallace, S. (1987), A piecewise linear upper bound on the network recourse function, *Math. Program.* 38, 133–146.

Wasan, M. (1969), Stochastic approximations, in: *Cambridge Transactions in Math. and Math. Phys. 58*, Cambridge University Press, Cambridge.

Wets, R. (1966), Programming under uncertainty: The equivalent convex program, *SIAM J. Appl. Math.* 14, 89–105.

Wets, R. (1974), Stochastic programs with fixed resources: The equivalent deterministic problem, *SIAM Rev.* 16, 309–339.

Wets, R. (1982), Stochastic programming: Solution techniques and approximation schemes, in: M.G.A. Bachen and B. Korte, eds., *Mathematical Programming: The State of the Art*, pp. 566–603, Springer-Verlag, Berlin.

Wets, R. (1988), Large scale programming techniques in stochastic programming, in: Y. Ermoliev and R. Wets, eds., *Numerical Methods in Stochastic Programming*, Springer-Verlag, Berlin.

White, W. (1972), Dynamic transshipment networks: An algorithm and its application to the distribution of empty containers, *Networks* 2(3), 211–236.

White, W., and A. Bomberault (1969), A network algorithm for empty freight car allocation, *IBM Systems J.* 8(2), 147–171.

Wilkinson, W. (1971), An algorithm for universal maximal dynamic flows in a network, *Oper. Res.* 19(7), 1602–1612.

Williams, A. (1963), A stochastic transportation problem, *Oper. Res.* 11, 759–770.

Zadeh, N. (1974), On building minimum cost communication networks over time, *Networks* 4, 19–34.

Zahorik, A., L. Thomas, and W. Trigeiro (1984), Network programming models for production scheduling in multi-stage, multi-item capacitated systems, *Manage. Sci.* 30(3), 308–325.

Zangwill, W. (1968), Minimum concave cost flows in certain networks, *Manage. Sci.* 14(7), 429–450.

Zangwill, W. (1969), A backlogging model and a multi-echelon model of a dynamic economic lot size product system – a network approach, *Manage. Sci.* 15(9), 506–527.

Zipkin, P. (1982), Transportation problems with aggregated destinations when demands are uncertain, *Nav. Res. Logist. Q.* 29, 257–270.