An Alternating Direction Method of Multiplier Based Problem Decomposition Scheme for Iteratively Improving Primal and Dual Solution Quality in Vehicle Routing Problem

Yu Yao
School of Traffic and Transportation, Beijing Jiaotong University
Beijing, 100044, China
Email: yaoyu1@bjtu.edu.cn

Xiaoning Zhu
School of Traffic and Transportation, Beijing Jiaotong University
Beijing, 100044, China
Email: xnzhu@bjtu.edu.cn

Hongyu Dong*
Logistics R&D Department, Beijing Jingdong Zhenshi Information Technology Co., Ltd.
Beijing, 100176, China
Email: donghongyu@jd.com

Shengnan Wu
Logistics R&D Department, Beijing Jingdong Zhenshi Information Technology Co., Ltd.
Beijing, 100176, China
Email: wushengnan1@jd.com

Hailong Wu
Logistics R&D Department, Beijing Jingdong Zhenshi Information Technology Co., Ltd.
Beijing, 100176, China
Email: wuhailong@jd.com

Lu (Carol) Tong
School of Electronic and Information Engineering, Beihang University,
National Engineering Laboratory for Comprehensive Transportation Big Data Application Technology,
Beijing 100091, China
Email: ltong@buaa.edu.cn

Xuesong Zhou*
School of Sustainable Engineering and the Built Environment, Arizona State University
Tempe, AZ, 85281, USA
Email: xzhou74@asu.edu

(*Corresponding Authors)

ABSTRACT

Emerging urban logistics applications need to address a wide range of challenges, including complex traffic conditions and time-sensitive requirements. In this paper, in the context of urban logistics we consider a vehicle routing problem with time-dependent travel times and time windows (VRPTW), and the goal is to minimize the total generalized costs including the transportation cost, vehicular waiting cost and the fixed cost associated with each vehicle. We adopt a high-dimensional space-time network flow model to formulate the underlying vehicle routing problem with a rich set of criteria and constraints. A thorny issue, when solving VRP, is how to iteratively improve both primal and dual solution quality in general and how to break symmetry due to many identical solutions, especially with homogeneous vehicles. Along this line, many coupling constraints, such as the consensus constraints across different agents or decision makers, need to be carefully addressed to find high-quality optimal or close-to-optimal solutions under medium-scale or large-scale instances. Currently, Alternating Direction Method of Multipliers (ADMM) is widely used in the field of convex optimization, as an integration of Augmented Lagrangian relaxation and block coordinate descent methods for a large number of machine learning and large-scale continuous system optimization and control. In this paper, we first introduce the ADMM to the multi-vehicle routing problem, which is a special case of integer linear programming, and demonstrate the way to reduce the quadratic penalty terms used in ADMM into simple linear functions. In a broader context, a computationally reliable dual decomposition framework is developed to iteratively improve both primal and dual solution quality. Essentially, the shortest path sub-problem or other similar sub-problem involving binary decisions can be embedded into a sequential solution scheme with an output of both lower bound estimates and upper bound feasible solutions. We examine the performance of the proposed approach using classical Solomon VRP benchmark instances. A real-world instance is specifically tested, based on a problem solving competition offered by Jingdong Logistics, a major E-commerce company.

Keywords: Urban logistics, Vehicle routing problem with time windows, Alternating direction method of multipliers, problem decomposition

1. Introduction

In this paper, we are interested in how to develop a computationally reliable and theoretically sound problem decomposition scheme for solving a wide range of emerging scheduling and routing problems in dynamic transportation networks. In particular, our research is also motivated by solving vehicle routing and closely related ride sharing problems due to rapid development of e-commerce industries and traveler mobility services. The solution algorithm of vehicle routing problems is one of most important building blocks for providing "door to door" freight distributions (Savelsbergh and Van Woensel, 2016) and customized public transportation services (Tong et al., 2017), with high frequency and punctuality in congested urban areas.

1.1 Literature review on VRP

Compared with long-distance transportation between cities, the efficiency of urban logistics highly depends on real-time traffic conditions. At the same time, the customers have stricter requirements in delivery time and locations. These complex traffic conditions and increasingly complicated requirements motivate the development of optimized strategic and operational decision making in order to manage the logistic processes more efficiently. In the context of urban logistics, the VRP algorithmic development should be developed with the consideration of time varying traffic conditions, time window of customer requirements and vehicle carrying capacity.

In this study, we consider a vehicle routing problem with time windows (VRPTW), which includes various constraints such as service time windows and vehicle carrying capacity constraints. The VRPTW is widely studied in the literature (Desrochers et al., 1992; Kallehauge, 2008; Zhou et al., 2018). Typical solution methods of VRPTW can be classified into two categories: heuristics and optimization approaches. Although there are abundant efficient heuristic methods such as savings algorithm (Clarke and Wright, 1964), matching based algorithm (Altinkemer and Gavish, 1991), sweep based algorithm (Gillett and Miller, 1974; Renaud and Boctor, 2002), cluster first, route second (Fisher and Jaikumar, 1981) and meta-heuristics such as Tabu search (Taillard, 1993), Adaptive Large Neighborhood Search (ALNS) algorithm (Ghilas et al., 2016; Goel and Gruhn, 2005), and ant systems optimization (Reimann et al., 2004), these widely used heuristic or meta-heuristic methods typically do not offer measures on optimality gaps. On the other hand, exact or approximate optimization approaches, such as Branch and Cut, Branch and Price and Lagrangian decomposition methods, deserve a particular attention as they can provide a yardstick to evaluate the obtained solutions and further reach the right balance between the solution search efforts and required optimality.

1.2 Literature review on problem decomposition

Decomposition is a general approach to solve large-scale problems. Its core is to break the original problem into smaller sub-problems and to solve each of them separately, either in parallel or sequential. There are a wide range of classical decomposition methods, such as Dantzig-Wolfe (Dantzig and Wolfe, 1960), Benders (Benders, 1962), Column generation (Ford Jr and Fulkerson, 1958), Lagrangian relaxation (Held and Karp, 1970) and Branch and Price (Nemhauser et al., 1991). Interested readers can find a number of surveys and textbook chapters on various optimization domains, to name a few, contributed by Boyd and Vandenberghe (2004), Lasdon (2002), chapter 6 of Bertsekas (1999), Wolsey and Rinaldi (1993) and chapter 12 of Bradley et al. (1977) for linear, nonlinear and integer programming problems.

These decomposition schemes have been widely used to solve vehicle routing problems. In following classical literature, Branch and Bound (Christofides et al., 1981), Branch and Cut (Laporte et al., 1985), Lagrangian relaxation (Fisher et al., 1997) and Branch and Cut and Price (Fukasawa et al., 2006) are adopted specifically to solve the vehicle routing problem. Recently, notable research attention has been devoted to the VRP in the context of time-dependent transportation networks. Dabia et al. (2013) adopted the branch-and-price framework to decompose the arc-based formulation into a set-partitioning problem as the master problem, and the pricing problem is constructed as a time-dependent shortest path problem with resource constraints. Aiming to embed vehicle capacity and pick up and deliver precedence constraints in a layered graph structure, Mahmoudi and Zhou (2016) constructed a multi-dimensional commodity flow formulation where possible transportation states are enumerated; then a Lagrangian relaxation approach was used to decompose the original model into a sequence of shortest path sub-problems. As demonstrated in the Branch and Price literature by Barnhart et al. (1998), and in a recent publication by Niu et al. (2018) focusing on integrated transit vehicle assignment and scheduling, researchers need to fully recognize the critical solution symmetry issues, and develop effective symmetry breaking techniques, when handling the multi-vehicle routing problem that consists of numerous homogeneous vehicles.

In this study, with a dual decomposition paradigm, we develop a computationally reliable solution framework based on Alternating Direction Method of Multipliers (ADMM), which is a variation of dual decomposition that provides improved theoretical and practical convergence properties. The literature of ADMM can be date back to the classical paper by

Glowinski and Marroco (1975); and its convergence analysis and many related theoretical building blocks have been subsequently established by a number of studies in the field of convex programming (Eckstein and Bertsekas, 1992; Ruszczyński, 1989). A relatively recent overview offered by Boyd et al. (2011) further populated the use of ADMM in many disciplines, particularly in a few emerging branches of big data and machine learning. In the area of distributed optimization with multiple agents, one of the leading studies by Nedic and Ozdaglar (2009), proposed an ADMM-motivated consensus and sharing mechanism where each agent has its own convex, potentially non-differentiable objective function. In the seminar survey by Boyd et al. (2011), the use of ADMM for nonconvex statistical learning problems have been also examined in problems, e.g., for regressor selection and factor model fitting, in which the individual steps can be carried out exactly. In the perspective of stochastic multi-stage mixed-integer programs, Boland et al. (2018) have creatively adopted the ADMM-based solution procedure to handle the progressive hedging model, a stochastic form of augmented Lagrangian based on scenario decomposition. They pointed out that the drawback of primal-dual methods for mixed-integer programs was hard to guarantee the convergence. To address this issue, a FW-PH algorithm for computing lower bounds was presented and proved that it was convergence to Lagrangian bounds.

To our limited knowledge, very few studies have specifically focused on the adaptation of ADMM on the vehicle routing problems, which is essentially a deterministic combinatorial optimization problem in transportation networks. More importantly, our proposed ADMM-based solution framework can be further applied to reformulate a broader class of consensus constraints so as to binary decision variables are carefully constructed to enable a wide range of computationally efficient algorithms in transportation networks.

1.3 Motivation and potential contributions

While the vehicle routing problem has been widely studied, we hope the introduction of ADMM as an improved dual decomposition algorithm could shed more light in the following aspects. First, compared to meta-heuristic methods, a desirable algorithm should not only obtain good feasible solutions, but also establish a strong lower bound estimation to precisely measure the quality of the solutions. Second, in the classical Branch and Price solution framework for VRP, each vehicle maintains multiple alternative routes in the column pool, the master problems are usually solved by invoking linear programming (LP) solvers with needed sophisticated branching strategies. In contrast, our proposed ADMM framework only keeps one path column for each vehicle at an iteration, which offers a relatively simpler algorithmic implementation structure, especially for time-indexed formulation problems with a huge number of columns to manage. Essentially, this ADMM framework aims to iteratively improve both primal and dual feasibility; and this decomposition procedure could be further extend to handle other transportation problems beyond VRP, such as problems with consistency constraints between optimization layers.

In our research, ADMM is adopted as a high-level problem decomposition and modular coordination framework, which obtains the solution of the large-scale problem through solving a set of much smaller sub-problems with efficient algorithms as the base operations. We perform a sequence of reformulation steps, namely (a) dualization and augmentation, (b) decomposition, and (c) linearization. Specifically, this research can address a number of modeling challenges.

(1) If a LP relaxation (of the set covering problem) is used in the restricted master problem for VRP, e.g. in the form of D-W decomposition or Branch and Bound, it typically needs significant efforts to obtain and improve feasible integer solutions. On the other hand, the limitation of the standard Lagrangian relaxation method for VRP is its inherent solution symmetry due to homogeneous vehicles. By adding the augmented terms, we present an improved dual decomposition to effectively break the solution symmetries and quickly generate good feasible integer solutions.

(2) ADMM has been widely used in the field of convex programing, but to our limited knowledge, there are very few applications of ADMM in linear integer programming in general and VRP specially, which involves the optimal coordination of multiple vehicles with a large number of integer decision variables subject to a set of complex side constraints. To tackle this difficulty, we use a hyper-dimensional network model to simplify side constraints and further enable an efficient dynamic programing algorithm for the dualized problem.

(3) Another challenge in applying ADMM in VRP is how to linearize the quadratic objective function in its inner penalty term involving integer variables. The regular way is to linearize the objective function, e.g., using Frank-Wolf method or first-order Taylor expansion (Nishi et al., 2005), which is still computationally expensive in its own right. In this paper, we clearly show that, within the block coordinate descent method of Gauss-Seidel type, the quadratic penalty term used in the VRP-ADMM model is separable for each "x-update", and could be reduced to a much simpler linear functional form if only binary decision values are involved. The establishment of this equivalence gives rise to the possibility and promise of computationally efficient iterative solution searching procedures.

(4) If applying the ADMM method alone, only upper bound solutions can be obtained; and information provided here is insufficient to evaluate the corresponding solution quality with respect to the system-wide optimum. In this paper, we present a coherent solution approach for simultaneously estimating upper bound and lower bound values. This process

can accordingly access and reduce the global optimality gaps by iteratively solving two closely related problems namely Lagrangian dual and the augmented models.

The remainder of this paper is structured as follows. In Section 2, the formulation of VRPTW is represented based on the state-space-time network. In Section 3, the ADMM-based decomposition framework and solution procedure are presented. Section 4 examines the symmetry-breaking and computationally reliable properties of the ADMM using examples. In Section 5, we discuss the convergence and potential extensions of the ADMM. In Section 6, the proposed model and solution framework are applied to Solomon benchmark and a real-world case. Finally, we give our conclusions in Section 7.

2. Problem statement and model formulation

2.1 Problem statement of the VRPTW

The VRPTW in this paper aims to find a set of routes that minimizes total generalized system-wide costs, including the transportation cost, vehicular waiting cost and fixed cost. Given a physical transportation network denoted by (N, M), where N is a set of nodes and M is a set of directed links. We use $TT(i, j, t)$ to represent the travel time on the link $(i, j) \in M$ when departing at time t. For simplicity, the time-dependent travel times are assumed to be deterministic and can be pre-calculated according to predictable traffic conditions during a day.

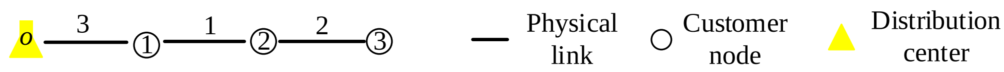
Nodes in this physical network are categorized into two different types, including customer nodes denoted by $p, p \in P$ and the distribution center denoted by o. That is, $N = \{o\} \cup P$. Each customer $p \in P$ is characterized by a volume and a weight of demand and a preferred service time window $[e_p, l_p]$, where e_p is the desired earliest time of service and l_p is the latest time. All customer nodes in P must be visited by a single vehicle exactly once.

The distribution center o is the origin and destination of all vehicles, while a vehicle enables to load and unload commodities at the distribution center in the middle of its tour. That is, after delivering one batch of packages, a vehicle can go back to the distribution center to load the assigned items for the next batch. Along the planning horizon, each vehicle is designated with a given service period $[e_v, l_v]$, where e_v is the earliest time to depart form the distribution center and l_v is the latest time to return.

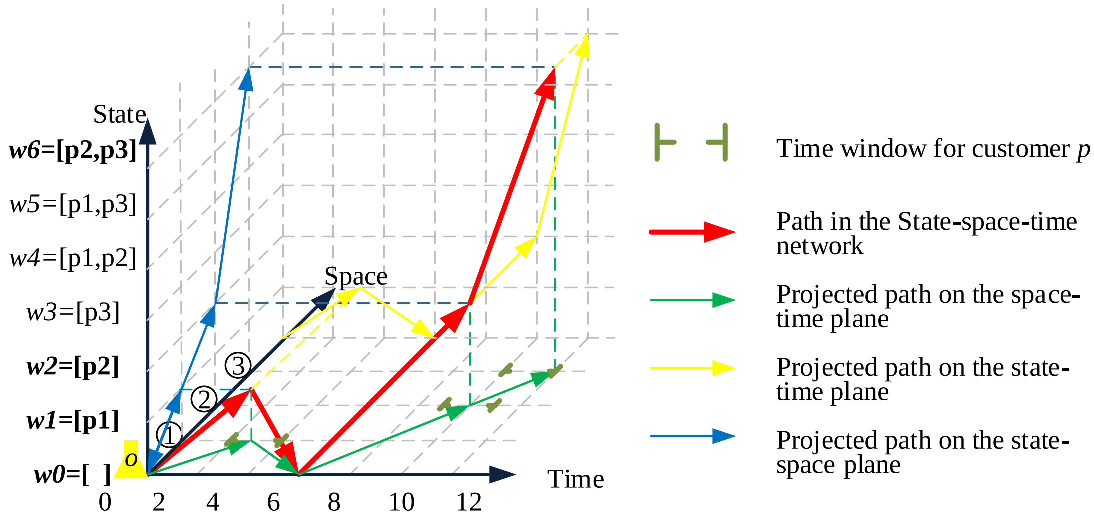
2.2 Time-indexed and state-indexed network representation for VRPTW

A standard way to formulate the VRP model is to use cumulative time and cumulative load variables, based on a customer node-oriented network directly (Cordeau, 2006), where variables are used in a set of the time window and vehicle carrying capacity constraints. Another widely used mathematical formulation is established through a time-expanded network construction (Boland et al., 2017), in which the time window requirements are naturally enforced.

Within a dual decomposition framework, a desirable model should have a very limited number of complicated constraints to be dualized; and a computationally efficient algorithm is required to solve the relaxed problem with high-dimensionality. Therefore, we adopt the hyper-dimensional multi-commodity flow modeling framework proposed by Mahmoudi and Zhou (2016), and construct a time-indexed and state-indexed network with three dimensions, including (a) space dimension, (b) time dimension and (c) cumulative service state dimension. $G = (E, A)$ is used to represent the hyper-network with a set of vertices E and a set of arcs A. Vertex$(i, t, w) \in E$ is extended from the node $i \in N$, and each arc $(i, j, t, s, w, w') \in A$ indicates a directed state-space-time path from vertex (i, t, w) to vertex (j, s, w'). Specifically, t represents the uniformly discretized time interval (e.g., 1 min) in the planning time horizon. Note that this dimension naturally embeds the time window constraint. Besides, w represents "cumulative service state" of the vehicle, which tracks served customers and provides the corresponding carrying volume and weight information of packages in order to satisfy the vehicle capacity constraints. The cumulative service state of a vehicle is updated into initial status after going back to the distribution center o. Fig. 1 illustrates a simple example of state-space-time path, which corresponds to a sequence of nodes $(O,0,w_0) \rightarrow (1,3,w_1) \rightarrow (O,6,w_0) \rightarrow (2,10,w_2) \rightarrow (3,12,w_6)$. It is clear that the time window and capacity constraints are embedded in the network through time and state dimensions.



(a) Physical Network



(b) A path in the State-space-time network

Fig. 1. A simple example of state-space-time path adopted from Mahmoudi and Zhou (2016)

2.3 Time-discretized, multi-dimensional multi-commodity flow formulation of VRPTW

The sets, indexes, variables and parameters used for model formulation and network construction are described in Tables 1 and 2.

Table 1
Sets, indexes and variables used for model formulation.

Symbol	Definition
V	Set of physical vehicles
P	Set of customers
W	Set of cumulative service states
A_v	Set of state-space-time arcs in vehicle v's network
$\Psi_{p,v}$	Set of unloading arcs for customer p in vehicle v's network
v	Index of vehicles
p	Index of customers
w	Index of cumulative service states
(i,t,w), (j,s,w')	Indexes of state-space-time vertexes
(i,j,t,s,w,w')	Index of a space-time-state arc indicating that one travels from node i at time t with cumulative service state w to the node j at time s with state w'
a	The abbreviation of arc (i,j,t,s,w,w')
$x^v_{i,j,t,s,w,w'}$	$= 1$ if arc (i,j,t,s,w,w') is used by vehicle v; $= 0$ otherwise

Table 2
Parameters used in network construction and model formulation

Symbol	Definition
$[e_p, l_p]$	Time window of customer p
$[e_v, l_v]$	Service period of vehicle v
o_v	Origin of vehicle v
d_v	Destination of vehicle v
w_0	Initial state
$c_{i,j,t,s,w,w'}$	Cost of arc (i,j,t,s,w,w')
$TT(i,j,t)$	Travel time on the link (i,j) when departing at time t

Here we frame the VRPTW model as follows.

Objective Function:

$$\min\ Z = \sum_{v\in V} \sum_{(i,j,t,s,w,w')\in A_v} c_{i,j,t,s,w,w'} x^v_{i,j,t,s,w,w'} \tag{1}$$

The objective function of the proposed model is to minimize the total costs of all selected arcs, which can be divided into the following three types: (a) transportation arcs, (b) waiting arcs and (c) loading and unloading arcs. The transportation arc (i,j,t,s,w,w) represents a vehicle moving from node i to node j based on the given time-dependent travel time $TT(i,j,t)$, $s = t + TT(i,j,t)$. The waiting arc $(i,i,t,t+1,w,w)$ represents a waiting activity at node i from time t to t+1, i.e. vehicle location and carrying state remains unchanged for one time interval. The loading and unloading arcs (i,i,t,s,w,w') represent a loading activity at distribution center or an unloading activity at customer nodes with the state changing from w to w'.

Flow balance constraints:

As the time window, vehicle carrying volume and weight capacity constraints for each vehicle are inexplicitly represented in the hyper-network construction so that we just need to ensure all of the selected arcs can constitute feasible paths from the origin to the destination by following flow balance constraints.

Flow balance constraints at vehicle v's origin vertex:

$$\sum_{(i,j,t,s,w,w')\in A_v} x^v_{i,j,t,s,w,w'} = 1 \quad i = o_v, t = e_v, w = w_0, \qquad \forall v \in V \tag{2}$$

Flow balance constraint at vehicle v's destination vertex:

$$\sum_{(i,j,t,s,w,w')\in A_v} x^v_{i,j,t,s,w,w'} = 1 \quad j = d_v, s = l_v, \qquad \forall v \in V \tag{3}$$

Flow balance constraint at intermediate vertex:

$$\sum_{(j,s,w'')} x^v_{i,j,t,s,w,w''} - \sum_{(j',s',w')} x^v_{j',i,s',t,w',w} = 0 \qquad (i,t,w) \notin \{(o_v,e_v,w_0),(d_v,l_v,w)\}, \qquad \forall v \in V \tag{4}$$

Constraints (2) and (3) ensure that each vehicle departs from the origin o_v at the planning time horizon beginning e_v with initial state w_0, and arrives at destination d_v at the end of the planning time horizon l_v. Constraint (4) guarantees the flow balance on other intermediate nodes.

Request satisfaction constraint:

$$\sum_{v\in V} \sum_{(i,j,t,s,w,w')\in \Psi_{p,v}} x^v_{i,j,t,s,w,w'} = 1 \quad \forall p \in P \tag{5}$$

Constraint (5) ensures that each customer is served exactly once.

Binary definitional constraint:

$$x^v_{i,j,t,s,w,w'} \in \{0,1\} \qquad \forall (i,j,t,s,w,w') \in A_v, \forall v \in V \tag{6}$$

Constraint (6) defines the binary variables for arc selection.

3. Dualization and augmentation, decomposition and linearization techniques for applying ADMM in VRP

In this section, we first review the background information of ADMM briefly, mainly for continuous convex problems. In our specific application of vehicle routing problems with discrete binary decision variables, the proposed modeling process contains three steps for reformulation, namely (a) dualization and augmentation, (b) decomposition and (c) linearization. As a method of multipliers, this dualization and augmentation procedure relaxes hard constraints at the cost of breaking the separable structure of problem. To address this issue, we decompose the model by utilizing the iterative principle in ADMM, and then a series of nonlinear sub-problems are obtained. In the last step, the linearization technique is used to reduce the sub-problems into a much simpler linear formulation. Each sub-problem is solved by the efficient dynamic programming algorithm with search region reduction techniques. To address the issue of curse of dimensionality, one could use a multi-label dynamical programming algorithm with state-space relaxation or dominance criteria to solve the pricing sub-problems in the typical column generation framework (Boland et al., 2006; Eilon et al., 1974).

3.1 Generic formulation of ADMM

Essentially, ADMM is an integration of augmented Lagrangian relaxation (see Appendix) and block coordinate descent methods. Consider a problem with a separable objective function and linear equality constraints of the form as (7), for example, with vehicle variables *x* and *y* in the context of VRP.

$$\begin{aligned} &\min \quad f(x) + g(y) \\ &\text{subject to } \; Ax + By - c = 0 \end{aligned} \tag{7}$$

where $x \in R^n$, $y \in R^m$ $A \in R^{p\times n}$, $B \in R^{p\times m}$ and $c \in R^P$, and $f(x)$ and $g(y)$ are assumed to be convex. The augmented Lagrangian function based on the relaxation of the constraints is shown as Eq. (8) with the corresponding dual problem (9).

$$L(x, y, \lambda, \rho) = f(x) + g(y) + \lambda^T(Ax + By - c) + \frac{\rho}{2}\|Ax + By - c\|_2^2 \tag{8}$$

$$\max \inf_{x,y} L(x, y, \lambda, \rho) \tag{9}$$

The variables x, y and multipliers λ are updated separately and sequentially by following the iterative principle (10).

$$\begin{aligned} x^{k+1} &:= \underset{x}{\operatorname{argmin}} L(x, y^k, \lambda^k, \rho) \\ y^{k+1} &:= \underset{y}{\operatorname{argmin}} L(x^{k+1}, y, \lambda^k, \rho) \\ \lambda^{k+1} &:= \lambda^k + \rho(Ax^{k+1} + By^{k+1} - c) \end{aligned} \tag{10}$$

Note that the specific dual update step length is ρ, which is different from the step length α^k in Lagrangian relaxation method (see Appendix, Eq. (A4)).

In a case with inequality constraints, we can introduce slack variable *s* to transform inequality constraints into equality constraints. That is, replacing $Ax + By - c < 0$ by $Ax + By - c = s$, s<0, leads to Eq. (11).

$$L(x, y, \lambda, \rho) = f(x) + g(y) + \lambda^T(Ax + By - c - s) + (\rho/2)\|Ax + By - c - s\|_2^2 \tag{11}$$

The necessary and sufficient optimality conditions for the ADMM model are primal feasibility and dual feasibility (Eckstein and Bertsekas, 1992), as shown in (12).

$$\begin{aligned} &\text{Primal feasibility: } \; Ax + By - c = 0 \\ &\text{Dual feasibility: } \; \nabla f(x^*) + A^T\lambda^* = 0 \\ &\qquad\qquad\qquad\quad \nabla g(y^*) + B^T\lambda^* = 0 \end{aligned} \tag{12}$$

The corresponding primal and dual residuals can be calculated as (13).

$$\begin{aligned} &\text{Primal residuals: } \; r^{k+1} = Ax^{k+1} + By^{k+1} - c \\ &\text{Dual residuals: } \; s^{k+1} = \rho A^T B(y^{k+1} - y^k) \end{aligned} \tag{13}$$

By using ρ as the step size for updating the dual variables, the second condition always holds for the iterative $(x^{k+1}, y^{k+1}, \lambda^{k+1})$; the primal residuals r^{k+1} and dual residuals s^{k+1} will convergence to zero and the primal and the first dual feasibility condition will be achieved as $k \to \infty$, in a well defined convex programming problem.

3.2 Reformulation of the VRP model

In the proposed VRP model, the coupling constraint (5) across different vehicles is now the single set of hard constraints. With a Lagrangian relaxation, we **dualize** Eq. (5) into the objective function as (14). Further, quadratic penalty terms are added to transform the proposed model into the **augmented** objective function L' as (15),

$$\min L = \sum_{v\in V}\sum_{a\in A_v} c_a x_a^v + \sum_{p\in P}\lambda_p\left(\sum_{v\in V}\sum_{a\in \Psi_{p,v}} x_a^v - 1\right) \tag{14}$$

$$\min L' = \sum_{v\in V}\sum_{a\in A_v} c_a x_a^v + \sum_{p\in P}\lambda_p\left(\sum_{v\in V}\sum_{a\in \Psi_{p,v}} x_a^v - 1\right) + \frac{\rho}{2}\sum_{p\in P}\left(\sum_{v\in V}\sum_{a\in \Psi_{p,v}} x_a^v - 1\right)^2 \tag{15}$$

subject to constraints (2),(3),(4) and (6). For notational simplicity, we denote (i, j, t, s, w, w') as a, and define a new variable μ_p^v as the total number of times that passenger p has been served by all the other vehicles except *v*, as shown in Eq. (16).

$$\mu_p^v = \sum_{v'\in V/\{v\}}\sum_{a\in \Psi_{p,v'}} x_a^{v'} \quad \forall p \in P \tag{16}$$

Obviously, the Lagrangian problem can be decomposed into several sub-problems L_v for each vehicle *v* as Eq. (17).

$$L_v = \sum_{a\in A_v} c_a x_a^v \ + \sum_{p\in P}\sum_{a\in\Psi_{p,v}} \lambda_p x_a^v \tag{17}$$

Within the block coordinate descent method inside ADMM, the above augmented problem can also be **decomposed** into several sub-problems L'_v as Eq. (18),

$$L'_v = \sum_{a\in A_v} c_a x_a^v \ + \sum_{p\in P}\sum_{a\in\Psi_{p,v}} \lambda_p x_a^v + \frac{\rho}{2}\sum_{p\in P}\left(\sum_{a\in\Psi_{p,v}} x_a^v + \mu_p^v - 1\right)^2 \tag{18}$$

In general, the quadratic term imposes a great deal of computational difficulties. However, the following proof shows that, the quadratic term in the sub-problem can be separated and regrouped as a **linearized** objective function because of the binary nature of its decision variables.

Proposition. *Each sub-problem can be reduced to a dynamic programming problem with linear objective function:*

$$L'_v = \sum_{a\in A_v} \hat{c}_a^v x_a^v \tag{19}$$

$$\hat{c}_a^v = \begin{cases} c_a + \lambda_p + \rho\mu_p^v - \dfrac{\rho}{2} & a \in \Psi_{p,v} \\ c_a & otherwise \end{cases} \tag{20}$$

Proof.

Let us start with a simple quadratic form $(x+b)^2$, where x is a binary variable and b is a constant. It is obvious that, as $x^2 = x$ because of the property of the binary variable, $(x^2 + 2xb + b^2)$ can be further reduced to $(x + 2xb + b^2) = (2b+1)x + b^2$, which is a linear function of x. Similarly, in our specific VRP applications, considering vehicle v and all the other vehicles as two parts, we can divide the quadratic term into three parts as Eq. (21),

$$\left(\sum_{a\in\Psi_{p,v}} x_a^v + \mu_p^v - 1\right)^2 = \left(\sum_{a\in\Psi_{p,v}} x_a^v\right)^2 + 2\sum_{a\in\Psi_{p,v}} x_a^v\left(\mu_p^v - 1\right) + \left(\mu_p^v - 1\right)^2 \tag{21}$$

where $\{x_a^v\}$ is the decision variable of sub-problem L_v, and μ_p^v can be calculated by Eq. (16). One can easy check that $\left(\sum_{a\in\Psi_{p,v}} x_a^v\right)^2$ is the only quadratic term. As each customer can be served at most once by one vehicle in our hyper-dimension network construction, $\sum_{a\in\Psi_{p,v}} x_a^v = \{0,1\}$ and the square of a binary variable equals to itself.

$$\left(\sum_{a\in\Psi_{p,v}} x_a^v\right)^2 = \sum_{a\in\Psi_{p,v}} x_a^v \tag{22}$$

Then we can regroup the objective function of the sub-problems for each vehicle as a linear form in Eq. (23),

$$L'_v = \sum_{a\in A_v} c_a x_a^v + \sum_{p\in P}\sum_{a\in\Psi_{p,v}} \left[\lambda_p x_a^v + \frac{\rho}{2} x_a^v\left(2\mu_p^v - 1\right)\right] + Q = \sum_{a\in A_v} \hat{c}_a^v x_a^v + Q \tag{23}$$

where $\hat{c}_a^v$ is a combined cost term inside each updating for a vehicle, and Q is a constant term.

3.3 ADMM-based solution procedure for vehicle routing problem

Based on the above three-step reformulation techniques, an ADMM-based solution procedure is employed here to solve the multi-vehicle routing problem.

//**Step 1**: Initialization

Initialize iteration number $k = 0$;

Initialize Lagrangian multipliers $\lambda_p^{k=0}$ and penalty parameter $\rho^{k=0}$;

Initialize upper bound and lower bound solutions $\{X_{LB}^0\}$ and $\{X_{UB}^0\}$;

Set the best lower bound $LB^* = -\infty$ and the best upper bound $UB^* = +\infty$.

//Step 2: Minimize augmented Lagrangian over each individual vehicle v sequentially

Step 2.1 // call the forward dynamic programming algorithm over vehicles v

For each vehicle $v \in V$

Update the arc cost $\hat{c}_a^v \in \Psi_{p,v}$ by Eq. (20);

Find the state-space-time shortest path for vehicle v (solve L'_v) by calling the forward dynamic programming algorithm;

End for

Step 2.2 // Update Lagrangian multipliers and quadratic penalty parameter

Update Lagrangian multipliers:

$\lambda_p^{k+1} := \lambda_p^k + \rho^k\left(\sum_{v\in V}\sum_{a\in\Psi_{p,v}} x_a^v - 1\right), \ \forall p \in P$;

Update the quadratic penalty parameter by:

$$\rho^{k+1} := \begin{cases} \rho^{k} + \beta & if \ \ \sum_{p\in P}(\sum_{v\in V}\sum_{a\in\Psi_{p,v}} x_a^{vk} - 1)^2 \geq \gamma \sum_{p\in P}(\sum_{v\in V}\sum_{a\in\Psi_{p,v}} x_a^{vk-1} - 1)^2 \\ \rho^{k} & otherwise \end{cases}, \text{ where } 1 \leq \beta \leq 10$$

and $0.25 \leq \gamma \leq 0.5$ can be chosen.

//**Step 3:** Generate upper bound solution and compute UB^k

//**Step 3.1:** Find feasible solution $\{X_{UB}^k\}$ for the primal problem

Adopt the passenger-to-vehicle assignment results in step 2.1

For each customer $p \in P$ do

If the customer is served by more than one vehicle, then designate one of the vehicles for him/her.

If the customer is not served by any vehicle, then assign a backup vehicle for him/her.

End for

// **Step 3.2:** Compute UB^k

Compute UB^k by substituting solution $\{X_{UB}^k\}$ in Eq. (1);

$UB^* = min\{UB^*,\ UB^k\}$.

// **Step 4:** Generate lower bound solution and compute LB^k

// **Step 4.1:** Generate lower bound solution $\{X_{LB}^k\}$ by solving the pure Lagrangian dual problem

For each vehicle $v \in V$

Solve sub-problem (17) by calling the forward dynamic programming algorithm, where the Lagrangian multipliers $\{\lambda_p^k\}$ are adopted from step 2; note that sub-problem (17) does not contain any quadratic penalty terms related to ρ, as it is a sub-problem of pure Lagrangian dual problem;

End for

// **Step 4.2:** Compute LB^k

Initialize LB^k=0;

Compute LB^k by substituting solution $\{X_{LB}^k\}$ in Eq. (14);

$LB^* = max\{LB^*,\ LB^k\}$.

//**Step 5:** Evaluate solution quality and termination condition test

Compute the relative gap between LB^* and UB^* by:

Gap= $\frac{UB^* - LB^*}{UB^*} \times 100\%$;

If the solution satisfies a convergence condition, e.g., primal residuals r^k and dual residuals s^k as shown in Eq (13) are sufficiently small, or k reaches the maximum iteration number, terminate the algorithm and output the best lower bound LB^* and best upper bound UB^*; otherwise, k:=k+1 and go back to Step 2.

A subtle but very important point to make is that, when solving the sub-problem for m^{th} individual vehicle, first, only the routing decision variables for m^{th} vehicle are being optimized at this inner iteration step; second, as shown in Fig. 2, the routes of all the other vehicles are temporally kept to be fixed, including the routes from the current k^{th} iteration for vehicle $v_1, v_2, \ldots, v_{m-1}$ and from the last $(k\text{-}1)^{th}$ iteration for vehicle $v_{m+1}, v_{m+2}, \ldots, v_n$, where n is the size of the vehicle set V. In contrast, in the standard Largrangian relaxation method, the sub-problems are solved in parallel, with a given set of multipliers.

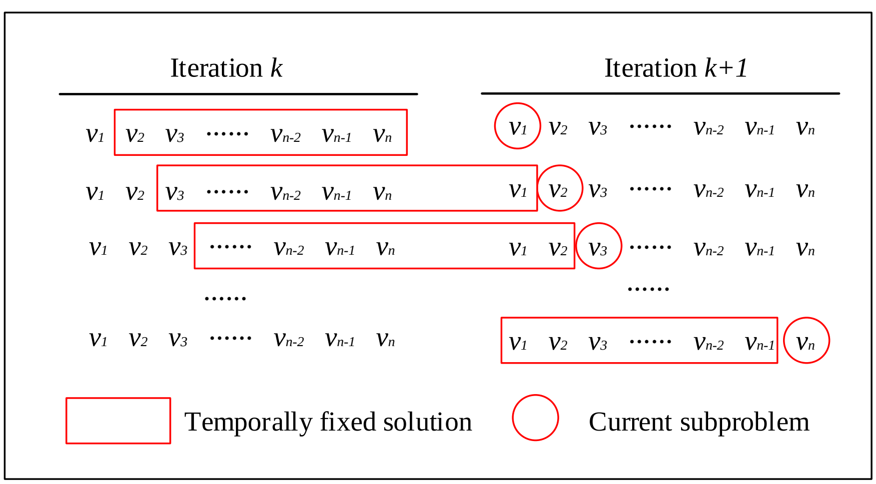


Fig. 2. Iterative pattern in the ADMM based framework for VRP.

4. Illustration example for breaking symmetry and improving the primal and dual solution

To demonstrate the symmetry breaking and computationally reliable properties of ADMM, we take a simple example to compare the proposed ADMM technique with the standard Lagrangian relaxation method. The physical network is shown as Fig. 3, in which nodes 1 and 2 represent two customers, and the cost of each link is assumed to be a constant and marked out beside the corresponding link. Each vehicle starts from original node o and finally goes back to destination d, with a set capacity of one. The other parameters and initial values include: λ_p^0=-6; ρ=2; α^0=2, $\alpha^{k+1} = 0.5\alpha^k$, and $\pi_{p,v}^k$ denotes the service price for vehicle v to serve customer p in iteration k.

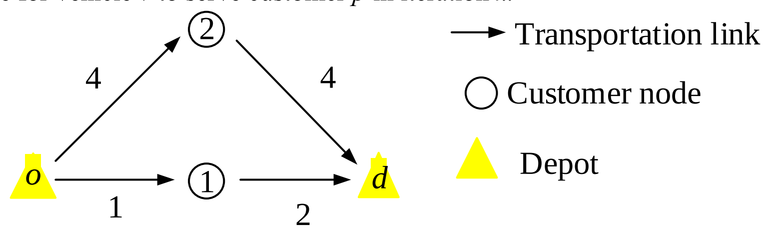


Fig. 3. A simple physical network

In the standard Lagrangian relaxation method, the original problem is decomposed into two shortest-path problems for each vehicle. The relaxed constraint is ensured by adjusting multiplier λ_p during a iterative procedure, where λ_p can be regarded as the service profit for customer p. Specifically, if a customer has not been served, his/her service profit should be increased in the next iteration to attract vehicles, vice versa. However, two vehicles could simultaneously commit to the same route as two sub-problems are identical. Only if a set of minimal cost paths could together cover each customer exactly once, a feasible and optimal solution has been found (Kohl and Madsen, 1997), or Lagrangian relaxation can just provide the lower bound estimation to primal problem.

We illustrate the graph of solution geometry (GSG) of the relaxed problem in Fig. 4, where the four corner points mapped in the plane represent four solutions detailed in Table 4. Three axes in the graph represent v_1, v_2 and L, respectively. For illustration purposes, the (discrete) surface of L is plotted in a continuous fashion. The detailed optimization process is demonstrated in Table 5, where we can see that two paths with the same cost are obtained in iteration 5.

Table 4
Details of the corner points in the graph of solution geometry.

Solution points in GSG	Assignment results		Feasibility in the primal problem	Number of times that a passenger served by vehicles	
	v_1	v_2		p_1	p_2
(1, 1)	p_1	p_1	Infeasible	2	0
(1, 2)	p_1	p_2	Feasible	1	1
(2, 1)	p_2	p_1	Feasible	1	1
(2, 2)	p_2	p_2	Infeasible	0	2

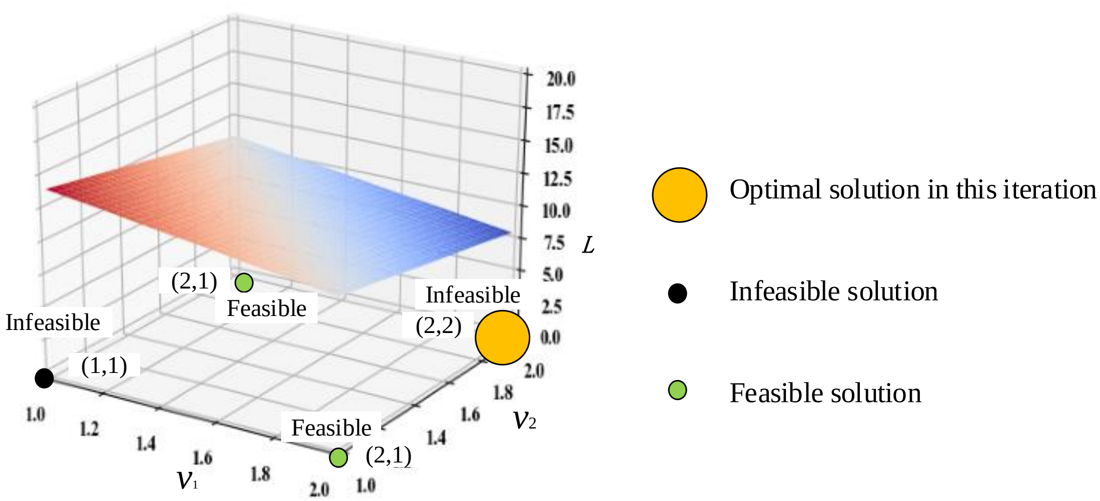
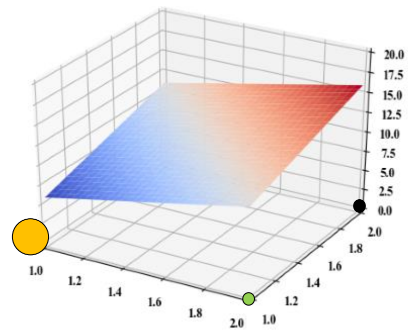
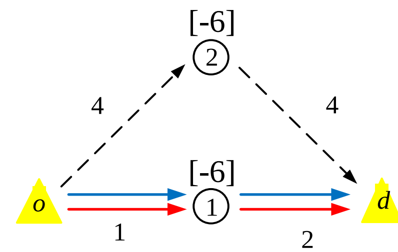
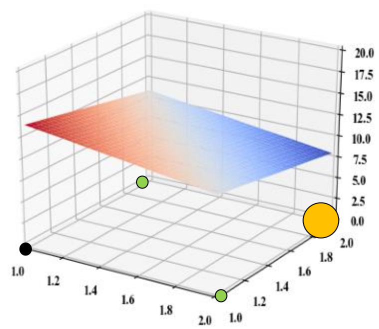
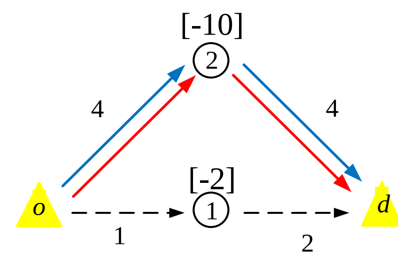
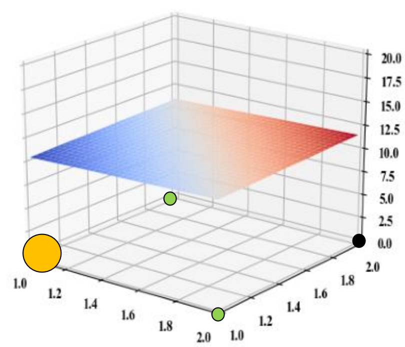
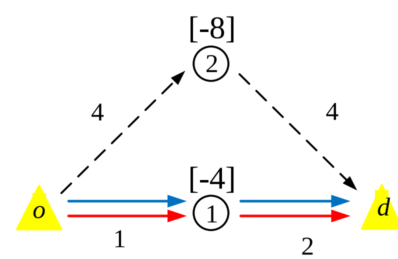
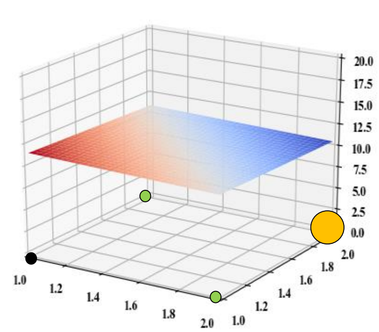
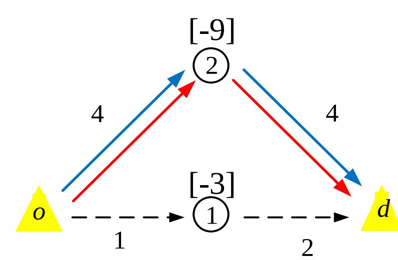
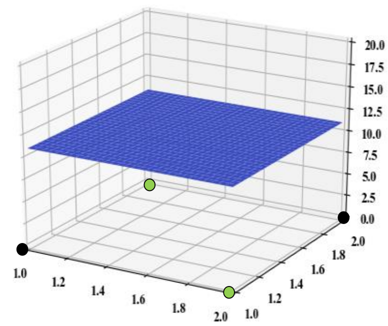
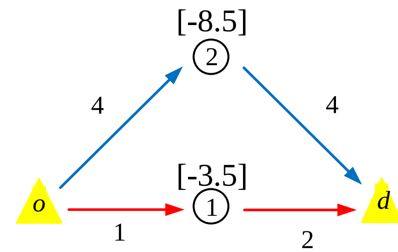


Fig.4. An example of graph of solution geometry

Table 5
Detailed optimization process using Lagrangian relaxation method.

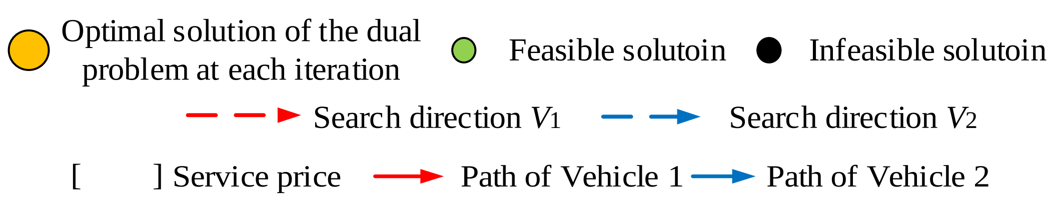
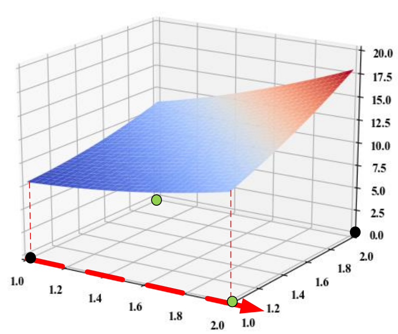
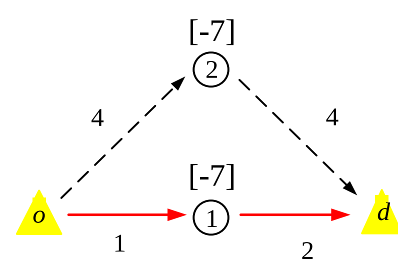
Details of computation	**Geometry of sub-problem**	**Graphical solution**
Optimal solution of the dual problem at each iteration	Feasible solutoin	Infeasible solutoin
[ ] Service price	Path of Vehicle 1	Path of Vehicle 2
Iteration 0: $\lambda_1^0 = \lambda_2^0 = -6$ $\pi_{1,1}^0 = \pi_{1,2}^0 = -\lambda_1^0$ $\pi_{2,1}^0 = \pi_{2,2}^0 = -\lambda_2^0$ The shortest path: [o-1-d] $L_1 = 1 + 2 - \pi_{1,1}^0 = -3$ $L_2 = 1 + 2 - \pi_{1,2}^0 = -3$ $L = 6$		
Iteration 1: $\alpha^0 = 4$ $\lambda_1^1 = -2, \lambda_2^1 = -10$ $\pi_{1,1}^1 = \pi_{1,2}^1 = -\lambda_1^1$ $\pi_{2,1}^1 = \pi_{2,2}^1 = -\lambda_2^1$ The shortest path: [o-2-d] $L_1 = 4 + 4 - \pi_{2,1}^1 = -2$ $L_2 = 4 + 4 - \pi_{2,2}^1 = -2$ $L = 8$		
Iteration 2: $\alpha^1 = 2$ $\lambda_1^2 = -4, \lambda_2^2 = -8$ $\pi_{1,1}^2 = \pi_{1,2}^2 = -\lambda_1^2$ $\pi_{2,1}^2 = \pi_{2,2}^2 = -\lambda_2^2$ The shortest path: [o-1-d] $L_1 = 1 + 2 - \pi_{1,1}^2 = -1$ $L_2 = 1 + 2 - \pi_{1,2}^2 = -1$ $L = 10$		
Iteration 3: $\alpha^2 = 1$ $\lambda_1^3 = -3, \lambda_2^3 = -9$ $\pi_{1,1}^3 = \pi_{1,2}^3 = -\lambda_1^3$ $\pi_{2,1}^3 = \pi_{2,2}^3 = -\lambda_2^3$ The shortest path: [o-2-d] $L_1 = 4 + 4 - \pi_{2,1}^3 = -1$ $L_2 = 4 + 4 - \pi_{2,2}^3 = -1$ $L = 10$		

Iteration 4: $\alpha^3 = 0.5$ λ_1^4 =-3.5, λ_2^4 =-8.5 $\pi_{1,1}^4 = \pi_{1,2}^4 = -\lambda_1^4$ $\pi_{2,1}^4 = \pi_{2,2}^4 = -\lambda_2^4$ The shortest paths: [o-1-d], [o-2-d] L_1=1+2$-\pi_{1,1}^4$ =-0.5, L_2=4+4$-\pi_{2,2}^4$ =-0.5 $L = 11$		

Compared to Lagrangian relaxation, the dualized problem in ADMM has an additional quadratic penalty, which can differentiate the combined costs used in different but homogeneous sub-problems and mitigate the symmetry issue. In details, as shown in Eq. (20), the service profit for each vehicle to serve customer p not only depends on Lagrangian multiplier λ_p but also the penalty parameter ρ. That is, a vehicle would be discouraged (but not prohibited) from serving a customer if he/she has already been served by other vehicles. Interestingly, one connection with the primal solution based heuristics is that, when the value of ρ is set to infinity, the vehicle would seek to avoid violating any feasibility constraint and search for a completely feasible solution. The detailed optimization process is listed in Table 6. One can see that the optimal solution is obtained in the second iteration, which shows faster convergence compared with the Lagrangian relaxation method.

Table 6

Detailed optimization process of ADMM approach.

Details of computation	**Graph of sub-problems**	**Graphical Solution**
		
Iteration 0: $\lambda_1^0 = \lambda_2^0 = 6$ $\rho = 2$ **Step 1:** $\pi_{1,1}^0 = -\lambda_1^0 + \frac{\rho}{2}$=7 $\pi_{2,1}^0 = -\lambda_2^0 + \frac{\rho}{2}$=7 The shortest path: [o-1-d] L'_1=1+2$-\pi_{1,1}^0$=-4		

Step 2:

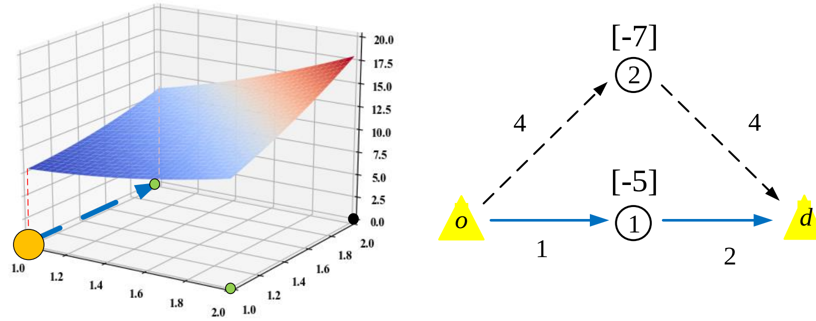
$\pi^0_{1,2} = -\lambda^0_2 - \frac{\rho}{2}$=5,

$\pi^0_{2,2} = -\lambda^0_2 + \frac{\rho}{2}$=7

The shortest path: [o-1-d]

L'_2=1+2$-\pi^0_{1,2}$=-2

$L' = 10$



Iteration 1:

$\lambda^1_1 = 4$, $\lambda^1_2 = 8$

$\rho = 2$

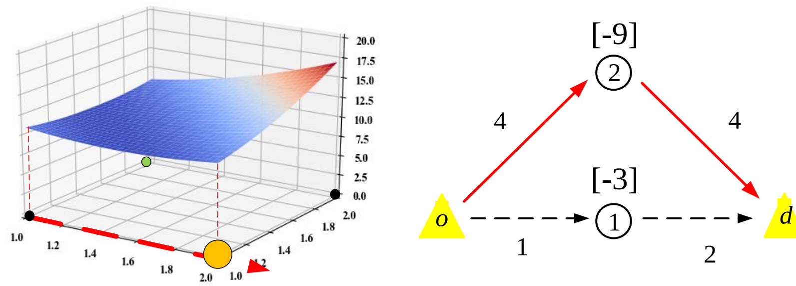
Step 1:

$\pi^1_{1,1} = -\lambda^1_1 - \frac{\rho}{2}$=3,

$\pi^1_{2,1} = \lambda^1_2 + \frac{\rho}{2} = 9$

The shortest path: [o-2-d]

L'_1=4+4$-\pi^1_{2,1}$=-1



Step 2:

$\pi^1_{1,2} = -\lambda^2_1 + \frac{\rho}{2}$=5,

$\pi^1_{2,2} = -\lambda^2_2 - \frac{\rho}{2} = 7$

The shortest path: [o-1-d]

L'_2=1+2$-\pi^1_{1,2}$ =-2

L' =11

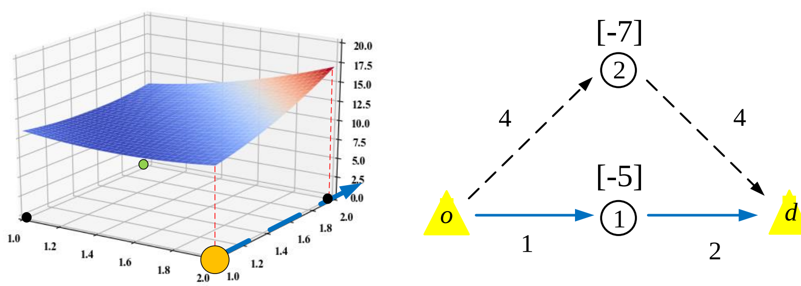
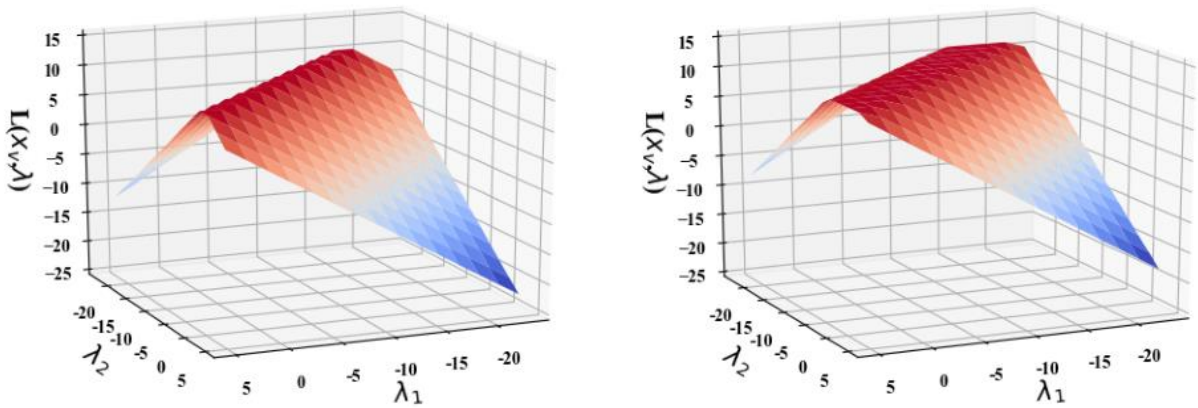
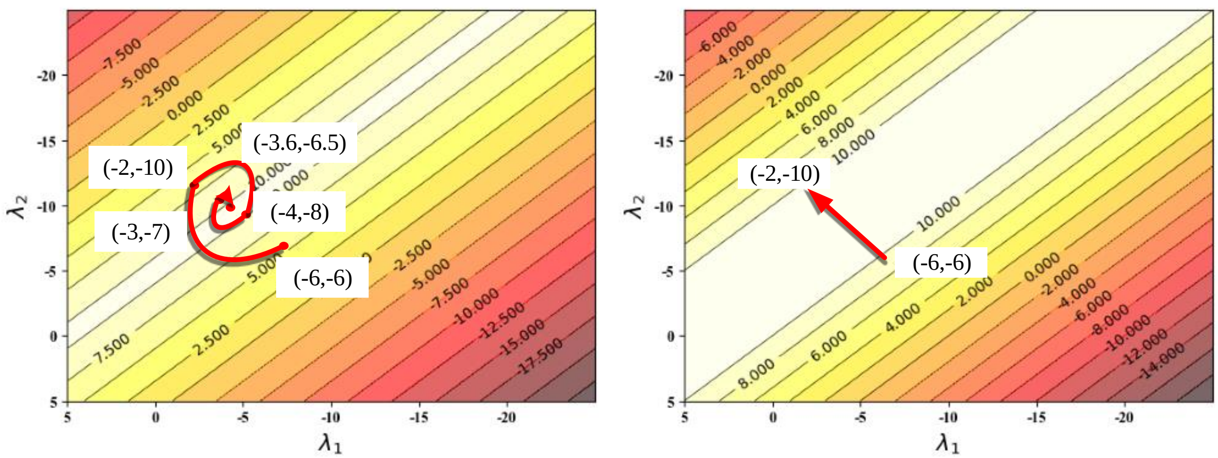


Fig. 5 further visualizes the surfaces and contours for the dual problem of Lagrangian and augmented Lagrangian respectively. The optimal region in surface of the Lagrangian dual problem is a narrow edge (Fig. 5 (a)); in augmented Lagrangian dual problem (see Fig. 5 (b)), the surface is smoother and has a broadened area of the optimal searching region thanks to the quadratic penalties. The (empirical) convergence trajectory is visualized in the contour maps shown in Fig. 5 (c) and 5 (d). The solutions of Lagrangian dual problem oscillate across the optimal region even with slight changes of the multipliers. A more reliable convergence of ADMM is shown in Fig. 5 (d), where the multiplier zigzagging issues are much reduced and well-managed, and the solution reaches the optimal solution with less iterations.



(a) surface of Lagrangian problem (b) surface of augmented Lagrangian problem



(c) contour and solution process of LR (d) contour and solution process of ADMM

Fig. 5. The surface, contour and empirical convergence process for dual problem of Lagrangian and ADMM.

5. Discussions

5.1 Discussions on theoretical convergence

The theoretical convergence of ADMM for convex solution sets has been discussed in many studies (Gabay, 1983; Eckstein and Bertsekas, 1992), a general theoretical result given by Boyd et al. (2011) shows that the residual and objective convergence can be proved under the following two assumptions.

Assumption 1: the function of original problem f and g: $R^n \rightarrow R \cup \{+\infty\}$ are closed, proper, and convex.

Assumption 2: The unaugmented (pure) Lagrangian L has a saddle point.

Researchers also recognized that there was no guarantee for the convergence of ADMM when it is applied to nonconvex problems. In our study, the problem is clearly nonconvex due to the nature of integer programming. Specifically, the original problem could have multiple optimal solutions with the same objective value. In other words, although we can reach one of the exact optimal solution(s) in each x-update and y-update steps through the dynamic programming algorithm, the convergence of ADMM is not guaranteed, which means that the different value of parameters may lead to different local optimal solutions (and in particular, not the global optimal solution). As a result, the ADMM is considered as heuristics motivated from the primal and dual solution framework or an improved version of sub-gradient method. To address this issue, recent studies (Boland et al., 2018; Gade et al., 2016) have presented effective methods for computing lower bounds in the Progressive Hedging Algorithm (PHA). Although they are written for stochastic multi-stage mixed-integer programs, the concepts and methods provide important theoretical references for us to further tackle transportation optimization problems in context of dual decomposition with consensus constraints. In our study, we solve a serial of pure Lagrangian problems to compute a lower bound on the optimal objective function by using dual prices, which is obtained during the iteration of ADMM. In this way, the upper and lower bounds can be provided simultaneously at each iteration of ADMM.

In summary, the ADMM is not an exact algorithm for nonconvex problems. Future research could further deploy a Branch and Price solution search paradigm to create mutually exclusive sub-problems and fully close the optimality gaps. On the other hand, the main advantages of our proposed method could be highlighted in the following: (a) the dual multipliers can provide system-wide price information; (b) the quadratic penalties can manage the solution search process in the dualized problem to better reach primal feasibility; (c) the LR-based lower bound of optimal solution is available. These benefits enable ADMM to have better solution optimality measures compared to the other heuristic methods.

5.2 ADMM for other broader classes of constraints

The proposed ADMM based framework can also apply to transportation optimization models with broader classes of constraints, which can be examined in the following categories.

(1) Consensus constraints

The term of consensus is typically referred to the decision across different agents or decision makers. Specifically, in our proposed VRPTW model, the service request satisfaction constraint is the consensus constraints for vehicles. This kind of constraints is widely used in other transportation problems as well, for examples, the headway constraint in railway timetable problems, and the road capacity constraints in traffic assignment problems.

(2) Consistency constraints

The consistency should be ensured across different decision and variable spatial and temporal representation layers. In a broader set of scheduled transportation system optimization, the ADMM-based framework can be useful in providing a theoretically sound and numerically reliable algorithmic foundation, for iteratively optimizing decision variables in different layers. For example, in the train service plan optimization problem, consistency constraints across layers are used to integrate three types of decisions: (a) passenger service selection behavior, (b) service plan and timetabling and (c) track and rolling stock capacity utilization. It is interesting to examine how to jointly coordinate the decisions with the help of the Lagrangian multipliers and quadratic penalties. Along this line, a recent noticeable study by Liu and Dessouky (2018) investigated the integrated scheduling of freight and passenger trains, where the train precedence sub-problem and the train routing sub-problem are innovatively decomposed and dualized.

(3) Non-anticipative constraints

The non-anticipative constraints, in the context of stochastic programming such, are enforced between the first static stage and second dynamic stage (Boland et al., 2018). Consider a scenario-based stochastic programming problem, which can be reformulated as corresponding deterministic equivalent as (24),

$$\min \sum_{s\in S} p_s(cx_s + q_s y_s) \tag{24}$$

subject to, $x_s = z, \forall\ (x_s, y_s) \in K_s, s \in S$

where S is the set of scenarios, and constraint $x_s = z$ enforces non-anticipativity for first-stage decisions. The augmented Lagrangian function with relaxation of the non-anticipativity constraints is (25),

$$L_s^{\rho} = cx_s + q_s y_s + \omega_s(x_s - \mathrm{z}) + (\rho/2)\|x_s - \mathrm{z}\|_2^2 \tag{25}$$

which can be decomposed into independent sub-problems for each scenario and solved by using the iterative principle of ADMM.

(4) Pair-based consensus constraint

In the decentralized optimization problem, the consensus constraint is established between a pair of adjacent agents, through the use of limited communication resources (Ling and Ribeiro, 2014; Tsianos et al., 2012). Consider a separable problem which minimizes the sum of local cost functions in a multi-agent network as (26),

$$\min \quad f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x) \tag{26}$$

subject to, $x_i = z_{ij}$, $x_j = z_{ij}$, $\forall (i, j) \in \mathcal{A}$

where $\mathcal{A}$ is a set of agent pairs, the variable z_{ij} is introduced to enforce the consensus between agents i and j. Using the definitions of the $\mathcal{A}$, the constraints of (26) can further be reformulated as (27).

$$\text{subject to, } Ax + Bz = 0 \tag{27}$$

The augmented Lagrangian function of original problem can be expressed as Eq. (28).

$$L(x, z, \lambda, \rho) = f(x) + g(z) + \lambda^{\mathrm{T}}(Ax + Bz - c) + (\rho/2)\|Ax + Bz - c\|_2^2 \tag{28}$$

Then the problem can be solved by using the iterative principle of ADMM.

In the following discussion, we use a detailed example to illustrate the application of ADMM in another type of VRP model with consistency constraint. First, we specifically reformulate the proposed VRP model to create an assignment-and-routing mapping by a variable spitting technique (Fisher et al., 1997; Niu et al., 2018). The consistency constraint between assignment and routing, to be dualized using our problem decomposition framework, can better help us use Lagrangian multipliers to coordinate the tasks of assigning vehicles to multiple passengers. In details, we introduce an additional assignment decision variable y_p^v, = 1if customer p is served by vehicle v; =0 otherwise, and replace constraint Eq. (5) by following Eq. (29) , Eq. (30) and Eq. (31).

Request assignment constraint:

$$\sum_{v\in V} y_p^v = 1 \quad \forall p \in P \tag{29}$$

Consistency constraint between assignment and routing:

$$\sum_{(i,j,t,s,w,w')\in\Psi_{p,v}} x_{i,j,t,s,w,w'}^v = y_p^v \quad \forall p \in P, v \in V \tag{30}$$

Binary definitional constraint:

$$y_p^v \in \{0,1\} \tag{31}$$

Constraint (29) ensures that each customer is visited by a single vehicle exactly once. Constraint (30) is the consensus constraint between detailed vehicle routing decisions and assignment decisions. Constraint (31) defines the binary variables for assignment. Within the proposed ADMM decomposition framework, the constraint (29) is dualized into the objective function, and then the relaxed model can be decomposed as two sub-problems, both involving binary decision variables: (a) a generalized assignment sub-problem to find the passenger-to-vehicle assignment and (b) a time-dependent shortest path problem to detail the vehicle routes. These two sub-problems in different decision level are solved alternately and iteratively as the pattern shown in Fig. 6. This ADMM-based decomposition framework across different decision levels also presents additional opportunities for re-interpreting and refining many efficient heuristics from a primal and dual perspective, such as savings algorithm (Clarke and Wright, 1964) and sweep based algorithm (Gillet and Miller, 1974).

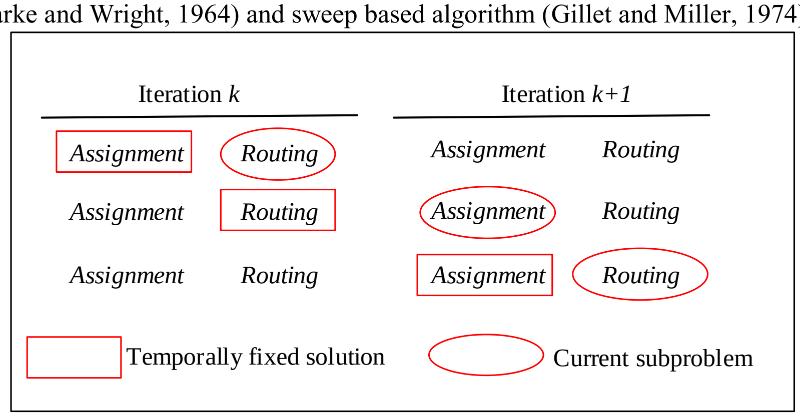


Fig. 6. Iterative pattern in the ADMM based framework for assignment and routing problem.

6. Numerical examples

The algorithm proposed in the paper was implemented in the Python platform and evaluated on a personal Windows computer with a 1.70 GHz CPU and 8 GB memory. In this section, we examine the performance of our model and proposed ADMM solution framework on two sets of experiments, including a set of Solomon VRP benchmark instances (Solomon, 1987) and a real-world instance from Jingdong logistics, a major logistics company affiliated to one of global online retailers, JD.com. This algorithmic framework can be also easily extended and applied to other test data sets and problems, and our open-source Python code is provided at https://github.com/YaoYuBJTU/ADMM_Python. For example, an implementation using C++ could dramatically improve the computational speed, and our related study for large-scale cyclic timetabling problem can be found at https://www.researchgate.net/profile/Yongxiang_Zhang7.

6.1 Solomon benchmark instances

The well-known Solomon benchmark instances highlight several factors that affect the behavior of routing and scheduling algorithms, including geographical data, the number of customers serviced by a vehicle, percent of time-constrained customers, and tightness and positioning of the time windows. Three different classes of instances (C, R, and RC, respectively) are contained in the Solomon dataset, where geographical data is clustered in set C, randomly generated in set R and a mix of random and clustered structures in set RC.

Our solution framework is evaluated on date set C101, R101 and RC101 with different scales, specially, for 25, 50 and 100 customers respectively. The fleet size of vehicles is set as 25, with each vehicle capacity of 200. The details of comparison between ADMM solutions and the best known solutions are shown in Table 7.

For the 25-customer cases, the good upper bound solution can be obtained within 5 iterations. The average number of iterations for reaching a good upper bound in the 50-customer and 100-customer cases is about 25 and 50, respectively. In the data set C101, our solutions are very close to the optimal solutions, with relative gaps as 0.26%, 0.22% and 1.77% for different scales. The relative gaps are reported within 5% in R101, and within 10% in RC101. The computational time increases with the scale of problems but within 150s in our largest cases. Overall, the results show that our proposed ADMM framework can obtain good solutions efficiently.

Table 7
Comparison between ADMM solutions and the best known solutions.

Dataset	Number of customers	Best known Solution	ADMM Solution	Computing time	Gap between best known and ADMM solution
R101	25	617.1	634.9	0.34s	2.80%
	50	1044.0	1083.6	3. 52s	3.65%
	100	1637.7	1724.9	48.49s	5.01%
C101	25	191.3	191.8	1.55s	0.26%
	50	362.4	363.2	12.11s	0.22%
	100	827.3	842.2	106.28s	1.77%
RC101	25	461.1	470.7	3.08s	2.04%
	50	944	1012.5	21.76s	6.71%
	100	1619.8	1761.8	143.26s	8.06%

6.2 The real-world case study

We also test our solution framework on the data set extracted from the optimization competition launched by Jingdong Logistics. Focusing on smart logistics and supply chain, one of the competition topics is an urban truck routing and scheduling problem; and numerous real data based on the company's B2B delivery scenarios in Beijing has been provided. More details and the dataset of the competition can be obtained from the website: https://jdata.joybuy.com/en/.

As shown in Fig. 6, we select a subset of customer data from the competition data set as our test case, where the distribution center provides urban distribution services for 100 customers in the urban area. Besides, several assumptions considered in this data set are presented as follows:

(1) Each vehicle starts working after 8:00 and going back to the distribution center before 00:00.
(2) The fixed usage charge for each vehicle is 200 RMB/day, the transportation cost is 12 RMB/km and the waiting cost is 24 RMB/hour. In addition, the number of backup vehicles is assumed to be sufficient.
(3) The weight and volume capacity of each vehicle is 12,000 kg and 12m^3.
(4) Each order or customer has a particular time window.
(5) The travel time of each link is given as a constant based on the real-world road network.
(6) The service time for each customer is set as 30 min.

The weighted objective to be minimized includes the transportation cost, the waiting cost and the fixed vehicle cost.

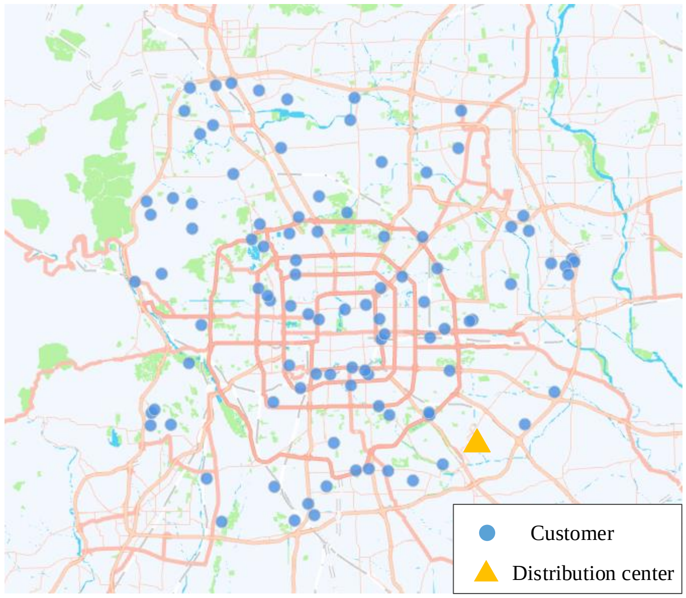


Fig. 6. The distribution center and customers in Beijing road network.

The proposed ADMM based algorithm is performed 100 iterations to solve this case, and the computing time for each iteration is about 35s. We set the parameter β as 3, γ as 0.25, ρ as 2 and the initial Lagrangian multipliers λ_p as 180.

The evolution trends of the lower and upper bounds are demonstrated in Fig. 7. We can see that the upper bound solutions improve significantly in the first 26 iterations, and obtain a stable good value in 73th iteration with the final gap of 11.79%. The lower bound also has tightened continuously in the first 20 iterations.

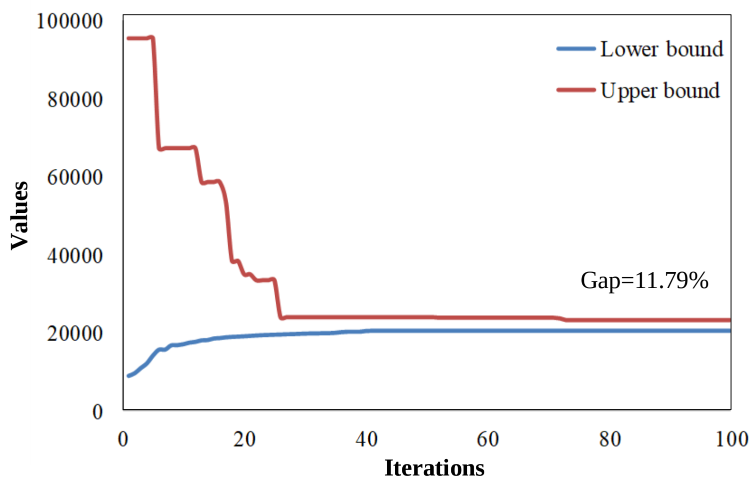


Fig. 7. Evolution curves of the upper bounds and lower bounds.

In the best upper bound solution, twelve vehicles are needed to serve all the given customers, with the total waiting time as 103 min and a total distance of 1659.19km. The details of the optimized result are shown in Table 8, the bold number in the second column represents the vehicle needs to wait until the time window beginning. The routes for vehicles are further illustrated in Fig. 8.

Table 8
The Details of the Optimized Solution.

Vehicle id	Vehicle routing solution	Waiting time (min)	Travel distance (km)
1	0-98-61-16-26-39-51-**13**-96-76-100	1	202.99
2	0-7-45-18-40-85-93-41-33-86-74-100	0	175.48
3	0-4-58-36-63-99-97-**2**-11-27-28-100	8	164.65
4	0-31-59-88-78-71-23-91-47-43-100	0	105.55
5	0-9-22-15-34-5-69-68-87-55-100	0	121.44
6	0-57-54-17-84-79-38-95-80-24-100	0	78.88

7	0-21-70-60-32-6-**82**-29-46-90-100	9	181.25
8	0-25-52-1-50-56-65-19-77-12-00	0	170.42
9	0-20-42-35-73-**53**-66-44-100	36	155.41
10	0-14-**37**-64-92-**94**-8-100	22	83.74
11	0-62-67-**48**-72-**3**-75-49-83-100	9	137.86
12	0-30-81-**10**-89-100	18	81.52
Total	-	103	1659.19
Average	-	8.58	138.27

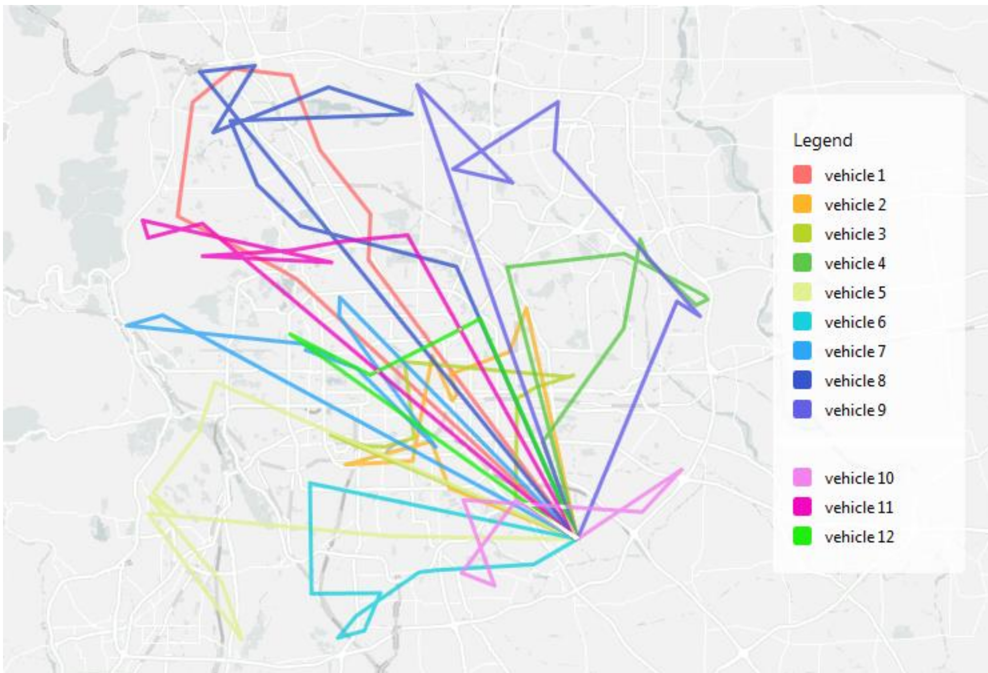


Fig. 8. The optimized routes and schedules for vehicles.

7. Conclusions

In this paper, focusing on the consensus constraints in vehicle routing problems, we proposed an ADMM based decomposition framework to iteratively improve primal and dual solution quality simultaneously. To demonstrate the broader benefit of our proposed algorithmic framework, we demonstrate how a wide range of transportation optimization models with consensus/consistency constraints can be reformulated and solved efficiently, as long as the decomposed sub-problem involves binary decision variables in a framework of block coordinate descents and augmented Lagrangian. For the VRP problem, firstly, by using a state-space-time network diagram, we constructed a multi-dimensional commodity flow formulation for VRP, where time window and vehicle capacity constraints are embedded to simplify the dual process. Then, ADMM was particularly introduced in order to develop a reliable decomposition framework, in which the original model was decomposed into a serial of shortest path problems through three steps, namely dualization and augmentation, decomposition and linearization, and sub-problem solving by the dynamic programming algorithm. To measure the quality of the solutions, a lower bound estimate was established by solving a pure Lagrangian relaxation problem.

To examine the effectiveness of the proposed ADMM based framework, two sets of numerical examples were implemented, including a set of Solomon benchmark instances and a real-world instance provided by Jingdong Logistics. The computational results showed that, ADMM-based approach is able to efficiently obtain good quality solutions with relative tight lower bounds.

In summary, the proposed ADMM framework offers a relatively simpler and reliable algorithmic implementation structure, which can be commonly used in solving many other transportation problems as well. Our future research will focus on the following two major aspects. (1) Extend the ADMM framework for other more complicated transportation problems, for instance, the problem with multi decision variables across layers. (2) Add the branch techniques into the framework to improve the solution quality and relative gap between lower bound and upper bound.

Appendix A: Formulations of Lagrangian relaxation and augmented Lagrangian method

For the sake of completeness, in this appendix we review the basic framework of Lagrangian relaxation method and augmented Lagrangian method briefly.

Consider a linear equality-constrained problem expressed as (A1),

$$\begin{aligned} &\min \quad f(x) \\ &\text{subject to } \; Ax - c = 0 \end{aligned} \tag{A1}$$

where $x \in R^n$, $A \in R^{p\times n}$, and $c \in R^P$, and $f(x)$ are assumed to be convex. The Lagrangian relaxation form for problem (A1) can be written as Eq. (A2) and the Lagrangian dual problem is shown as Eq. (A3).

$$L(x,\lambda) = f(x) + \lambda^T(Ax - c) \tag{A2}$$

$$\max \inf_x L(x,\lambda) \tag{A3}$$

Explicitly, the dual problem can be solved through the following step (A4),

$$\begin{aligned} &x^{k+1} := \operatorname*{argmin}_x L(x,\lambda^k) \\ &\lambda^{k+1} := \lambda^k + \alpha^{k+1}\,(Ax^{k+1} - c) \end{aligned} \tag{A4}$$

where λ is the Lagrange multiplier for constraint $Ax - c = 0$ and α is the step size for updating λ. In the general case, after each iteration the equality constraint residual contributions $(Ax^k - c)$ are required to be collected in order to compute the Lagrangian multipliers of the next iteration. The main advantage of Lagrangian relaxation method is that each *x*-minimization step can be split into separate sub-problems and solved in parallel when $f(x)$ is separable.

By adding a quadratic penalty, the augmented Lagrangian function is shown as Eq. (A5),

$$L(x,\lambda,\rho) = f(x) + \lambda^T(Ax - c) + (\rho/2)\|Ax - c\|_2^2 \tag{A5}$$

where ρ is the parameter of the quadratic penalty. The standard augmented Lagrangian method would minimize $L(x,\lambda,\rho)$ with respect to each sub-vector of *x* jointly.

Acknowledgements

We thank Prof. Natashia Boland and Prof. Martin Savelsbergh at Georgia Institute of Technology for their valuable comments. The first two authors of this paper are supported by the Fundamental Research Funds for the Central Universities (2018YJS090). The last author is partially funded by National Science Foundation-United States under NSF Grant No. CMMI 1538105 “Collaborative Research: Improving Spatial Observability of Dynamic Traffic Systems through Active Mobile Sensor Networks and Crowdsourced Data” and NSF Grant No. CMMI 1663657. “Real-time Management of Large Fleets of Self-Driving Vehicles Using Virtual Cyber Tracks”.

References

Altinkemer, K., Gavish, B., 1991. Parallel savings based heuristics for the delivery problem. *Operations Research* 39(3), 456-469.

Barnhart, C., Johnson, E.L., Nemhauser, G.L., Savelsbergh, M.W., Vance, P.H., 1998. Branch-and-price: Column generation for solving huge integer programs. *Operations research* 46(3), 316-329.

Benders, J.F., 1962. Partitioning procedures for solving mixed-variables programming problems. *Numerische mathematik* 4(1), 238-252.

Bertsekas, D.P., 1999. *Nonlinear programming*. Athena scientific Belmont.

Boland, N., Christiansen, J., Dandurand, B., Eberhard, A., Linderoth, J., Luedtke, J., Oliveira, F., 2018. Combining Progressive Hedging with a Frank--Wolfe Method to Compute Lagrangian Dual Bounds in Stochastic Mixed-Integer Programming. *SIAM Journal on Optimization* 28(2), 1312-1336.

Boland, N., Dethridge, J., Dumitrescu, I., 2006. Accelerated label setting algorithms for the elementary resource constrained shortest path problem. *Operations Research Letters* 34(1), 58-68.

Boland, N., Hewitt, M., Marshall, L., Savelsbergh, M., 2017. The continuous-time service network design problem. *Operations Research* 65(5), 1303-1321.

Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning* 3(1), 1-122.

Boyd, S., Vandenberghe, L., 2004. *Convex optimization*. Cambridge university press.

Bradley, S.P., Hax, A.C., Magnanti, T.L., 1977. Applied mathematical programming.

Christofides, N., Mingozzi, A., Toth, P., 1981. Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations. *Mathematical programming* 20(1), 255-282.

Clarke, G., Wright, J.W., 1964. Scheduling of vehicles from a central depot to a number of delivery points. *Operations research* 12(4), 568-581.

Cordeau, J.-F., 2006. A branch-and-cut algorithm for the dial-a-ride problem. *Operations Research* 54(3), 573-586.

Dabia, S., Ropke, S., Van Woensel, T., De Kok, T., 2013. Branch and price for the time-dependent vehicle routing problem with time windows. *Transportation science* 47(3), 380-396.

Dantzig, G.B., Wolfe, P., 1960. Decomposition principle for linear programs. *Operations research* 8(1), 101-111.

Desrochers, M., Desrosiers, J., Solomon, M., 1992. A new optimization algorithm for the vehicle routing problem with time windows. *Operations research* 40(2), 342-354.

Eckstein, J., Bertsekas, D.P., 1992. On the Douglas—Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming* 55(1-3), 293-318.

Eilon, S., Watson-Gandy, C., Christofides, N., de Neufville, R., 1974. Distribution management-mathematical modelling and practical analysis. *IEEE Transactions on Systems, Man, and Cybernetics*(6), 589-589.

Fisher, M.L., Jörnsten, K.O., Madsen, O.B., 1997. Vehicle routing with time windows: Two optimization algorithms. *Operations research* 45(3), 488-492.

Fisher, M.L., Jaikumar, R., 1981. A generalized assignment heuristic for vehicle routing. *Networks* 11(2), 109-124.

Ford Jr, L.R., Fulkerson, D.R., 1958. A suggested computation for maximal multi-commodity network flows. *Management Science* 5(1), 97-101.

Fukasawa, R., Longo, H., Lysgaard, J., de Aragão, M.P., Reis, M., Uchoa, E., Werneck, R.F., 2006. Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Mathematical programming* 106(3), 491-511.

Gade, D., Hackebeil, G., Ryan, S.M., Watson, J.-P., Wets, R.J.-B., Woodruff, D.L., 2016. Obtaining lower bounds from the progressive hedging algorithm for stochastic mixed-integer programs. *Mathematical Programming* 157(1), 47-67.

Ghilas, V., Demir, E., Van Woensel, T., 2016. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows and scheduled lines. *Computers & Operations Research* 72, 12-30.

Gillett, B.E., Miller, L.R., 1974. A heuristic algorithm for the vehicle-dispatch problem. *Operations research* 22(2), 340-349.

Glowinski, R., Marroco, A., 1975. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires. *Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique* 9(R2), 41-76.

Goel, A., Gruhn, V., 2005. Large Neighborhood Search for rich VRP with multiple pickup and delivery locations, *Proceedings of the 18th Mini EURO Conference on VNS (MEC-VNS)*.

Held, M., Karp, R.M., 1970. The traveling-salesman problem and minimum spanning trees. *Operations Research* 18(6), 1138-1162.

Kallehauge, B., 2008. Formulations and exact algorithms for the vehicle routing problem with time windows. *Computers & Operations Research* 35(7), 2307-2330.

Kohl, N., Madsen, O.B., 1997. An optimization algorithm for the vehicle routing problem with time windows based on Lagrangian relaxation. *Operations research* 45(3), 395-406.

Laporte, G., Nobert, Y., Desrochers, M., 1985. Optimal routing under capacity and distance restrictions. *Operations research* 33(5), 1050-1073.

Lasdon, L.S., 2002. *Optimization theory for large systems*. Courier Corporation.

Ling, Q., Ribeiro, A., 2014. Decentralized linearized alternating direction method of multipliers, *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, pp. 5447-5451.

Liu, L., Dessouky, M., 2018. Stochastic passenger train timetabling using a branch and bound approach. *Computers & Industrial Engineering*.

Mahmoudi, M., Zhou, X., 2016. Finding optimal solutions for vehicle routing problem with pickup and delivery services with time windows: A dynamic programming approach based on state–space–time network representations. *Transportation Research Part B: Methodological* 89, 19-42.

Nedic, A., Ozdaglar, A., 2009. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control* 54(1), 48-61.

Nemhauser, G.L., Savelsbergh, M.W.P., Sigismondi, G.C., 1991. *Constraint classification for mixed integer programming formulations*. Department of Mathematics and Computing Science, University of Technology.

Nishi, T., Ando, M., Konishi, M., 2005. Distributed route planning for multiple mobile robots using an augmented Lagrangian decomposition and coordination technique. *IEEE Transactions on Robotics* 21(6), 1191-1200.

Niu, H., Zhou, X., Tian, X., 2018. Coordinating assignment and routing decisions in transit vehicle schedules: A variable-splitting Lagrangian decomposition approach for solution symmetry breaking. *Transportation Research Part B: Methodological* 107, 70-101.

Reimann, M., Doerner, K., Hartl, R.F., 2004. D-ants: Savings based ants divide and conquer the vehicle routing problem. *Computers & Operations Research* 31(4), 563-591.

Renaud, J., Boctor, F.F., 2002. A sweep-based algorithm for the fleet size and mix vehicle routing problem. *European Journal of Operational Research* 140(3), 618-628.

Ruszczyński, A., 1989. An augmented Lagrangian decomposition method for block diagonal linear programming problems. *Operations Research Letters* 8(5), 287-294.

Savelsbergh, M., Van Woensel, T., 2016. 50th anniversary invited article—city logistics: Challenges and opportunities. *Transportation Science* 50(2), 579-590.

Solomon, M.M., 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints.

Operations research 35(2), 254-265.

Taillard, É., 1993. Parallel iterative search methods for vehicle routing problems. *Networks* 23(8), 661-673.

Tong, L.C., Zhou, L., Liu, J., Zhou, X., 2017. Customized bus service design for jointly optimizing passenger-to-vehicle assignment and vehicle routing. *Transportation Research Part C: Emerging Technologies* 85, 451-475.

Tsianos, K.I., Lawlor, S., Rabbat, M.G., 2012. Consensus-based distributed optimization: Practical issues and applications in large-scale machine learning, *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*. IEEE, pp. 1543-1550.

Wolsey, L., Rinaldi, G., 1993. Integer programming and combinatorial optimization.

Zhou, X., Tong, L., Mahmoudi, M., Zhuge, L., Yao, Y., Zhang, Y., Shang, P., Liu, J., Shi, T., 2018. Open-source VRPLite Package for Vehicle Routing with Pickup and Delivery: A Path Finding Engine for Scheduled Transportation Systems. *Urban Rail Transit* 4(2), 68-85.