RAIRO-Oper. Res. 60 (2026) 1757–1791
https://doi.org/10.1051/ro/2026045

RAIRO Operations Research
www.rairo-ro.org

AN ITERATIVE SOLUTION APPROACH TO CAPACITATED TWO-STAGE TIME MINIMIZATION TRANSPORTATION PROBLEM WITH DESTINATION-STAGE DEMAND SPECIFIED

Lijun Zhou[1] and Fanrong Xie[1,2,3,*]

Abstract. Because of realistic necessity for timely satisfying destination-stage demand, *capacitated two-stage time minimization transportation problem with destination-stage demand specified* (C2SD) is crucial and important optimization problem with no research report in literature due to its super intractability. In this paper, by creating C2SD's mathematical model named C2SDM (*i.e.*, the main model systematically formulating C2SD problem) along with auxiliary models and constructing network, C2SD is reduced to a series of search for feasible flow in the constructed network, and consequently four iterative algorithms, with two derived respectively from other two by applying binary search, are developed to solve C2SD. It is proved that the optimum solution is found for C2SD by each of the four algorithms in a polynomial time. Owing to sufficient exploitation of the network structure of C2SD, the proposed algorithms exhibit high computational efficiency and avoid memory overflow. They are also easy to implement computationally and can be readily extended to capacitated multistage time minimization transportation problems with destination-stage demand specified. Distinct size examples are presented to showcase the application value and practical performance difference of the four algorithms compared to generic solver LINGO applied to C2SDM model, and the causes for incurring the performance difference are analyzed. Computation experiments on different size instances produced at random are conducted to further validate the practical performance of the four iterative algorithms compared with LINGO. It is validated that all the four algorithms overwhelmingly excel LINGO in solving quality and rival LINGO in computation time when applied to C2SD instances, successfully overcoming the defect of LINGO with poor optimization effect. It is indicated that the developed four algorithms are robust and efficient exact solution approach to C2SD, and they are able to serve as a powerful tool to tackle other relevant complex optimization problems.

Mathematics Subject Classification. 90-08, 65K05, 90C08.

Received May 18, 2025. Accepted April 21, 2026.

Keywords. Transportation, binary search, iterative algorithm, time minimization, network with lower and upper arc capacities.

[1] School of Mathematics and Statistics, Sichuan University of Science and Engineering, Zigong 643000, P.R. China.

[2] Institute of Computational Mathematics and Scientific/Engineering Computing, Chinese Academy of Sciences, Beijing 100080, P.R. China.

[3] Department of Mathematics, Nanchang University, Nanchang 330031, P.R. China.

[*]Corresponding author: fanrongxie@sina.com

1. Introduction

Economy development and social life are inseparable from transportation. With the globalization development of economy and the increasing advancement of human life level, more and more companies and enterprises attach importance to *transportation problem* (TP). As a kind of important optimization problem prevalent in practice [2], TP can be partitioned into two classes as per goal, where the first class is *cost minimization transportation problem* (CMTP) and its variants, and the second class is *time minimization transportation problem* (TMTP) and its variants. CMTP is to find the cheapest plan to ship homogeneous commodities from multiple origins to several destinations, and many scholars such as Balaji *et al.* [4], Gurupada *et al.* [15], Xie and Li [42], Midya *et al.* [26] and others, have deeply researched CMTP and its variants with important achievements acquired [11, 45]. TMTP is to find the feasible transportation plan with *duration* (the slowest route shipping time) minimized. As is well known, TMTP usually appears in many practical application scenarios, including the shipment of perishable commodities such as vegetables and fruits, and the military operations for emergent shipment of military troop to battle field, where the losses of shipment delay vastly exceed the advantages of any lower cost shipment. For convenience of practical application, many scholars including Ahuja [1], Bansal and Puri [5], Garfinkel *et al.* [12], Hammer [16], Issermann [18], and others [3, 6, 12], have deeply researched TMTP and its variants with precious achievements acquired from distinct perspectives.

Because of resource finiteness, *capacitated time minimization transportation problem* (CTMTP) with each route shipping capacity considered attracts the great interest of decision makers. As is well known [5], CTMTP is a generalization of TMTP. It involves concave function minimization over the transportation polytope and therefore belongs to the class of concave minimization problems. As per the concavity of objective function, the optimum solution is findable for CTMTP in its basic feasible solution set. Besides, many algorithms [31] available for solving TMTP are extendable to CTMTP, and the optimal feasible solution is findable for CTMTP by solving associated *capacitated cost minimization transportation problem* (CCMTP) belonging to *capacitated transportation problem* (CTP) class. Moreover, CTP class with precious achievements acquired has been studied from different perspectives by many scholars such as Hassain *et al.* [17], Rachev and Olkin [28], Dahiya and Verma [7], and Xie and Jia [40]. As is notable, majority researches on CTP assume transportation occurrence in one stage.

As is notable in Kaur *et al.* [21] and Jain *et al.* [20], in some scenarios [13], including the shipment of perishable commodities to natural calamity hit areas as well as the delivery of military equipments to distinct war fields, single stage occurrence assumption for TP is naturally unreasonable because the products more than demands are unacceptable to the destinations due to their storage constraints. Clearly, it is best to transport products in multiple stages in these scenarios [8, 9, 14]. Accordingly, multistage TP has been studied deeply by many scholars [30, 39], and important research achievements have been acquired from different perspectives as mentioned below [37].

On one hand, many researchers [10, 20, 23, 32, 34, 36, 45, 47] are devoted to investigating *multilevel hierarchical time minimization transportation problem* (MHTMTP) (or *capacitated multilevel hierarchical time minimization transportation problem* (CMHTMTP)) stated below, with important achievements acquired.

Homogeneous products are to be shipped in multiple stages from multiple origins with available supply given to multiple destinations with specified demand to be met. A division of the route set is made as multiple nonempty disjoint subsets respectively used in the transportation in corresponding stage, the transportation in identical stage is parallel conducted, and the transportation in current stage starts only after the transportation in previous stage is completed. In the above context, MHTMTP (or CMHTMTP) aims at finding a multilevel hierarchical transportation plan with *duration* (measured by the sum of all the stage shipping times) minimized to satisfy the destination demand subject to restrictions on origin supply (or origin supply and route shipping capacity).

It is clear that CMHTMTP is a generalization of MHTMTP, meanwhile MHTMTP is treatable as the CMHTMTP with each route capacity infinite or sufficiently big, but CMHTMTP is more intractable than MHTMTP. Also, the MHTMTP (or CMHTMTP) with no greater than three levels is well studied with robust and efficient solution technique acquired. However, due to the intractability, the MHTMTP (or CMHTMTP) with more than three levels has never been researched in literature so far.

As is well known, Sonia and Puri [36] originally developed a polynomial time iterative algorithm for efficient solution to *two-level hierarchical time minimization transportation problem* (2HTMTP, *i.e.*, the MHTMTP with two levels) by generating two finite sequences of the pair of Level-1 and Level-2 shipping time. Afterwards, Sharma *et al.* [32] developed a more efficient algorithm to solve 2HTMTP as compared to Sonia and Puri [36]'s algorithm by generating only one finite sequence of the pair of Level-1 and Level-2 shipping time to make Level-1 shipping time rigorously decrease and Level-2 shipping time monotonously increase for any pair. Both Sharma *et al.* [32] and Sonia and Puri [36] adopted the equivalent weights technique proposed in Sherali [33]. As a consequence, their solution techniques suffer from memory overflow issues in computer implementation. For this, Xie *et al.* [45] proposed an iterative algorithm with polynomial time complexity to solve 2HTMTP transformed into a series of search for feasible flow in a network with lower and upper arc capacities, and successfully overcame the defect of memory overflow with regard to Sharma *et al.* [32]'s and Sonia and Puri [36]'s algorithm for the first time. Afterwards, Kaur *et al.* [23] developed a strongly polynomial time algorithm for solving 2HTMTP, with fast descent characteristics and no memory overflow and superiority to all the extant solution methods in computation time.

Unfortunately, like Sharma *et al.* [32]'s and Sonia and Puri [36]'s algorithm, Kaur *et al.* [23]'s algorithm developed for solving 2HTMTP holds the deficiency of difficult extension to the MHTMTP (or CMHTMTP) with greater than two levels owing to not exploiting the network structure of 2HTMTP. Subsequently, Ding and Xie [10] proposed a robust and efficient solution technique for *capacitated two-level hierarchical time minimization transportation problem* (C2HTMTP, *i.e.*, a generalized 2HTMTP, or a generalization of *two-level priority based assignment problem* (2PBAP) [19, 21, 38, 46], or the CMHTMTP with two levels), with superiority to all the extant 2HTMTP solving techniques in computation time. Based on creating mathematical model along with auxiliary models and constructing network to reduce C2HTMTP to a series of search for feasible flow in the constructed network, Ding and Xie [10]'s solution technique developed for C2HTMTP includes two polynomial time iterative algorithms and their binary search iterative algorithms, where two algorithms have fast descent characteristics. Also, Singh and Singh [34] developed a solution procedure consisting of new algorithms hybridized within particle swarm optimization to solve 2HTMTP. Nevertheless, as a population-based random search optimization technique, Singh and Singh [34]'s solution approach to 2HTMTP is vastly inferior to Ding and Xie [10]'s solution technique for C2HTMTP in computation time due to no exploitation of the network structure of 2HTMTP.

Moreover, Jain *et al.* [20] originally developed a polynomial time iterative algorithm for solving *three-level hierarchical time minimization transportation problem* (3HTMTP, *i.e.*, the MHTMTP with three levels) reduced to resolving a collection of the restricted version of standard TMTP. Recently, Zhang and Xie [47] developed a technique to solve *capacitated three-level hierarchical time minimization transportation problem* (C3HTMTP, *i.e.*, a generalized 3HTMTP, or the CMHTMTP with three levels) more efficiently than Jain *et al.* [20]'s solution approach to 3HTMTP by creating mathematical model along with auxiliary models including commonly used auxiliary model and constructing network to reduce C3HTMTP to a series of search for feasible flow in the constructed network. Different from Ding and Xie [10]'s solution technique for C2HTMTP, Zhang and Xie [47]'s solution technique for C3HTMTP is constituted of two iterative algorithms with polynomial time complexity and their binary search iterative algorithms, two as heuristics with fast descent characteristics and high accuracy, another two as exact optimal algorithms. As compared to Ding and Xie [10]'s solution technique for C2HTMTP, due to more fully exploiting the network structure of C3HTMTP, Zhang and Xie [47]'s solution technique for C3HTMTP is more easily extendable to the CMHTMTP with more than three levels.

As is notable, extant research on MHTMTP or CMHTMTP assumes that any route can be used only for unitary stage shipment. However, there are some scenarios where any route can be used for multistage shipment. Considering the scenarios with multistage usability for any route, *two-stage time minimization transportation problem* (2TMTP) was originally investigated by Sonia and Malhotra [35] developing an iterative algorithm with polynomial time complexity for solving 2TMTP. Afterwards, Sonia and Malhotra [35]'s algorithm inability in guaranty for finding the optimum solution to 2TMTP was found by Sharma *et al.* [31], and a polynomial time iterative algorithm was consequently proposed to solve *capacitated two-stage time minimization transportation problem* (C2TMTP, *i.e.*, a generalized 2TMTP), with guaranty for finding the optimum solution to 2TMTP successfully.

Because of adopting the equivalent weights technique proposed in Sherali [33], both Sharma *et al.* [31]'s solution technique developed for C2TMTP and Sonia and Malhotra [35]'s solution technique developed for 2TMTP hold the defect of memory overflow in computer implementation, especially for large size instances [25, 45]. As such, Xie and Li [43] developed a more robust and efficient C2TMTP solving technique, successfully overcoming the defect of memory overflow of Sharma *et al.* [31]'s and Sonia and Malhotra [35]'s algorithm for the first time. Based on creating mathematical model along with auxiliary models and constructing network to reduce C2TMTP to a series of search for feasible flow in the constructed network, Xie and Li [43]'s technique developed for solving C2TMTP includes two polynomial time iterative algorithms, one with fast descent characteristics, and both with no application of binary search to enhancing the computational efficiency.

The extant researches [31, 43] on C2TMTP ignore the second stage practical demand satisfaction at each destination, and the ignoring is not only unviable in reality because of violating the law of supply for demand, but also results in the difficulty of their extension to the scenarios with more than two stages [22, 24]. Because of destination-stage demand timely satisfaction importance and resource finiteness and realistic existence of multistage usable route, it is much needed to deeply investigate *capacitated multistage time minimization transportation problem with destination-stage demand specified* (CMTMTPSD) as stated below.

Homogeneous products are to be transported in multiple stages from multiple origins with available supply given to multiple destinations with each stage demand specified to be timely met. The identical stage shipment is parallel conducted, and current stage shipment commences after previous stage shipment is completed. Each route from origin to destination can be used for multiple stage shipment, but the total shipment volume in all the stages for a route does not exceed the shipping capacity of the route. In the above context, CMTMTPSD aims at finding a multistage shipment plan with minimal *duration* (measured by the sum of all the stage shipping times) to satisfy each stage demand at all the destinations.

Clearly, CMTMTPSD is intimately related to but intrinsically distinct from C2TMTP. The main difference lies in that C2TMTP only considers the first stage practical demand satisfaction at each destination, but ignores the second stage practical demand satisfaction at each destination, and hence it is difficult for C2TMTP to be extended to the scenarios with stage exceeding two, while CMTMTPSD considers each stage practical demand satisfaction at each destination, and covers the scenarios with stage more than two. Also, from mathematical modeling perspective, CMTMTPSD is completely different from CMHTMTP, although both are dedicated to deal with the time minimization of capacitated multistage transportation problem. As compared to CMHTMTP, CMTMTPSD is an interesting but more challenging optimization problem in logistics management. To the best of our knowledge, there is no research report on CMTMTPSD in literature so far due to its super intractability. This paper is to explore the simplest but crucial case of CMTMTPSD, which is *capacitated two-stage time minimization transportation problem with destination-stage demand specified* (C2SD, *i.e.*, the CMTMTPSD with two stages). The main contribution of this paper is as follows.

By creating C2SD's mathematical model named C2SDM (*i.e.*, the main model systematically formulating C2SD problem) along with auxiliary models and constructing network, C2SD is reduced to a series of search for feasible flow in the constructed network [10, 43, 46, 47]. Based on feasible flow algorithm FFn-A in Xie and Li [43], two iterative algorithms C2SD-A1 and C2SD-A2, as well as two binary search iterative algorithms C2SD-A1bs and C2SD-A2bs derived respectively from application of binary search to C2SD-A1 and C2SD-A2, are developed [10, 46, 47] to solve C2SD. It is theoretically proved that our four algorithms (C2SD-A1, C2SD-A1bs, C2SD-A2, C2SD-A2bs) can find the optimum solution to C2SD in a polynomial time. Because of fully utilizing the network structure of C2SD, our four algorithms hold the merits of high computational efficiency, no memory overflow, easy computer implementation, and easy extension to the CMTMTPSD with more than two stages.

Different size examples are presented to demonstrate the application and performance difference of our four algorithms in comparison with generic solver LINGO applicable to solving main model C2SDM, and the causes for resulting in the performance difference are analyzed. Computation experiments on distinct size instances produced at random are performed to further verify the practical performance for our four algorithms to find the optimum solution to C2SD efficiently, in comparison with generic solver LINGO in optimization effect and computation time. It is revealed that in computation time, C2SD-A2 significantly outperforms C2SD-A2bs, C2SD-A2bs is much better than C2SD-A1bs, and C2SD-A1bs has overwhelming superiority to C2SD-A1. Generally speaking, in computation time, the bigger the size of instance solved, the bigger the advantage of C2SD-A2 over C2SD-A2bs, C2SD-A2bs over C2SD-A1bs, C2SD-A1bs over C2SD-A1, and C2SD-A2 over C2SD-A1. In particular, when applied to solving the instances of C2SD, all of our four algorithms overwhelmingly excel LINGO in solving quality and rival LINGO in computation time, successfully overcoming the poor optimization effect deficiency of LINGO finding only a feasible solution far from the optimal for C2SD.

Consequently, a novel high efficiency exact solution technique, with easy extension to more complex optimization problems in logistics management, is developed in this paper for challenging and crucial C2SD unstudied in literature but emerging in reality.

The remaining parts of this paper are organized as follows. The formulation and solution approach are presented for C2SD in Section 2. Distinct size examples are presented in Section 3 to demonstrate the application and performance difference of our developed four algorithms (C2SD-A1, C2SD-A1bs, C2SD-A2, C2SD-A2bs) compared to LINGO. The causes for resulting in the performance difference of our four algorithms in comparison with LINGO are analyzed in Section 4. Computation experiments on distinct size instances produced at random are conducted in Section 5 to further validate the practical performance of our four algorithms as compared with LINGO. Concluding remarks are made in Section 6.

2. Formulation and solution approach

2.1. Usable notations

For better comprehension of solution approach to C2SD, usable notations are presented below (see Appendix A).

2.2. Mathematical model and solving analysis

C2SD can be stated as follows. Homogeneous products are to be transported in two stages from $\tilde{m}$ origins comprising origin i with available supply as $\hat{s}_i$ $(i \in I = \{1, 2, \cdots, \tilde{m}\})$ to $\tilde{n}$ destinations comprising destination j with first and second stage demand respectively as $\hat{d}_j^{(1)}$ and $\hat{d}_j^{(2)}$ $(j \in J = \{1, 2, \cdots, \tilde{n}\})$. The identical stage shipment is parallel conducted, and the second stage shipment starts after the first stage shipment is completed. For the route usable for both stage shipment from origin $i \in I$ to destination $j \in J$, *i.e.*, route $(i, j) \in I \times J$, the shipping time and capacity are $\hat{t}_{ij}$ and $\hat{u}_{ij}$ respectively. The total demand at all the destinations does not exceed the total available supply at all the origins, *i.e.*, $\sum_{j\in J}(\hat{d}_j^{(1)} + \hat{d}_j^{(2)}) \leq \sum_{i\in I} \hat{s}_i$. For any route $(i, j) \in I \times J$, $\hat{t}_{ij} > 0$ and $\hat{u}_{ij} > 0$. For any origin $i \in I$, $\hat{s}_i > 0$. For any destination $j \in J$, $\hat{d}_j^{(1)} > 0$ and $\hat{d}_j^{(2)} > 0$. On the above context, the goal is to find a feasible shipment plan with minimal *duration* (*i.e.*, the sum of the first and second stage shipping time).

Let $x_{ij}^{(1)}$ and $x_{ij}^{(2)}$ respectively denote the first and second stage shipment volume of route $(i, j) \in I \times J$. Let z denote the duration. Then C2SD is formulated as mathematical model named C2SDM below.

$$
\text{C2SDM}:\begin{cases}
\min z=\max\left\{\hat{t}_{ij}\mid (i,j)\in I\times J, x_{ij}^{(1)}>0\right\}+\max\left\{\hat{t}_{ij}\mid (i,j)\in I\times J, x_{ij}^{(2)}>0\right\}\\
\text{s.t.}\\
\quad\sum_{j\in J}\left(x_{ij}^{(1)}+x_{ij}^{(2)}\right)\le \hat{s}_i \quad \forall i\in I & (1)\\
\quad x_{ij}^{(1)}+x_{ij}^{(2)}\le \hat{u}_{ij} \quad \forall (i,j)\in I\times J & (2)\\
\quad\sum_{i\in I}x_{ij}^{(1)}=\hat{d}_j^{(1)} \quad \forall j\in J & (3)\\
\quad\sum_{i\in I}x_{ij}^{(2)}=\hat{d}_j^{(2)} \quad \forall j\in J & (4)\\
\quad x_{ij}^{(1)},x_{ij}^{(2)}\ge 0 \quad \forall (i,j)\in I\times J. & (5)
\end{cases}
$$

Clearly, our studied C2SD is significantly different from C2TMTP investigated in Xie and Li [43] and Sharma *et al.* [31]. As compared to C2TMTP, C2SD is more targeted in demand satisfaction, and hence more consistent with the law of supply for demand. For C2SD, both the first and second stage practical demand at each destination are specified, and therefore each stage shipment aims exactly at practical demand satisfaction at each destination. Whereas, for C2TMTP, the second stage practical demand at each destination is not specified, and consequently the second stage shipment in the found optimum solution is of blindness in demand satisfaction. Also, as compared to C2TMTP, C2SD is more easily extended to the scenarios with more than two stages. It follows that C2SD is more valuable in theory and practice as compared to C2TMTP, and extant C2TMTP solving technique [31, 43] is inapplicable to resolving C2SD. It is known that generic solver LINGO is usable for solving main model C2SDM, but it is findable (for validation, see Sect. 3) that the optimization effect is very poor for C2SDM model based LINGO computation program finding only a feasible solution far from the optimal for C2SD. Therefore, it is much needful to develop a robust and efficient solution technique for C2SD.

Since C2SD is a problem extended from C2TMTP, a natural idea is the extension of extant C2TMTP solving algorithms to efficiently solving C2SD. However, it is notable that as compared to C2TMTP, C2SD has an additional constraint of satisfying the second stage practical demand at each destination, and the additional constraint with super intractability introduces tremendous challenges for the extension of extant C2TMTP solving algorithms to resolving C2SD in high efficiency.

Also, it is known that in the existing C2TMTP solving techniques [31, 43], the best in scalability is Xie and Li [43]'s approach with core idea [10, 46, 47] as reducing the studied problem to a series of search for feasible flow in the constructed network by the creation of main model (*i.e.*, the mathematical model systematically formulating studied problem) along with auxiliary models and the construction of network. It involves a series of innovative work such as creation of suitable main and auxiliary models, construction of network reflecting main and auxiliary models, and transformation of main and auxiliary models into network flow model. Notably, based on the above core idea, high efficiency solution technique for 2PBAP, C2HTMTP and C3HTMTP was successfully developed by Xie *et al.* [46], Ding and Xie [10] and Zhang and Xie [47], respectively. Because the additional constraints can be readily reflected in the constructed network by elaborately constructing network [47], it is very promising to extend Xie and Li [43]'s solution method developed for C2TMTP to efficiently resolve C2SD by fully exploiting the network structure of studied problem. In contrast, owing to not fully exploiting the network structure of studied problem, the other algorithms from the previous literature [31] cannot be easily adapted to C2SD.

By virtue of the above reasons [48], our goal is to develop a robust and efficient C2SD solving technique by modifying the previous work in Xie and Li [43] and Xie *et al.* [46] and Ding and Xie [10] and Zhang and Xie [47] to solving the extended problem C2SD.

For attainment of our goal, a rigorously descent sequence $\alpha(1)$, $\alpha(2)$, $\alpha(3)$, $\cdots$, $\alpha(\omega)$, with ω as the number of distinct values in all the route shipping times, is first produced by sorting all the elements in $\{\hat{t}_{ij}|(i,j)\in I\times J\}$ and treating equal ones as the identical.

Next, four auxiliary models called respectively as AMSD(1), AMSD(2), AMSD(3) and AMSD(4) are created below for main model C2SDM. Auxiliary models AMSD(1) and AMSD(2) do not have controllable parameter.

Auxiliary model AMSD(3) has only one controllable parameter Tu indicating an upper bound on the first stage shipping times of some feasible solutions to main model C2SDM. Auxiliary model AMSD(4) has controllable parameters z_1 and z_2 respectively denoting the first and second stage shipping time of a feasible solution to main model C2SDM, as well as controllable parameter $p \in \Omega = \{1, 2, \cdots, \omega\}$.

$$\text{AMSD(1)} : \begin{cases} \min z^{(1)} = \max\{\hat{t}_{ij} | (i,j) \in I \times J, x_{ij}^{(1)} > 0\} \\ \text{s.t. constraints (1) to (5) in C2SDM.} \end{cases}$$

$$\text{AMSD(2)} : \begin{cases} \min z^{(2)} = \max\{\hat{t}_{ij} | (i,j) \in I \times J, x_{ij}^{(2)} > 0\} \\ \text{s.t. constraints (1) to (5) in C2SDM.} \end{cases}$$

$$\text{AMSD(3)} : \begin{cases} \min\ z^{(2)} = \max\left\{\hat{t}_{ij} | (i,j) \in I \times J, x_{ij}^{(2)} > 0\right\} \\ \text{s.t.} \begin{cases} \text{constraints (1) to (5) in C2SDM;} \\ \forall (i,j) \in I \times J, \text{if } \hat{t}_{ij} \geq Tu, \text{then } x_{ij}^{(1)} = 0. \end{cases} \end{cases}$$

$$\text{AMSD(4)} : \begin{cases} \min\ z^{(2)} = \max\left\{\hat{t}_{ij} | (i,j) \in I \times J, x_{ij}^{(2)} > 0\right\} \\ \text{s.t.} \begin{cases} \text{constraints (1) to (5) in C2SDM;} \\ \forall (i,j) \in I \times J, \text{if } \hat{t}_{ij} > \alpha(p), \text{then } x_{ij}^{(1)} = 0; \\ \forall (i,j) \in I \times J, \text{if } \hat{t}_{ij} \geq z_1 + z_2 - \alpha(p), \text{then } x_{ij}^{(2)} = 0. \end{cases} \end{cases}$$

As per the construction of four auxiliary models $\{\text{AMSD}(h) \,|\, h = 1, 2, 3, 4\}$, the following Proposition 1 is easily derived.

Proposition 1. *AMSD(1) and AMSD(2) and C2SDM have the same feasible solution set. The feasible solution set of AMSD(3) (or AMSD(4)) is a subset of the feasible solution set of C2SDM. Also, the AMSD(3) with controllable parameter Tu as positive infinity ($+\infty$) or sufficiently big positive number is equivalent to AMSD(2).*

Our intention is application of four auxiliary models $\{\text{AMSD}(h) \,|\, h = 1, 2, 3, 4\}$ to an efficient iteration process of search for the optimum solution to main model C2SDM. For this, an iterative algorithm based on feasible flow and its binary search iterative algorithm are developed below for each auxiliary model by virtue of Theorem 1 in Xie and Li [43].

2.3. Efficient solution approach to auxiliary model

For convenience of efficient solution to four auxiliary models $\{\text{AMSD}(h) \,|\, h = 1, 2, 3, 4\}$, two commonly used auxiliary models $\{\text{CUAM}(g) \,|\, g = 1, 2\}$ are created below.

$$\text{CUAM}(g) : \begin{cases} \min z^{(g)} = \max\left\{\hat{t}_{ij} \mid (i,j) \in I \times J, x_{ij}^{(g)} > 0\right\} \\ \text{s.t.} \\ \quad \sum_{j \in J} (x_{ij}^{(1)} + x_{ij}^{(2)}) \leq \hat{s}_i & \forall i \in I & (1) \\ \quad x_{ij}^{(1)} + x_{ij}^{(2)} \leq \hat{u}_{ij} & \forall (i,j) \in I \times J & (2) \\ \quad \sum_{i \in I} x_{ij}^{(1)} = \hat{d}_j^{(1)} & \forall j \in J & (3) \\ \quad \sum_{i \in I} x_{ij}^{(2)} = \hat{d}_j^{(2)} & \forall j \in J & (4) \\ \quad 0 \leq x_{ij}^{(1)} \leq u_{ij}^{(1)} & \forall (i,j) \in I \times J & (5) \\ \quad 0 \leq x_{ij}^{(2)} \leq u_{ij}^{(2)} & \forall (i,j) \in I \times J. & (6) \end{cases}$$

In CUAM(g) with g as 1 or 2, the controllable parameters $u_{ij}^{(1)}$ and $u_{ij}^{(2)}$ comply with the following stipulations: $u_{ij}^{(1)}$ (or $u_{ij}^{(2)}$) is taken value as $\hat{u}_{ij}$ (present in C2SDM) or 0, while $u_{ij}^{(1)} = 0$ (or $u_{ij}^{(2)} = 0$) indicates that route

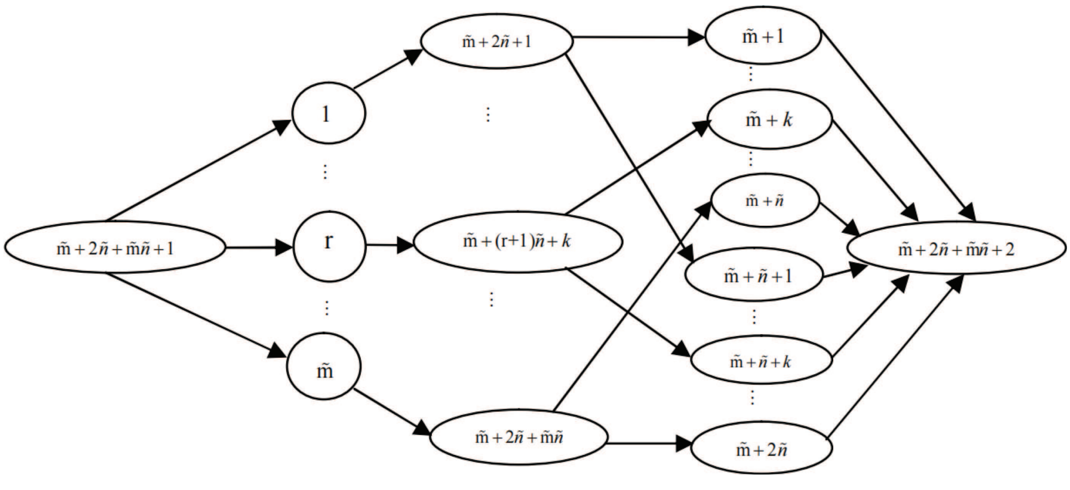

FIGURE 1. Network with lower and upper arc capacities, G.

(i,j) is unusable or disconnected in the first (or second) stage. The known parameters $(I, J, \hat{s}_i, \hat{d}_j^{(1)}, \hat{d}_j^{(2)}, \hat{t}_{ij}, \hat{u}_{ij})$ in CUAM(g) are the same as that in C2SDM. Also, the variables $x_{ij}^{(1)}$ (or $x_{ij}^{(2)}$) and $z^{(1)}$ (or $z^{(2)}$) in CUAM(1) (or CUAM(2)) denote the first (or second) stage shipment volume of route (i,j) and the first (or second) stage duration respectively.

Four auxiliary models $\{\mathrm{AMSD}(h)\,|\,h = 1,2,3,4\}$ are actually solvable by treatment as special cases of CUAM(g) with g as 1 or 2. By virtue of a network constructed in Definition 1 below and Theorem 1 in Xie and Li [43], a polynomial time iterative algorithm based on feasible flow is first developed to efficiently solve CUAM(g) with g as 1 or 2.

Definition 1. A network (see Fig. 1) $G = (V, s, t; A; \underline{C}, \overline{C})$ with lower and upper arc capacities is constructed for CUAM(g) with g as 1 or 2. For network G, node $s = \tilde{m} + 2\tilde{n} + \tilde{m}\tilde{n} + 1$ is source, node $t = \tilde{m} + 2\tilde{n} + \tilde{m}\tilde{n} + 2$ is sink, node set V and arc set A are constituted as given in Table 1. The lower and upper capacity are zero and $\hat{s}_r$ respectively for arc (s,r) in A ($r \in I$), both $\hat{d}_k^{(1)}$ for arc $(\hat{m} + k, t)$ in A ($k \in J$), and both $\hat{d}_k^{(2)}$ for arc $(\hat{m} + \tilde{n} + k, t)$ in A ($k \in J$). For any arc $(r, \tilde{m} + (r+1)\tilde{n} + k)$ or $(\tilde{m} + (r+1)\tilde{n} + k, \tilde{m} + k)$ or $(\tilde{m} + (r+1)\tilde{n} + k, \tilde{m} + \tilde{n} + k)$ in A ($r \in I, k \in J$), the lower capacity is zero, the upper capacity is $\hat{u}_{rk}$ or $u_{rk}^{(1)}$ or $u_{rk}^{(2)}$ respectively. $n = \tilde{m} + 2\tilde{n} + \tilde{m}\tilde{n} + 2$ is the number of nodes in G. As per Definition 1 in Xie *et al.* [45], the lower and upper arc capacity matrices $\underline{C} = (\underline{C}_{ij})_{n\times n}$ and $\overline{C} = (\overline{C}_{ij})_{n\times n}$ are determined by the stipulations given in Table 2.

Clearly, if there is such a feasible flow from source s to sink t in network G constructed in Definition 1 that for any $r \in I$ and $k \in J$, the flow on arc $(\tilde{m} + (r+1)\tilde{n} + k, \tilde{m} + k)$ and $(\tilde{m} + (r+1)\tilde{n} + k, \tilde{m} + \tilde{n} + k)$ is set as $x_{rk}^{(1)}$ and $x_{rk}^{(2)}$ respectively, then a feasible solution with first stage shipment plan as $x^{(1)} = \{x_{ij}^{(1)} \mid (i,j) \in I \times J\}$ and second stage shipment plan as $x^{(2)} = \{x_{ij}^{(2)} \mid (i,j) \in I \times J\}$ is obtained for CUAM(g). As a result, Proposition 2 below is derived.

Proposition 2. *The feasible solution structure of CUAM(g) with g as 1 or 2 is reflected by network G constructed in Definition 1. In another words, there exists one to one correspondence between the feasible solution to CUAM(g) with g as 1 or 2 and the feasible flow in network G constructed in Definition 1. Concretely, the feasible solution existence for CUAM(g) with g as 1 or 2 is equivalent to the feasible flow existence for network G constructed in Definition 1, and if existent, a feasible solution to CUAM(g) with g as 1 or 2 corresponds one to one to a feasible flow in network G constructed in Definition 1.*

TABLE 1. The constitution of node set V and arc set A for network $G = (V, s, t; A; \underline{C}, \overline{C})$.

All division subsets constituting node set V	All division subsets constituting arc set A
I	$\{(s, r) \mid r \in I\}$
$\{\tilde{m} + k \mid k \in J\}$	$\{(r, \tilde{m} + (r+1)\tilde{n} + k) \mid r \in I, k \in J\}$
$\{\tilde{m} + \tilde{n} + k \mid k \in J\}$	$\{(\tilde{m} + (r+1)\tilde{n} + k, \tilde{m} + k) \mid r \in I, k \in J\}$
$\{\tilde{m} + (r+1)\tilde{n} + k \mid r \in I, k \in J\}$	$\{(\tilde{m} + (r+1)\tilde{n} + k, \tilde{m} + \tilde{n} + k) \mid r \in I, k \in J\}$
$\{\tilde{m} + 2\tilde{n} + \tilde{m}\tilde{n} + 1\}$	$\{(\tilde{m} + k, t) \mid k \in J\}$
$\{\tilde{m} + 2\tilde{n} + \tilde{m}\tilde{n} + 2\}$	$\{(\tilde{m} + \tilde{n} + k, t) \mid k \in J\}$

TABLE 2. The stipulations for determination of matrices $\underline{C} = (\underline{C}_{ij})_{n\times n}$ and $\overline{C} = (\overline{C}_{ij})_{n\times n}$.

Manipulation for any $i, j \in \{1, 2, \cdots, n\}$
When $i = s$ and $j = r$ $(r \in I)$, set $\underline{C}_{ij} = 0$, $\overline{C}_{ij} = \hat{s}_r$.
When $i = r$ and $j = \tilde{m} + (r+1)\tilde{n} + k$ $(r \in I, k \in J)$, set $\underline{C}_{ij} = 0$, $\overline{C}_{ij} = \hat{u}_{rk}$.
When $i = \tilde{m} + (r+1)\tilde{n} + k$ and $j = \tilde{m} + k$ $(r \in I, k \in J)$, set $\underline{C}_{ij} = 0$, $\overline{C}_{ij} = u^{(1)}_{rk}$.
When $i = \tilde{m} + (r+1)\tilde{n} + k$ and $j = \tilde{m} + \tilde{n} + k$ $(r \in I, k \in J)$, set $\underline{C}_{ij} = 0$, $\overline{C}_{ij} = u^{(2)}_{rk}$.
When $i = \tilde{m} + k$ $(k \in J)$ and $j = t$, set $\underline{C}_{ij} = \hat{d}^{(1)}_k$, $\overline{C}_{ij} = \hat{d}^{(1)}_k$.
When $i = \tilde{m} + \tilde{n} + k$ $(k \in J)$ and $j = t$, set $\underline{C}_{ij} = \hat{d}^{(2)}_k$, $\overline{C}_{ij} = \hat{d}^{(2)}_k$.
In the other cases, if $i = j$, then set $\underline{C}_{ij} = 0$, $\overline{C}_{ij} = +\infty$, else set $\underline{C}_{ij} = 0$, $\overline{C}_{ij} = 0$.

From Theorem 1 in Xie and Li [43] and Proposition 2 herein, it is derivable that for CUAM(g) with g as 1 or 2, the feasible solution existence is judged and a feasible solution is found if existent, in a polynomial time by calling Xie and Li [43]'s algorithm FFn-A.

For convenience, with respect to route $(r, k) \in I \times J$, arc $(\widetilde{m}+(r+1)\widetilde{n}+k, \widetilde{m}+k)$ (or $(\widetilde{m}+(r+1)\widetilde{n}+k, \widetilde{m}+\widetilde{n}+k)$) in network G constructed in Definition 1 is referred to as the first (or second) stage flow controllable arc.

A novel idea for solving CUAM(g) with g as 1 or 2 in high efficiency is given below.

Gradually make an attempt to initially disconnect the first (or second) stage flow controllable arcs of all the routes with shipping time exceeding a specified upper bound $z^{(b)}$ in network G constructed in Definition 1 to get an updated network G, and then apply Xie and Li [43]'s feasible flow algorithm FFn-A to the recently gotten network G, until Xie and Li [43]'s algorithm FFn-A finds feasible flow nonexistence in the recently gotten network G or the check on all the necessary values of specified upper bound $z^{(b)}$ is completed. The specified upper bound $z^{(b)}$ is initialized as positive infinity $(+\infty)$ and next set as the objective function value of a possibly found feasible solution to CUAM(g) by Xie and Li [43]'s algorithm FFn-A, then decreases along the rigorously descent sequence produced by sorting all the route shipping times and treating equal ones as the identical. As a consequence of the attempt, the possibly found minimal specified upper bound $z^{(b)}$ corresponding to a feasible solution to CUAM(g) is surely the optimum objective function value of CUAM(g), with corresponding feasible solution surely as the optimum solution to CUAM(g).

By virtue of the above idea, Lemma 1 below is acquired.

Lemma 1. *Given g as 1 or 2, iterative algorithm called CUAM(g)-A below judges the feasible solution existence for CUAM(g) with optimum solution to be found if existent, in a polynomial time with $O(\omega T_f)$ as complexity.*

Herein, T_f is the computation time consumed to perform Xie and Li [43]'s algorithm FFn-A once for network G constructed in Definition 1, and ω is the number of distinct values in all the route shipping times.

CUAM(g)-A

Step 1: Produce a rigorously descent sequence $\alpha(1), \alpha(2), \ldots, \alpha(\omega)$ by sorting all the elements in $\{\hat{t}_{ij} \mid (i,j) \in I \times J\}$ and treating equal ones as the identical. Construct network $G = (V, s, t; A; \underline{C}, \overline{C})$ by Definition 1. Set $C_{ij} = \overline{C}_{ij}$ for any $i, j \in \{1, 2, \ldots, n\}$. Set $q = 1$, $Fd = 0$, $z^{(b)} = +\infty$.

Step 2: For any route $(r,k) \in I \times J$, set $i = \tilde{m} + (r+1)\tilde{n} + k$, set $j = \tilde{m} + k$ in case of $g = 1$ or set $j = \tilde{m} + \tilde{n} + k$ in case of g= 2, if $\hat{t}_{rk} > \alpha(q)$, then set $\overline{C}_{ij} = 0$, else set $\overline{C}_{ij} = C_{ij}$.

Step 3: For current network $G = (V, s, t; A; \underline{C}, \overline{C})$, call Xie and Li [43]'s algorithm FFn-A. If a feasible flow with $F = (F_{ij})_{n \times n}$ as flow matrix is found for current network G by Xie and Li [43]'s algorithm FFn-A, then successively manipulate as below: "for any route $(r,k) \in I \times J$, set $i = \tilde{m} + (r+1)\tilde{n} + k$, $j = \tilde{m} + k$, $x_{rk}^{(1)} = F_{ij}$; for any route $(r,k) \in I \times J$, set $i = \tilde{m} + (r+1)\tilde{n} + k$, $j = \tilde{m} + \tilde{n} + k$, $x_{rk}^{(2)} = F_{ij}$; find $z^{(g)} = \max\{\hat{t}_{ij} \mid (i,j) \in I \times J, x_{ij}^{(g)} > 0\}$; set $z^{(b)} = z^{(g)}$, and set $x_{ij}^{(b1)} = x_{ij}^{(1)}$ and $x_{ij}^{(b2)} = x_{ij}^{(2)}$ for any $(i,j) \in I \times J$; find $q = \max\{q \mid \alpha(q) \geq z^{(b)}; q = 1, 2, \ldots, \omega\}$; set $q = q + 1$; set $Fd = 1$; go to Step 4". Else, go to Step 5.

Step 4: If $q > \omega$, then go to Step 5. Otherwise, go to Step 2.

Step 5: If $Fd = 1$, then set $x_{ij}^{(1)} = x_{ij}^{(b1)}$ and $x_{ij}^{(2)} = x_{ij}^{(b2)}$ for any $(i,j) \in I \times J$, set $z^{(g)} = z^{(b)}$, output the optimum objective function value as $z^{(g)}$ and optimum solution with first stage shipment plan as $x^{(1)} = \{x_{ij}^{(1)} \mid (i,j) \in I \times J\}$ and second stage shipment plan as $x^{(2)} = \{x_{ij}^{(2)} \mid (i,j) \in I \times J\}$ for CUAM(g), and output $Fd = 1$ to indicate that the optimum objective function value and optimum solution have been found for CUAM(g). Otherwise, output $Fd = 0$ to indicate the feasible solution nonexistence for CUAM(g).

Proof. See Appendix B. □

As is derived from Lemma 1, it follows that CUAM(g) with g as 1 or 2 is solvable in a polynomial time. Also, the algorithm CUAM(g)-A for solving CUAM(g) is an iteration solution process of exploring along a rigorously descent sequence produced by sorting all the route shipping times and treating equal ones as the identical. By application of binary search to the iteration solution process as per the feasible solution properties of CUAM(g) and the characteristics of rigorously descent sequence, it is viable to derive a binary search iterative algorithm called CUAM(g)-bs with computational efficiency enhanced greatly in theory compared to CUAM(g)-A. As such, Lemma 2 below is acquired.

Lemma 2. *Given g as 1 or 2, binary search iterative algorithm called CUAM(g)-bs judges the feasible solution existence for CUAM(g) with optimum solution to be found if existent, in a polynomial time with $O(T_f \log \omega)$ as complexity. Herein, T_f is the computation time consumed to perform Xie and Li [43]'s algorithm FFn-A once for network G constructed in Definition 1, and ω is the number of distinct values in all the route shipping times.*

CUAM(g)-bs

Step 1: Produce a rigorously descent sequence $\alpha(1), \alpha(2), \ldots, \alpha(\omega)$ by sorting all the elements in $\{\hat{t}_{ij} \mid (i,j) \in I \times J\}$ and treating equal ones as the identical. Construct network $G = (V, s, t; A; \underline{C}, \overline{C})$ by Definition 1. Set $C_{ij} = \overline{C}_{ij}$ for any $i, j \in \{1, 2, \ldots, n\}$. Set $q = 1$, $Fd = 0$, $z^{(b)} = +\infty$.

Step 2: For any route $(r,k) \in I \times J$, set $i = \tilde{m} + (r+1)\tilde{n} + k$, set $j = \tilde{m} + k$ in case of g= 1 or set $j = \tilde{m} + \tilde{n} + k$ in case of g= 2, if $\hat{t}_{rk} > \alpha(q)$, then set $\overline{C}_{ij} = 0$, else set $\overline{C}_{ij} = C_{ij}$.

Step 3: For current network $G = (V, s, t; A; \underline{C}, \overline{C})$, call Xie and Li [43]'s algorithm FFn-A. If a feasible flow with $F = (F_{ij})_{n \times n}$ as flow matrix is found for current network G by Xie and Li [43]'s algorithm FFn-A, then successively manipulate as below: "for any route $(r,k) \in I \times J$, set $i = \tilde{m} + (r+1)\tilde{n} + k$, $j = \tilde{m} + k$, $x_{rk}^{(1)} = F_{ij}$; for any route $(r,k) \in I \times J$, set $i = \tilde{m} + (r+1)\tilde{n} + k$, $j = \tilde{m} + \tilde{n} + k$, $x_{rk}^{(2)} = F_{ij}$; find

$z^{(g)} = \max\{\hat{t}_{ij} \mid (i,j) \in I \times J, x_{ij}^{(g)} > 0\}$; set $z^{(b)} = z^{(g)}$, and set $x_{ij}^{(b1)} = x_{ij}^{(1)}$ and $x_{ij}^{(b2)} = x_{ij}^{(2)}$ for any $(i,j) \in I \times J$; find $q = \max\{q \mid \alpha(q) \geq z^{(b)}; q = 1, 2, \ldots, \omega\}$; set $Fd = 1$; set $q_1 = q$, $q_2 = \omega$, $q_{2s} = q_2$, $q = q_2$; go to Step 4". Else, go to Step 7.

Step 4: For any route $(r,k) \in I \times J$, set $i = \tilde{m} + (r+1)\tilde{n} + k$, set $j = \tilde{m} + k$ in case of $g = 1$ or set $j = \tilde{m} + \tilde{n} + k$ in case of $g = 2$, if $\hat{t}_{rk} > \alpha(q)$, then set $\overline{C}_{ij} = 0$, else set $\overline{C}_{ij} = C_{ij}$.

Step 5: For current network $G = (V, s, t; A; \underline{C}, \overline{C})$, call Xie and Li [43]'s algorithm FFn-A. If a feasible flow with $F = (F_{ij})_{n \times n}$ as flow matrix is found for current network G by Xie and Li [43]'s algorithm FFn-A, then successively manipulate as below: "for any route $(r,k) \in I \times J$, set $i = \tilde{m} + (r+1)\tilde{n} + k$, $j = \tilde{m} + k$, $x_{rk}^{(1)} = F_{ij}$; for any route $(r,k) \in I \times J$, set $i = \tilde{m} + (r+1)\tilde{n} + k$, $j = \tilde{m} + \tilde{n} + k$, $x_{rk}^{(2)} = F_{ij}$; find $z^{(g)} = \max\{\hat{t}_{ij} \mid (i,j) \in I \times J, x_{ij}^{(g)} > 0\}$; set $z^{(b)} = z^{(g)}$, and set $x_{ij}^{(b1)} = x_{ij}^{(1)}$ and $x_{ij}^{(b2)} = x_{ij}^{(2)}$ for any $(i,j) \in I \times J$; find $q = \max\{q \mid \alpha(q) \geq z^{(b)}; q = 1, 2, \ldots, \omega\}$; set $q_1 = q, q_2 = q_{2s}$; go to Step 6". Else, set $q_2 = q, q_{2s} = q$; go to Step 6.

Step 6: If $q_2 - q_1 \leq 1$, then go to Step 7. Otherwise, set $q = \lfloor (q_1 + q_2)/2 \rfloor$ as the maximal integer no more than $(q_1 + q_2)/2$, and go to Step 4.

Step 7: If $Fd = 1$, then set $x_{ij}^{(1)} = x_{ij}^{(b1)}$ and $x_{ij}^{(2)} = x_{ij}^{(b2)}$ for any $(i,j) \in I \times J$, set $z^{(g)} = z^{(b)}$, output the optimum objective function value as $z^{(g)}$ and optimum solution with first stage shipment plan as $x^{(1)} = \{x_{ij}^{(1)} \mid (i,j) \in I \times J\}$ and second stage shipment plan as $x^{(2)} = \{x_{ij}^{(2)} \mid (i,j) \in I \times J\}$ for CUAM(g), and output $Fd = 1$ to indicate that the optimum objective function value and optimum solution have been found for CUAM(g). Otherwise, output $Fd = 0$ to indicate the feasible solution nonexistence for CUAM(g).

Proof. See Appendix B. □

Lemma 3. *For any $h \in \{1, 2, 3, 4\}$, in a polynomial time with given complexity in Table 3 (where the parameters $\tilde{m}, \tilde{n}, \omega$, $d_j^{(1)}$ and $d_j^{(2)}$ are defined in Sect. 2.1), iterative algorithm (or binary search iterative algorithm) called AMSD(h)-A (or AMSD(h)-bs) below judges the feasible solution existence for AMSD(h) with optimum solution to be found if existent.*

AMSD(h)-A (or AMSD(h)-bs)

Step 1: By the stipulation given in Table 3, construct the first and second stage route shipping capacity controllable matrices $U^{(1)} = (u_{ij}^{(1)})_{\tilde{m} \times \tilde{n}}$ and $U^{(2)} = (u_{ij}^{(2)})_{\tilde{m} \times \tilde{n}}$ of the auxiliary model CUAM(g) equivalent to AMSD(h), where g is function given by Table 3 for h.

Step 2: Solve current auxiliary model CUAM(g) by CUAM(g)-A (or CUAM(g)-bs). If the optimum solution with first stage shipment plan as $x^{(1)} = \{x_{ij}^{(1)} \mid (i,j) \in I \times J\}$ and second stage shipment plan as $x^{(2)} = \{x_{ij}^{(2)} \mid (i,j) \in I \times J\}$ as well as optimum objective function value as z are found for auxiliary model CUAM(g) by CUAM(g)-A (or CUAM(g)-bs), then output AMSD(h)'s optimum solution with $x^{(1)} = \{x_{ij}^{(1)} \mid (i,j) \in I \times J\}$ and $x^{(2)} = \{x_{ij}^{(2)} \mid (i,j) \in I \times J\}$ respectively as first and second stage shipment plan as well as optimum objective function value as z, else output the feasible solution nonexistence for AMSD(h).

Proof. See Appendix B. □

From Lemma 3, it follows that four binary search iterative algorithms $\{$AMSD(h)-bs$\,|\, h = 1, 2, 3, 4\}$ with theoretically higher computational efficiency are derived from four original ones $\{$AMSD(h)-A$\,|\, h = 1, 2, 3, 4\}$ just by replacement of CUAM(g)-A in each of the four original ones with CUAM(g)-bs. Also, with h as 1 or 2, the optimum objective function value z output by AMSD(h)-A and AMSD(h)-bs, denoted as $T_{1\min}$ or $T_{2\min}$ respectively, is the first or second stage minimum feasible shipping time for C2SDM respectively.

TABLE 3. The stipulations and time complexities of solution algorithms for four auxiliary models $\{\text{AMSD}(h) \mid h = 1, 2, 3, 4\}$.

h	AMSD(h)	g	CUAM(g)	Algorithm	Stipulation	Time Complexity
1	AMSD(1)	1	CUAM(1)	AMSD(1)-A	For any route$(i,j) \in I \times J$, let $u_{ij}^{(1)} = \hat{u}_{ij}$ and $u_{ij}^{(2)} = \hat{u}_{ij}$.	$O\left(\omega\left(\tilde{m} + 2\tilde{n} + \tilde{m}\tilde{n} + 2\right)^4 \log\left(\sum_{j=1}^{\tilde{n}}\left(d_j^{(1)} + d_j^{(2)}\right)\right)\right)$
				AMSD(1)-bs		$O\left(\left(\log\omega\right)\left(\tilde{m} + 2\tilde{n} + \tilde{m}\tilde{n} + 2\right)^4 \log\left(\sum_{j=1}^{\tilde{n}}\left(d_j^{(1)} + d_j^{(2)}\right)\right)\right)$
2	AMSD(2)	2	CUAM(2)	AMSD(2)-A	For any route$(i,j) \in I \times J$, let $u_{ij}^{(1)} = \hat{u}_{ij}$ and $u_{ij}^{(2)} = \hat{u}_{ij}$.	$O\left(\omega\left(\tilde{m} + 2\tilde{n} + \tilde{m}\tilde{n} + 2\right)^4 \log\left(\sum_{j=1}^{\tilde{n}}\left(d^{(1)} + d^{(2)}\right)\right)\right)$
				AMSD(2)-bs		$O\left(\left(\log\omega\right)\left(\tilde{m} + 2\tilde{n} + \tilde{m}\tilde{n} + 2\right)^4 \log\left(\sum_{j=1}^{\tilde{n}}\left(d^{(1)} + d^{(2)}\right)\right)\right)$
3	AMSD(3)	2	CUAM(2)	AMSD(3)-A	For any route$(i,j) \in I \times J$, let $u_{ij}^{(2)} = \hat{u}_{ij}$; if $\hat{t}_{ij} \geq Tu$, then let $u_{ij}^{(1)} = 0$, else let $u_{ij}^{(1)} = \hat{u}_{ij}$.	$O\left(\omega\left(\tilde{m} + 2\tilde{n} + \tilde{m}\tilde{n} + 2\right)^4 \log\left(\sum_{j=1}^{\tilde{n}}\left(d_j^{(1)} + d_j^{(2)}\right)\right)\right)$
				AMSD(3)-bs		$O\left(\left(\log\omega\right)\left(\tilde{m} + 2\tilde{n} + \tilde{m}\tilde{n} + 2\right)^4 \log\left(\sum_{j=1}^{\tilde{n}}\left(d_j^{(1)} + d_j^{(2)}\right)\right)\right)$
4	AMSD(4)	2	CUAM(2)	AMSD(4)-A	For any route $(i,j) \in I \times J$, if $\hat{t}_{ij} > \alpha(p)$, then let $u_{ij}^{(1)} = 0$, else let $u_{ij}^{(1)} = \hat{u}_{ij}$; if $\hat{t}_{ij} \geq z_1 + z_2 - \alpha(p)$, then let $u_{ij}^{(2)} = 0$, else let $u_{ij}^{(2)} = \hat{u}_{ij}$.	$O\left(\omega\left(\tilde{m} + 2\tilde{n} + \tilde{m}\tilde{n} + 2\right)^4 \log\left(\sum_{j=1}^{\tilde{n}}\left(d_j^{(1)} + d_j^{(2)}\right)\right)\right)$
				AMSD(4)-bs		$O\left(\left(\log\omega\right)\left(\tilde{m} + 2\tilde{n} + \tilde{m}\tilde{n} + 2\right)^4 \log\left(\sum_{j=1}^{\tilde{n}}\left(d_j^{(1)} + d_j^{(2)}\right)\right)\right)$

2.4. Efficient solution approach to main model

With full exploitation of binary search, iterative search, greedy search, gradual approximating and global search idea, as well as the relationship (see Prop. 1) of main model C2SDM with four auxiliary models $\{\text{AMSD}(h) \mid h = 1, 2, 3, 4\}$, four iterative algorithms with two derived respectively from application of binary search to other two are developed to solve C2SDM in high efficiency. As such, Theorems 1 and 2 below are acquired.

Theorem 1. *In a polynomial time with $O(T_1 + \omega T_3)$ as complexity, iterative algorithm called C2SD-A1 (or C2SD-A1bs) below judges the feasible solution existence for C2SDM with optimum solution to be found if existent. Herein, T_1 and T_3 are the computation time consumed for AMSD(1)-A (or AMSD(1)-bs) and AMSD(3)-A (or AMSD(3)-bs) to solve AMSD(1) and AMSD(3) respectively, and ω is the number of distinct values in all the route shipping times.*

C2SD-A1 (or C2SD-A1bs)

Step 1: Solve AMSD(1) by AMSD(1)-A (or AMSD(1)-bs). If the optimum objective function value $T_{1\min}$ is found for AMSD(1) by AMSD(1)-A (or AMSD(1)-bs), then set $k = 0, Tu = +\infty$, $z^* = +\infty$, and go to Step 2. Else, output the feasible solution nonexistence for C2SDM, and stop.

Step 2: Solve current AMSD(3) by AMSD(3)-A (or AMSD(3)-bs). If the optimum solution with first stage shipment plan as $x^{(1)} = \{x_{ij}^{(1)} \mid (i,j) \in I \times J\}$ and second stage shipment plan as $x^{(2)} = \{x_{ij}^{(2)} \mid (i,j) \in I \times J\}$ is found for AMSD(3) by AMSD(3)-A (or AMSD(3)-bs), then go to Step 3, else go to Step 5.

Step 3: Set $z^{(1)} = \max\{\hat{t}_{ij} \mid (i,j) \in I \times J,\, x_{ij}^{(1)} > 0\}$, $z^{(2)} = \max\{\hat{t}_{ij} \mid (i,j) \in I \times J,\, x_{ij}^{(2)} > 0\}$, $z = z^{(1)} + z^{(2)}$. If $z \leq z^*$, then set $z^* = z$, $x_{ij}^{(*1)} = x_{ij}^{(1)}$ and $x_{ij}^{(*2)} = x_{ij}^{(2)}$ for any $(i,j) \in I \times J$. Set $k = k + 1$, $T_1^p(k) = z^{(1)}$, $T_2^p(k) = z^{(2)}$.

Step 4: If $T_1^p(k) > T_{1\min}$, then set $Tu = T_1^p(k)$ and go to Step 2. Else, go to Step 5.

Step 5: Output the optimum objective function value as z^* and optimum solution with first stage shipment plan as $x^{(*1)} = \{x_{ij}^{(*1)} \mid (i,j) \in I \times J\}$ and second stage shipment plan as $x^{(*2)} = \{x_{ij}^{(*2)} \mid (i,j) \in I \times J\}$ for C2SDM, and stop.

Proof. See Appendix B. □

Theorem 2. *In a polynomial time with $O(T_1+T_2+\omega T_4)$ as complexity, iterative algorithm called C2SD-A2 (or C2SD-A2bs) below judges the feasible solution existence for C2SDM with optimum solution to be found if existent. Herein, T_1, T_2 and T_4 are the computation time consumed for AMSD(1)-A (or AMSD(1)-bs), AMSD(2)-A (or AMSD(2)-bs) and AMSD(4)-A (or AMSD(4)-bs) to solve AMSD(1), AMSD(2) and AMSD(4) respectively, and ω is the number of distinct values in all the route shipping times.*

C2SD-A2 (or C2SD-A2bs)

Step 1: Produce a rigorously descent sequence $\alpha(1), \alpha(2), \ldots, \alpha(\omega)$ by sorting all the elements in $\{\hat{t}_{ij}|(i,j) \in I \times J\}$ and treating equal ones as the identical. Solve AMSD(1) by AMSD(1)-A (or AMSD(1)-bs). If the optimum objective function value $T_{1\min}$ is found for AMSD(1) by AMSD(1)-A (or AMSD(1)-bs), then find $pT_{1\min} \in \Omega = \{1, 2, \ldots, \omega\}$ by $\alpha(pT_{1\min}) = T_{1\min}$, set $k = 1$, go to Step 2. Else, output the feasible solution nonexistence for C2SDM, and stop.

Step 2: Find the optimum objective function value as $T_{2\min}$ and optimum solution with first stage shipment plan as $x^{(1)} = \{x_{ij}^{(1)}|(i,j) \in I \times J\}$ and second stage shipment plan as $x^{(2)} = \{x_{ij}^{(2)}|(i,j) \in I \times J\}$ for AMSD(2) by AMSD(2)-A (or AMSD(2)-bs). Set $z^{(1)} = \max\{\hat{t}_{ij}|(i,j) \in I \times J, x_{ij}^{(1)} > 0\}$, $z^{(2)} = T_{2\min}$, $z^{(o)} = z^{(1)} + z^{(2)}$, $x_{ij}^{(o1)} = x_{ij}^{(1)}$ and $x_{ij}^{(o2)} = x_{ij}^{(2)}$ for any $(i,j) \in I \times J$, $T_1(k) = z^{(1)}$, $T_2(k) = z^{(2)}$. Find $p = \max\{p \in \Omega|\alpha(p) \geq T_1(k)\}$.

Step 3: If $p = pT_{1\min}$, then go to Step 5. Else, set $z_1 = T_1(k)$, $z_2 = T_2(k)$, go to Step 4.

Step 4: Set $p = p + 1$. Solve current AMSD(4) by AMSD(4)-A (or AMSD(4)-bs). If the optimum solution with first stage shipment plan as $x^{(1)} = \{x_{ij}^{(1)}|(i,j) \in I \times J\}$ and second stage shipment plan as $x^{(2)} = \{x_{ij}^{(2)}|(i,j) \in I \times J\}$ is found for current AMSD(4) by AMSD(4)-A (or AMSD(4)-bs), then successively manipulate as below: "set $z^{(1)} = \max\{\hat{t}_{ij}|(i,j) \in I \times J, x_{ij}^{(1)} > 0\}$, $z^{(2)} = \max\{\hat{t}_{ij}|(i,j) \in I \times J, x_{ij}^{(2)} > 0\}$; if $z^{(1)} + z^{(2)} < z^{(o)}$, then set $z^{(o)} = z^{(1)} + z^{(2)}$, $x_{ij}^{(o1)} = x_{ij}^{(1)}$ and $x_{ij}^{(o2)} = x_{ij}^{(2)}$ for any $(i,j) \in I \times J$; set $k = k+1$, $T_1(k) = z^{(1)}$, $T_2(k) = z^{(2)}$; find $p = \max\{p \in \Omega|\alpha(p) \geq T_1(k)\}$, and go to Step 3". Else, if $p < pT_{1\min}$, then go to Step 4, else go to Step 5.

Step 5: Output the optimum objective function value as $z^{(o)}$ and optimum solution with first stage shipment plan as $x^{(o1)} = \{x_{ij}^{(o1)}|(i,j) \in I \times J\}$ and second stage shipment plan as $x^{(o2)} = \{x_{ij}^{(o2)}|(i,j) \in I \times J\}$ for C2SDM, and stop.

Proof. See Appendix B. □

In reality, C2SD-A1bs and C2SD-A2bs are born from C2SD-A1 and C2SD-A2 respectively by just substituting AMSD(1)-A, AMSD(2)-A, AMSD(3)-A and AMSD(4)-A in C2SD-A1 or C2SD-A2 with AMSD(1)-bs, AMSD(2)-bs, AMSD(3)-bs and AMSD(4)-bs respectively. To intuitively showcase the relationship derived from Lemmas 1–3 and Theorems 1 and 2 for the algorithms used in solving main model C2SDM and four auxiliary models and two commonly used auxiliary models, Figure 2 is provided as relationship chart, in which rectangle represents binary search or algorithm, line with arrow represents cause result link with head rectangle as result and tail rectangle as cause, black solid line and blue dashed line disjoint midway. For instance, as is shown in Figure 2, C2SD-A1 is born from AMSD(1)-A and AMSD(3)-A. Also, AMSD(2)-bs, AMSD(3)-bs and AMSD(4)-bs all are born only from CUAM(2)-bs, and CUAM(2)-bs is born from binary search and CUAM(2)-A.

As is derived from Theorem 1 in Xie and Li [43] as well as Lemmas 1–3 and Theorems 1 and 2 herein, all of our four algorithms (C2SD-A1, C2SD-A1bs, C2SD-A2, C2SD-A2bs) are polynomial time algorithm for solving C2SD. As is worth mentioning, theoretically speaking, in computation time, C2SD-A2 and C2SD-A2bs greatly outperform C2SD-A1 and C2SD-A1bs respectively because of their fast descent characteristics derived from greedy search, and C2SD-A1bs and C2SD-A2bs vastly excel C2SD-A1 and C2SD-A2 respectively because of time reduction role played by binary search. In particular, there exists no memory overflow in computer implementation for our four algorithms especially when applied to large size instances owing to no adoption of

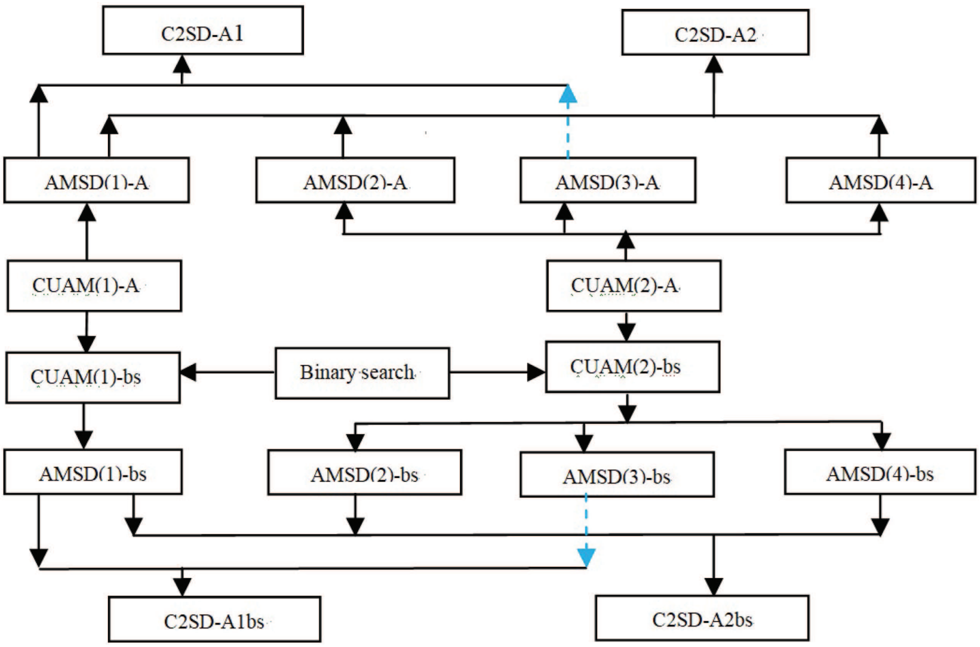

FIGURE 2. The relationship chart of solution algorithms for main model C2SDM and four auxiliary models and two commonly used auxiliary models.

TABLE 4. The space and time complexity for C2SD-A1, C2SD-A1bs, C2SD-A2 and C2SD-A2bs.

Algorithm	Time complexity	Space complexity
C2SD-A1	$O\left(\omega(\omega+1)(\tilde{m}+2\tilde{n}+\tilde{m}\tilde{n}+2)^4 \log\left(\sum_{j=1}^{\tilde{n}}\left(d_j^{(1)}+d_j^{(2)}\right)\right)\right)$	$O\left((\tilde{m}+2\tilde{n}+\tilde{m}\tilde{n}+2)^2\right)$
C2SD-A1bs	$O\left(\left(\log \omega^{(\omega+1)}\right)(\tilde{m}+2\tilde{n}+\tilde{m}\tilde{n}+2)^4 \log\left(\sum_{j=1}^{\tilde{n}}\left(d_j^{(1)}+d_j^{(2)}\right)\right)\right)$	
C2SD-A2	$O\left(\omega(\omega+2)(\tilde{m}+2\tilde{n}+\tilde{m}\tilde{n}+2)^4 \log\left(\sum_{j=1}^{\tilde{n}}\left(d_j^{(1)}+d_j^{(2)}\right)\right)\right)$	
C2SD-A2bs	$O\left(\left(\log \omega^{(\omega+2)}\right)(\tilde{m}+2\tilde{n}+\tilde{m}\tilde{n}+2)^4 \log\left(\sum_{j=1}^{\tilde{n}}\left(d_j^{(1)}+d_j^{(2)}\right)\right)\right)$	

the equivalent weights technique proposed in Sherali [33]. As a consequence, our four algorithms are a kind of exact solution technique with high robustness and efficiency developed for C2SD.

Also, as per Theorems 1 and 2 and Lemma 3, it is easy to derive Corollary 1 below.

Corollary 1. *All the four algorithms C2SD-A1, C2SD-A1bs, C2SD-A2 and C2SD-A2bs for solving C2SDM are polynomial time algorithm with complexities given in Table 4, where the parameters* $\tilde{m}$, $\tilde{n}$, ω, $d_{ij}^{(1)}$ *and* $d_{ij}^{(2)}$ *are defined in Section 2.1.*

With regard to four algorithms C2SD-A1, C2SD-A1bs, C2SD-A2 and C2SD-A2bs for solving C2SD, the relationship of space complexity with time complexity is clearly reflected in Table 4. As is derived from Table 4, there is positive correlation between space and time complexity for any of the four algorithms, *i.e.*, time complexity ascends with space complexity increase, and *vice versa*.

TABLE 5. The origin available supply, the first and second stage destination demand, and the route shipping time and capacity for 6×4 problem 1.

$\hat{t}_{ij}$	1	2	3	4	$\hat{u}_{ij}$	1	2	3	4	$\hat{s}_i$
1	5	6	4	3	1	10	20	15	15	30
2	7	9	12	10	2	20	20	10	15	40
3	2	8	7	4	3	15	15	15	20	45
4	11	5	9	8	4	15	10	20	20	25
5	6	10	5	3	5	20	20	20	20	50
6	12	4	2	10	6	10	20	20	15	20
$\hat{d}_j^{(1)}$	50	40	30	40	$\hat{d}_j^{(2)}$	10	10	10	20	

3. Illustrative examples

Distinct size examples are presented in this section to showcase the application and practical performance difference of our four algorithms (C2SD-A1, C2SD-A2, C2SD-A1bs, C2SD-A2bs), as compared with generic solver LINGO applied to main model C2SDM. For this, MATLAB is used to code the four algorithms, and the LINGO computation program based on C2SDM model is designed. All the computation programs, including the four algorithms based MATLAB programs and the C2SDM model based LINGO program, are run on PC with Intel(R) Core(TM) i7-10510U CPU @ 1.80 GHz 2.30 GHz and RAM 40.0 GB to conduct all the presented computation experiments.

3.1. Two small size instances

Four algorithms (C2SD-A1, C2SD-A2, C2SD-A1bs, C2SD-A2bs) and generic solver LINGO are first applied to solve an instance of C2SD with six origins and four destinations, *i.e.*, 6×4 problem 1, with data comprising the route shipping time ($\hat{t}_{ij}$), the route shipping capacity ($\hat{u}_{ij}$), the origin available supply ($\hat{s}_i$), the first stage destination demand ($\hat{d}_j^{(1)}$) and the second stage destination demand ($\hat{d}_j^{(2)}$) given in Table 5.

Solution: From Table 5, the known parameters are derived below. Clearly, the origin number $\tilde{m} = 6$, and the destination number $\tilde{n} = 4$. Also, the origin supply vector $\hat{s} = (\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_6)$, the first stage destination demand vector $\hat{d}^{(1)} = (\hat{d}_1^{(1)}, \hat{d}_2^{(1)}, \hat{d}_3^{(1)}, \hat{d}_4^{(1)})$, the second stage destination demand vector $\hat{d}^{(2)} = (\hat{d}_1^{(2)}, \hat{d}_2^{(2)}, \hat{d}_3^{(2)}, \hat{d}_4^{(2)})$, the route shipping time matrix $\hat{T} = (\hat{t}_{ij})_{6\times 4}$, and the route shipping capacity matrix $\hat{u} = (\hat{u}_{ij})_{6\times 4}$ are as follows.

$$\hat{s} = (30, 40, 45, 25, 50, 20); \quad \hat{d}^{(1)} = (50, 40, 30, 40); \quad \hat{d}^{(2)} = (10, 10, 10, 20)$$

$$\hat{T} = \begin{pmatrix} 5 & 6 & 4 & 3 \\ 7 & 9 & 12 & 10 \\ 2 & 8 & 7 & 4 \\ 11 & 5 & 9 & 8 \\ 6 & 10 & 5 & 3 \\ 12 & 4 & 2 & 10 \end{pmatrix} \quad \hat{u} = \begin{pmatrix} 10 & 20 & 15 & 15 \\ 20 & 20 & 10 & 15 \\ 15 & 15 & 15 & 20 \\ 15 & 10 & 20 & 20 \\ 20 & 20 & 20 & 20 \\ 10 & 20 & 20 & 15 \end{pmatrix}.$$

As a result, the identical optimum solution (see Tab. 6), with duration as 13 and the first and second stage shipping time pair as $(9, 4)$, is found for 6×4 problem 1 by C2SD-A1, C2SD-A2, C2SD-A1bs and C2SD-A2bs based MATLAB computation program using the above known parameters $(\tilde{m}, \tilde{n}, \hat{s}, \hat{d}^{(1)}, \hat{d}^{(2)}, \hat{T}, \hat{u})$ as input in a runtime of 0.032034, 0.018919, 0.023323 and 0.022833 s respectively.

Clearly, main model (C2SDM) based LINGO computation program given in Appendix B is usable for solving 6×4 problem 1. However, LINGO finds only a feasible solution (see Tab. 7) to 6×4 problem 1, with duration

TABLE 6. The identical optimum solution with duration as 13 and the first and second stage shipping time pair as $(9, 4)$ found for 6×4 problem 1 by our four algorithms.

$x_{ij}^{(1)}$	1	2	3	4	$x_{ij}^{(2)}$	1	2	3	4	$\hat{s}_i$
1	10	5	5		1			10		30
2	20	20			2					40
3		10	5	20	3	10				45
4		5		20	4					25
5	20		10		5				20	50
6			10		6		10			20
$\hat{d}_j^{(1)}$	50	40	30	40	$\hat{d}_j^{(2)}$	10	10	10	20	

TABLE 7. The feasible solution with duration as 24 and the first and second stage shipping time pair as $(12, 12)$ found for 6×4 problem 1 by LINGO.

$x_{ij}^{(1)}$	1	2	3	4	$x_{ij}^{(2)}$	1	2	3	4	$\hat{s}_i$
1	9	12	7		1			2		30
2	14	5	6	4	2	3	4	1	3	40
3	6	7	12	12	3	2	2	2	2	45
4	10	4	2	5	4	1	1	2		25
5	9	8	3	5	5	4	3	3	15	50
6	2	4		14	6					20
$\hat{d}_j^{(1)}$	50	40	30	40	$\hat{d}_j^{(2)}$	10	10	10	20	

as 24, much more than 13 as the duration of optimum solution found for 6×4 problem 1 by our four algorithms (C2SD-A1, C2SD-A2, C2SD-A1bs, C2SD-A2bs). Also, the runtime for finding the feasible solution to 6×4 problem 1 by LINGO is 0.70 s, much more than that by our four algorithms.

Next, 6×4 problem 2 given in Table 8 is resolved similarly to 6×4 problem 1 by application of our four algorithms and generic solver LINGO. As the consequence of solving, C2SD-A1, C2SD-A2, C2SD-A1bs and C2SD-A2bs find the identical optimum solution (see Tab. 9) for 6×4 problem 2, with duration as 319 and the first and second stage shipping time pair as $(112, 207)$, in a runtime of 0.03819, 0.020414, 0.035851 and 0.02673 s respectively. However, LINGO finds only a feasible solution (see Tab. 10) with duration as 685 and the first and second stage shipping time pair as $(337, 348)$ for 6×4 problem 2 in a runtime of 0.75 s.

3.2. Two medium size instances

Four algorithms (C2SD-A1, C2SD-A2, C2SD-A1bs, C2SD-A2bs) and generic solver LINGO are first applied to a medium size instance of C2SD, *i.e.*, 10×5 problem 1, with data comprising the route shipping time $(\hat{t}_{ij})$, the origin supply $(\hat{s}_i)$, the first stage destination demand $(\hat{d}_j^{(1)})$ and the second stage destination demand $(\hat{d}_j^{(2)})$ given in Table 11.

Solution: From Table 11, the known parameters are derived below. It is evident that the origin number $\tilde{m} = 10$ and the destination number $\tilde{n} = 5$. Also, the origin supply vector $\hat{s} = (\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_{10})$, the first stage destination demand vector $\hat{d}^{(1)} = (\hat{d}_1^{(1)}, \hat{d}_2^{(1)}, \ldots, \hat{d}_5^{(1)})$, the second stage destination demand vector $\hat{d}^{(2)} = (\hat{d}_1^{(2)}, \hat{d}_2^{(2)}, \ldots, \hat{d}_5^{(2)})$, the route shipping time matrix $\hat{T} = (\hat{t}_{ij})_{10 \times 5}$, and the route shipping capacity matrix $\hat{u} = (\hat{u}_{ij})_{10 \times 5}$ are obtained below, where positive infinity ∞ is generally substituted by sufficiently big positive

TABLE 8. The origin available supply, the first and second stage destination demand, the route shipping time and capacity for 6×4 problem 2.

$\hat{t}_{ij}$	1	2	3	4	$\hat{u}_{ij}$	1	2	3	4	$\hat{s}_i$
1	235	373	207	133	1	13	48	77	20	30
2	348	104	104	72	2	76	10	79	51	26
3	139	2	297	337	3	48	73	46	20	33
4	112	282	162	325	4	69	6	36	61	33
5	297	176	32	165	5	20	31	32	43	26
6	136	285	125	320	6	22	47	17	7	26
$\hat{d}_j^{(1)}$	1	11	19	9	$\hat{d}_j^{(2)}$	37	30	37	30	

TABLE 9. The identical optimum solution with duration as 319 and the first and second stage shipping time pair as $(112, 207)$ found for 6×4 problem 2 by our four algorithms.

$x_{ij}^{(1)}$	1	2	3	4	$x_{ij}^{(2)}$	1	2	3	4	$\hat{s}_i$
1					1			10	20	30
2			7	9	2				10	26
3		11			3	6	16			33
4	1				4	14		18		33
5			12		5		14			26
6					6	17		9		26
$\hat{d}_j^{(1)}$	1	11	19	9	$\hat{d}_j^{(2)}$	37	30	37	30	

number in practical computation.

$$\hat{s} = (20, 30, 10, 15, 15, 25, 10, 25, 10, 15); \quad \hat{d}^{(1)} = (10, 10, 15, 10, 20); \quad \hat{d}^{(2)} = (20, 30, 20, 20, 20)$$

$$\hat{T} = \begin{pmatrix} 2 & 3 & 5 & 11 & 4 \\ 4 & 7 & 9 & 5 & 10 \\ 12 & 25 & 9 & 6 & 26 \\ 8 & 7 & 9 & 24 & 10 \\ 3 & 5 & 6 & 7 & 9 \\ 11 & 24 & 20 & 12 & 3 \\ 11 & 1 & 5 & 4 & 15 \\ 9 & 8 & 3 & 10 & 4 \\ 8 & 7 & 10 & 5 & 4 \\ 19 & 6 & 16 & 2 & 3 \end{pmatrix} \quad \hat{u} = \begin{pmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \end{pmatrix}.$$

With input of the parameters $(\tilde{m}, \tilde{n}, \hat{s}, \hat{d}^{(1)}, \hat{d}^{(2)}, \hat{T}, \hat{u})$, C2SD-A1, C2SD-A2, C2SD-A1bs and C2SD-A2bs based MATLAB computation program find the identical optimum solution (see Tab. 12) for 10×5 problem 1, with duration as 10 and the first and second stage shipping time pair as $(3, 7)$, in a runtime of 0.115766, 0.038508, 0.073076 and 0.051012 s respectively.

Also, 10×5 problem 1 is solved similarly to 6×4 problem 1 by application of generic solver LINGO. However, only a feasible solution (see Tab. 13) with duration as 49 and the first and second stage shipping time pair as $(25, 24)$ is found for 10×5 problem 1 by LINGO in a runtime of 3.68 s.

TABLE 10. The feasible solution with duration as 685 and the first and second stage shipping time pair as $(337, 348)$ found for 6×4 problem 2 by LINGO.

$x_{ij}^{(1)}$	1	2	3	4		$x_{ij}^{(2)}$	1	2	3	4	$\hat{s}_i$
1			7			1	13		10		30
2		2	4	1		2	3	2	7	7	26
3		1	2	1		3	6	21	1	1	33
4		3	2	2		4	3	2	4	17	33
5	1	2	3	1		5	6	5	6	2	26
6		3	1	4		6	6		9	3	26
$\hat{d}_j^{(1)}$	1	11	19	9		$\hat{d}_j^{(2)}$	37	30	37	30	

TABLE 11. The origin available supply, the first and second stage destination demand, and the route shipping time for 10×5 problem 1.

$\hat{t}_{ij}$	1	2	3	4	5	$\hat{s}_i$
1	2	3	5	11	4	20
2	4	7	9	5	10	30
3	12	25	9	6	26	10
4	8	7	9	24	10	15
5	3	5	6	7	9	15
6	11	24	20	12	3	25
7	11	1	5	4	15	10
8	9	8	3	10	4	25
9	8	7	10	5	4	10
10	19	6	16	2	3	15
$\hat{d}_j^{(1)}$	10	10	15	10	20	
$\hat{d}_j^{(2)}$	20	30	20	20	20	

Subsequently, our four algorithms and generic solver LINGO are applied to 10×5 problem 2 given in Table 14 by running the MATLAB computation programs based on our four algorithms and the LINGO computation program based on C2SDM model. As the solving result, the identical optimum solution (see Tab. 15) with duration as 151 and the first and second stage shipping time pair as (48, 103) is found for 10×5 problem 2 by C2SD-A1, C2SD-A2, C2SD-A1bs and C2SD-A2bs in a runtime of 0.228574, 0.096516, 0.164895 and 0.112312 s respectively. But only a feasible solution (see Tab. 16) with duration as 240 and the first and second stage shipping time pair as $(120, 120)$ is found for 10×5 problem 2 by LINGO in a runtime of 4.11 s.

3.3. Two large size instances

Four algorithms C2SD-A1, C2SD-A2, C2SD-A1bs and C2SD-A2bs are first applied to 20×10 problem given in Table 17 by running the MATLAB computation programs based on the four algorithms. As is observable, C2SD-A1 and C2SD-A1bs find the identical optimum solution (see Tab. 18) with duration as 149 and the first and second stage shipping time pair as $(66, 83)$ in a runtime of 71.46616 and 26.22227 s respectively. But C2SD-A2 and C2SD-A2bs find the identical optimum solution (see Tab. 19) with duration as 149 and the first and

TABLE 12. The identical optimum solution with duration as 10 and the first and second stage shipping time pair as $(3,7)$ found for 10×5 problem 1 by our four algorithms.

$x_{ij}^{(1)}$	1	2	3	4	5	$x_{ij}^{(2)}$	1	2	3	4	5	$\hat{s}_i$
1	10					1	5				5	20
2						2	10	10		10		30
3						3				10		10
4						4		15				15
5						5	5		10			15
6					20	6					5	25
7		10				7						10
8			15			8			10			25
9						9					10	10
10				10		10		5				15
$\hat{d}_j^{(1)}$	10	10	15	10	20	$\hat{d}_j^{(2)}$	20	30	20	20	20	

TABLE 13. The feasible solution with duration as 49 and the first and second stage shipping time pair as $(25,24)$ found for 10×5 problem 1 by LINGO.

$x_{ij}^{(1)}$	1	2	3	4	5	$x_{ij}^{(2)}$	1	2	3	4	5	$\hat{s}_i$
1						1			13	7		20
2			5			2	14	11				30
3		7		3		3						10
4	7		1	3	1	4				3		15
5	1	1	2	1	2	5		3	3		2	15
6			4		1	6	4	13		3		25
7		1		2	2	7		2		3		10
8	1		3		10	8		1			10	25
9					2	9	2		2	4		10
10	1	1		1	2	10			2		8	15
$\hat{d}_j^{(1)}$	10	10	15	10	20	$\hat{d}_j^{(2)}$	20	30	20	20	20	

second stage shipping time pair as $(83,66)$ in a runtime of 9.397827 and 13.25411 s respectively. Besides, 20×10 problem is solved by the C2SDM model based LINGO computation program finding only a feasible solution (see Tab. 20) with duration as 280 and the first and second stage shipping time pair as $(140,140)$ in a runtime of 42.25 s.

Next, four algorithms C2SD-A1, C2SD-A2, C2SD-A1bs and C2SD-A2bs are applied to 20×20 problem given in Tables 21 and 22 by running the MATLAB computation programs based on the four algorithms. As the solving result, the identical optimum solution (see Tabs. 23 and 24) with duration as 210 and the first and second stage shipping time pair as (105, 105) is found for 20×20 problem by C2SD-A1, C2SD-A2, C2SD-A1bs and C2SD-A2bs in a runtime of 460.6666, 14.26211, 45.03909 and 36.44326 s respectively. Also, 20×20 problem is solved by the C2SDM model based LINGO computation program finding only a feasible solution (see Tabs. 25 and 26) with duration as 779 and the first and second stage shipping time pair as $(382,397)$ in a runtime of 39.66 s.

TABLE 14. The origin available supply, the first and second stage destination demand, and the route shipping time and capacity for 10×5 problem 2.

$\hat{t}_{ij}$	1	2	3	4	5	$\hat{u}_{ij}$	1	2	3	4	5	$\hat{s}_i$
1	117	108	25	115	56	1	43	38	45	20	26	57
2	73	97	25	79	67	2	22	23	30	23	20	55
3	50	82	85	46	48	3	36	40	36	45	22	56
4	103	103	79	119	112	4	25	41	34	31	24	53
5	42	90	118	44	73	5	41	48	36	22	43	53
6	30	120	88	21	78	6	32	34	35	26	30	58
7	30	30	100	48	24	7	22	38	25	48	23	50
8	49	58	50	115	118	8	33	28	47	43	28	58
9	60	48	36	36	85	9	40	27	22	20	30	58
10	61	61	91	52	83	10	44	29	27	34	27	57
$\hat{d}_j^{(1)}$	30	34	39	38	34	$\hat{d}_j^{(2)}$	80	80	80	80	60	

TABLE 15. The identical optimum solution with duration as 151 and the first and second stage shipping time pair as $(48, 103)$ found for 10×5 problem 2 by our four algorithms.

$x_{ij}^{(1)}$	1	2	3	4	5	$x_{ij}^{(2)}$	1	2	3	4	5	$\hat{s}_i$
1			32			1					25	57
2			7			2		1	5	23	19	55
3				35	21	3						56
4						4		41	12			53
5	7			3		5		14		13	16	53
6	21					6			16	21		58
7	2	12			13	7				23		50
8						8		11	47			58
9		22				9	36					58
10						10	44	13				57
$\hat{d}_j^{(1)}$	30	34	39	38	34	$\hat{d}_j^{(2)}$	80	80	80	80	60	

4. PERFORMANCE COMPARISON AND ANALYSIS

In the section, the causes for incurring the performance difference of our four algorithms (C2SD-A1, C2SD-A2, C2SD-A1bs, C2SD-A2bs) compared with generic solver LINGO, are analyzed.

It is the common idea of our four algorithms to iteratively resolve a collection of C2SDM's auxiliary models by constructing network and calling Xie and Li [43]'s feasible flow algorithm FFn-A for high efficiency solution to C2SDM model. For this, a finite sequence of the pair of the first and second stage shipping time is produced, so that in the produced sequence, each pair of the first and second stage shipping time corresponds to a feasible solution to C2SDM, and the pair of the first and second stage shipping time, corresponding to the optimum solution to C2SDM, is included. Also, in the produced sequence of pair of the first and second stage shipping time, for any pair, the first stage shipping time rigorously descends; and the second stage shipping time monotonously ascends for C2SD-A1 (or C2SD-A1bs) but not for C2SD-A2 (or C2SD-A2bs); and consequently the sum of the first and second stage shipping time in any pair is not guaranteed to descend rigorously for any of our four algorithms.

TABLE 16. The feasible solution with duration as 240 and the first and second stage shipping time pair as $(120, 120)$ found for 10×5 problem 2 by LINGO.

$x_{ij}^{(1)}$	1	2	3	4	5		$x_{ij}^{(2)}$	1	2	3	4	5	$\hat{s}_i$
1							1	5	13	19	20		57
2	11	1			1		2	10	16	8	5	3	55
3	5	16	4	11	4		3			6	9	1	56
4	2	4	7	5	4		4	6	7	7	6	5	53
5	2	3	4	2	5		5		18			19	53
6	2	4	5	5	4		6	9	8	8	7	6	58
7	2		1	4	5		7	5	5		21	7	50
8	2	2	12	1	4		8	7	2	21	3	4	58
9	2	4	5	9	4		9	8	8	5	7	6	58
10	2		1	1	3		10	30	3	6	2	9	57
$\hat{d}_j^{(1)}$	30	34	39	38	34		$\hat{d}_j^{(2)}$	80	80	80	80	60	

TABLE 17. The origin available supply, the first and second stage destination demand, and the route shipping time and capacity for 20×10 problem.

$\hat{t}_{ij}$	1	2	3	4	5	6	7	8	9	10		$\hat{u}_{ij}$	1	2	3	4	5	6	7	8	9	10	$\hat{s}_i$
1	123	123	99	139	132	62	110	138	64	93		1	40	65	64	64	50	53	57	64	57	46	77
2	50	140	108	41	98	50	50	120	68	44		2	46	51	57	50	56	53	57	41	56	46	75
3	69	78	70	135	138	80	68	56	56	105		3	69	68	70	66	52	64	70	48	45	61	76
4	81	81	111	72	103	60	58	98	48	86		4	40	50	61	43	43	49	52	57	49	49	73
5	131	66	119	78	69	132	103	103	83	49		5	65	49	51	52	66	52	62	60	52	69	73
6	96	110	132	124	42	94	132	83	108	90		6	60	55	46	58	63	54	68	48	60	67	78
7	91	86	75	80	67	45	64	138	46	79		7	48	51	40	49	70	63	58	58	48	55	70
8	76	89	55	87	65	103	94	55	134	106		8	64	66	42	56	52	42	53	61	55	52	78
9	91	79	50	119	86	116	100	124	41	61		9	69	70	58	56	59	42	45	48	40	44	78
10	47	50	73	52	40	94	106	94	123	48		10	53	57	57	64	41	52	41	49	58	56	77
11	59	108	85	76	55	111	133	93	49	116		11	58	50	68	63	57	53	62	45	41	47	70
12	80	86	89	60	73	49	99	57	133	49		12	63	69	50	61	64	41	64	54	46	58	74
13	84	67	128	115	67	108	65	49	43	72		13	68	69	60	40	60	69	40	69	42	49	79
14	119	70	63	88	65	74	44	88	60	127		14	58	65	41	41	57	49	52	55	62	53	78
15	99	116	133	73	94	48	104	81	137	51		15	49	42	52	57	63	57	60	65	52	64	74
16	133	102	75	55	88	62	140	53	42	74		16	62	61	62	45	63	42	47	69	65	46	80
17	95	133	94	81	125	123	107	112	140	74		17	45	47	68	47	47	62	60	63	63	49	79
18	90	81	110	58	82	94	122	94	83	91		18	65	62	67	46	51	57	51	46	57	59	70
19	63	102	89	108	129	77	70	69	55	93		19	44	41	59	65	59	53	45	55	70	63	80
20	62	99	76	128	88	59	109	115	102	118		20	51	55	56	60	41	60	67	50	43	53	74
$\hat{d}_j^{(1)}$	55	58	50	56	55	53	56	57	52	53		$\hat{d}_j^{(2)}$	80	90	80	80	120	88	90	100	110	130	

As is notable, due to the algorithm flow process designed with greedy search, C2SD-A2 and C2SD-A2bs have fast descent characteristics respectively in comparison with C2SD-A1 and C2SD-A1bs. Also, C2SD-A1bs and C2SD-A2bs are gotten from C2SD-A1 and C2SD-A2 respectively, and the only difference of C2SD-A1bs (or C2SD-A2bs) from C2SD-A1 (or C2SD-A2) is that C2SD-A1bs (or C2SD-A2bs) calls binary search iterative algorithm rather than respective original one to solve all the necessary auxiliary models.

Due to application of binary search, binary search iterative algorithm significantly outperforms original one in computation time. For this reason, theoretically speaking, in computation time, C2SD-A1bs and C2SD-A2bs significantly excel C2SD-A1 and C2SD-A2 respectively. Also, theoretically, with fast descent characteristics, C2SD-A2 and C2SD-A2bs outdo C2SD-A1 and C2SD-A1bs respectively in computation time. But according to

TABLE 18. The identical optimal solution with duration as 149 and first and second stage shipping time pair as $(66, 83)$ found for 20×10 problem by C2SD-A1 and C2SD-A1bs.

$x_{ij}^{(1)}$	1	2	3	4	5	6	7	8	9	10	$x_{ij}^{(2)}$	1	2	3	4	5	6	7	8	9	10	$\hat{s}_i$
1						1			52		1						23			1		77
2						1	3			10	2							35		26		75
3											3		2	7			2	55	10			76
4						11	52				4									10		73
5		1								43	5					6				3	20	73
6					55						6					2			21			78
7							1				7					7	7				55	70
8			9					2			8	25		11		31						78
9			18								9		23	37								78
10		57									10	15			5							77
11	55										11				2	13						70
12				56							12	12				6						74
13								55			13		10						14			79
14			23								14					55						78
15						40					15				28						6	74
16											16			25					55			80
17											17				30						49	79
18											18		55		15							70
19											19	10								70		80
20											20	18					56					74
$\hat{d}_j^{(1)}$	55	58	50	56	55	53	56	57	52	53	$\hat{d}_j^{(2)}$	80	90	80	80	120	88	90	100	110	130	

TABLE 19. The identical optimum solution with duration as 149 and the first and second stage shipping time pair as $(83, 66)$ found for 20×10 problem by C2SD-A2 and C2SD-A2bs.

$x_{ij}^{(1)}$	1	2	3	4	5	6	7	8	9	10	$x_{ij}^{(2)}$	1	2	3	4	5	6	7	8	9	10	$\hat{s}_i$
1						1			7		1						24			45		77
2						3			17	4	2	4			1		1	25			20	75
3	1		1			36	1		13		3								7	17		76
4				8		13	52				4											73
5				4							5		33								36	73
6								15			6					63						78
7			7								7						5	58				70
8	14		7					4			8			29					24			78
9									15		9			30						20	13	78
10			13								10		57		7							77
11				1							11	58								11		70
12	17										12				57							74
13		3			55		3				13							7	11			79
14											14			21		57						78
15				13							15										61	74
16			22								16								58			80
17				30						49	17											79
18		55									18				15							70
19	23							38			19	2								17		80
20											20	16					58					74
$\hat{d}_j^{(1)}$	55	58	50	56	55	53	56	57	52	53	$\hat{d}_j^{(2)}$	80	90	80	80	120	88	90	100	110	130	

Theorem 1 in Xie and Li [43] as well as Lemmas 1–3 and Theorems 1 and 2 and Corollary 1 herein, all of our four algorithms can find the optimum solution to C2SDM in a polynomial time.

As is revealed from the above analyses, theoretically speaking, our four algorithms are very good exact solution technique for C2SD as compared to generic solver LINGO. Also, in computation time, C2SD-A2bs ranks best, C2SD-A2 next, C2SD-A1bs third, C2SD-A1 fourth, and C2SD-A1bs is significantly superior to C2SD-A1, C2SD-A2 is much better than C2SD-A1bs, and C2SD-A2bs has significant preponderance over C2SD-A2. In particular,

TABLE 20. The feasible solution with duration as 280 and the first and second stage shipping time pair as $(140, 140)$ found for 20×10 problem by LINGO.

$x_{ij}^{(1)}$	1	2	3	4	5	6	7	8	9	10	$x_{ij}^{(2)}$	1	2	3	4	5	6	7	8	9	10	$\hat{s}_i$
1	5	1		4	3	3	7	3		3	1				6	14	7	11	3	1	6	77
2	4	3	3	3		1	3	2	2	1	2		13	17	2	2	4		6	6	3	75
3	5	4	3	7	2	2	3	5	3	1	3		1	9	4	6	4	1	6	6	4	76
4	1	3	3	4	3	2	2	2	3	3	4		5	5	5	6	4	5	5	6	6	73
5	3	3	3	3	3	3	3	3	3	3	5	3	6	1	5	5	4	5	1	6	7	73
6	3	3	3	2	3	3	3	3	3	3	6	5	5	5	2	4	4	5	7	6	6	78
7		3	2	4	3	2	3	2	3	3	7	4	4	4	6	3	7	4	3	5	5	70
8	2	3	3	3	3	4	1	1	3	3	8	6	5	4	5	6	4	5	5	6	6	78
9	2	3	3	3	3	2	4	4	3	3	9	7	2	4	3	6	4	5	5	6	6	78
10	3	1	3	3	2	3		2	3	3	10	9	5	1	5	6	4	4	8	6	6	77
11	3	1	1	2	4	3	2	5	3	3	11		4	4	4	6	4	3	5	5	8	70
12	3	3	3	1	3	3	5	3	2	3	12	3	5	1	5	6	4	5	6	6	4	74
13	3	3	3	2	2	2	3		1	3	13	8	5	4	4	5	8	5	4	6	8	79
14	3	3	3	2	3	3	3	3	4	1	14	6	5	3	4	6	4	4	6	6	6	78
15	2	3	3	1	3	2	3	4	3	3	15	3	4	7	5	6	4	5	2	7	4	74
16	3	3	3	2	3	3		1	3	3	16	10	4	2	3	13	4	4	5	5	6	80
17	3	3	3	3	3	3	3	7	3	3	17	7	5	5		1	3	6	5	6	7	79
18	2	3	2	3	2	2	3	3	2	2	18			1	3	4		4	8	6	20	70
19	3	6		2	3	5	2	3	3	1	19	4	6	3	4	10	6	4	4	5	6	80
20	2	3	3	2	4	2	3	1	2	5	20	5	6		5	5	5	5	6	4	6	74
$\hat{d}_j^{(1)}$	55	58	50	56	55	53	56	57	52	53	$\hat{d}_j^{(2)}$	80	90	80	80	120	88	90	100	110	130	

TABLE 21. The origin available supply, the first stage destination demand, and the route shipping time for 20×20 problem.

$\hat{t}_{ij}$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	$\hat{s}_i$
1	165	166	286	131	254	84	75	234	33	184	363	105	315	152	116	368	253	252	172	40	31
2	225	278	366	334	10	218	367	173	272	201	206	186	142	162	108	23	98	392	25	157	36
3	146	196	63	190	103	252	217	63	376	262	203	157	43	314	184	302	239	334	8	85	22
4	30	43	133	52	1	215	263	218	331	33	77	272	182	143	60	282	372	213	36	304	32
5	161	185	197	84	132	39	236	68	372	40	178	110	350	301	110	270	103	36	13	130	30
6	317	119	95	193	102	137	18	193	83	346	236	302	372	133	218	33	254	165	385	46	26
7	370	249	140	60	192	88	398	53	12	139	220	370	216	163	339	331	269	289	399	136	33
8	199	166	279	72	170	218	326	217	172	204	92	248	196	273	355	149	121	118	61	212	33
9	90	234	146	351	192	77	274	299	246	313	65	324	82	383	27	25	318	152	186	48	26
10	47	70	20	286	214	225	87	188	299	301	160	362	299	36	254	286	7	173	161	111	26
11	395	322	279	168	294	111	143	174	378	49	259	140	42	75	32	174	384	217	198	390	37
12	88	152	159	113	202	56	207	387	224	364	263	177	278	26	303	281	199	63	90	131	37
13	314	21	208	303	321	131	390	322	270	363	351	167	50	382	319	279	161	7	68	66	32
14	204	163	43	111	258	340	200	76	359	150	130	309	88	179	95	352	245	150	156	345	40
15	235	373	207	133	348	104	104	72	139	2	297	337	112	282	162	325	297	176	32	165	39
16	136	285	125	320	61	238	383	98	376	46	394	254	240	362	230	99	345	30	176	304	25
17	99	152	159	211	109	233	84	32	359	45	261	361	93	380	339	177	197	308	334	154	34
18	79	131	166	62	248	40	83	278	202	74	397	189	2	171	116	302	357	329	71	44	40
19	5	61	151	76	392	349	233	292	59	104	12	332	314	317	134	181	223	322	226	82	25
20	84	146	230	138	212	171	230	21	220	83	386	374	398	336	164	323	396	104	70	278	31
$\hat{d}_j^{(1)}$	3	20	14	1	12	3	3	16	6	1	6	8	7	19	20	90	6	4	4	13	

owing to full exploitation of the network structure of C2SDM, our four algorithms significantly excel LINGO in solving quality and efficiency.

However, the theoretical derivation does not always completely conform to the practice. This point can be verified by the comparison (see Tab. 27) on computation result and runtime for our four algorithms and generic solver LINGO to solve six specified instances. Items in Table 27 have explanation below. Item ZO (or ZL) denotes the duration of the optimum (or feasible) solution found for a specified instance by our four algorithms

TABLE 22. The second stage destination demand, and the route shipping capacity for 20×20 problem.

$\hat{u}_{ij}$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	31	5	9	12	12	61	22	12	4	56	68	42	65	79	42	6	51	27	23	55
2	33	31	17	71	13	33	49	44	59	10	26	1	63	74	24	19	77	16	34	30
3	26	79	47	43	11	72	3	64	58	20	76	38	58	32	19	49	45	37	52	62
4	6	13	9	5	44	77	19	28	72	39	19	68	6	11	78	67	66	80	20	54
5	51	78	34	5	71	14	20	49	30	43	74	44	77	62	33	61	46	36	4	4
6	30	34	30	51	15	35	13	78	22	5	65	57	21	64	70	79	22	38	74	74
7	41	71	46	34	66	59	72	13	62	54	74	67	63	63	38	60	46	14	25	38
8	72	70	31	78	41	79	36	63	25	7	71	47	69	65	62	74	5	66	19	22
9	5	73	80	21	44	28	18	17	11	49	47	69	8	17	51	37	23	67	25	78
10	32	64	65	49	29	47	53	47	5	68	5	13	26	35	34	69	5	72	20	5
11	63	3	11	54	22	68	17	19	41	50	49	9	76	31	28	4	72	40	33	75
12	21	45	4	66	38	44	31	12	59	18	17	26	34	60	10	35	66	46	62	24
13	6	50	41	56	29	51	13	73	30	66	14	40	3	60	68	24	34	42	1	54
14	79	8	63	4	78	80	56	56	15	14	78	56	71	11	13	22	3	26	73	20
15	57	67	7	66	66	17	9	50	7	50	37	76	16	24	49	20	26	8	1	77
16	59	35	37	37	69	33	57	31	31	19	19	6	64	66	27	34	80	56	40	67
17	28	13	73	41	64	50	28	77	30	25	18	62	72	10	73	50	72	61	27	44
18	66	13	64	35	44	6	5	67	79	2	53	77	10	31	40	73	14	67	53	18
19	4	73	40	54	6	2	77	17	72	36	14	24	61	69	21	28	70	69	34	48
20	16	76	21	2	26	49	31	52	60	57	15	5	20	16	79	15	53	73	22	21
$\hat{d}_j^{(2)}$	40	30	20	10	30	40	20	30	20	20	20	10	20	10	10	20	20	10	40	30

(or generic solver LINGO). Items t-A1, t-A2, t-A1bs, t-A2bs and t-L indicate the runtime (in second) for C2SD-A1, C2SD-A2, C2SD-A1bs, C2SD-A2bs and LINGO respectively to solve a specified instance. Items A2/A1, A2bs/A1bs, A1bs/A1, and A2bs/A2 indicate the percentage ratio of the runtime for solving a specified instance by C2SD-A2 to that by C2SD-A1, that by C2SD-A2bs to that by C2SD-A1bs, that by C2SD-A1bs to that by C2SD-A1, and that by C2SD-A2bs to that by C2SD-A2, respectively. As is worth mentioning, the best performing in our four algorithms is indicated in bold runtime in Table 27, while the entries with percentage ratios A2/A1, A2bs/A1bs, A1bs/A1 and A2bs/A2 more than one are indicated in bold blue in Table 27.

The comparison given in Table 27 tentatively confirms the accuracy of our theoretic derivations, except that C2SD-A2bs ranks best and C2SD-A2 next in computation time. Table 27 clearly reveals that our four algorithms are excellent exact solution technique for C2SD, and in computation time, C2SD-A2 ranks best, C2SD-A2bs next, C2SD-A1bs third, and C2SD-A1 fourth. Also, in computation time, the bigger the size of instance solved, the bigger the advantage of C2SD-A2 over C2SD-A1, C2SD-A2bs over C2SD-A1bs, C2SD-A1bs over C2SD-A1, and C2SD-A2 over C2SD-A2bs. In particular, it follows from Table 27 that all of our four algorithms greatly excel generic solver LINGO in solving quality and rival LINGO in computation time when applied to solving an instance of C2SD. The main reasons given below for incurring the above performance difference of our four algorithms and LINGO are believable. First, all of our four algorithms fully exploit the network structure of C2SDM by constructing network and calling feasible flow algorithm, while the network structure of C2SDM isn't fully exploited by generic solver LINGO with no special design for C2SDM's network structure characteristics. Second, C2SD-A2 and C2SD-A2bs have fast descent characteristics due to adoption of greedy search. Third, C2SD-A1bs and C2SD-A2bs apply binary search to expedite the C2SDM model solving process of C2SD-A1 and C2SD-A2 respectively.

As is worth mentioning, C2SD-A2bs is inferior to C2SD-A2 in computation time, *i.e.*, application of binary search to C2SD-A2 does not play the computation time reduction role. Why does such phenomenon appear?

Table 23. The first stage shipment plan with duration as 105 corresponding to the optimum solution found for 20×20 problem by our four algorithms.

$\hat{x}_{ij}^{(1)}$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	$\hat{s}_i$
1						3	3					8									31
2																3	2		1		36
3			4										2							3	22
4				1						1											32
5																		4			30
6			2		12											6					26
7									6												33
8																			3		33
9											6				10					10	26
10		16															4				26
11													5	5	10						37
12														14							37
13																					32
14			8																		40
15																					39
16																					25
17	3																				34
18																					40
19		4																			25
20								16													31
$\hat{d}_j^{(1)}$	3	20	14	1	12	3	3	16	6	1	6	8	7	19	20	9	6	4	4	13	

Table 24. The second stage shipment plan with duration as 105 corresponding to the optimum solution found for 20×20 problem by our four algorithms.

$\hat{x}_{ij}^{(2)}$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	$\hat{s}_i$
1							1					10								5	31
2					9												5		16		36
3					1								3						4	5	22
4					10										10				10		32
5																	15	10			30
6																6					26
7						7			20												33
8											20								10		33
9																					26
10							6														26
11													17								37
12						13								10							37
13		10																		20	32
14			20					10													40
15						16		13		10											39
16					10											14					25
17							13	7		10											34
18	26			10		4															40
19		20																			25
20	14																				31
$\hat{d}_j^{(2)}$	40	30	20	10	30	40	20	30	20	20	20	10	20	10	10	20	20	10	40	30	

TABLE 25. The first stage shipment plan with duration as 382 corresponding to the feasible solution found for 20×20 problem by LINGO.

$\hat{x}_{ij}^{(1)}$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	$\hat{s}_i$
1																					31
2																			4		36
3			11			3		2	6												22
4	2				5		3	9			4				2	2	2				32
5		5										5	3								30
6					1							3	2								26
7								1			2					4		4			33
8													2				4			10	33
9			3		6			2													26
10														7							26
11																					37
12		5													4						37
13								2						4		3					32
14		1																			40
15				1						1				3							39
16		2													5						25
17	1													1	4					3	34
18		7												4	5						40
19																					25
20																					31
$\hat{d}_j^{(1)}$	3	20	14	1	12	3	3	16	6	1	6	8	7	19	20	9	6	4	4	13	

TABLE 26. The second stage shipment plan with duration as 397 corresponding to the feasible solution found for 20×20 problem by LINGO.

$\hat{x}_{ij}^{(2)}$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	$\hat{s}_i$
1						11													18		31
2		6						9												17	36
3																					22
4																	3				32
5			3							4	4	6									30
6				4			12						4								26
7					3	3			1	4		4			6			1			33
8			3	3						3									7	1	33
9				3		2	3						3								26
10					2		1				4		3	9							26
11		3	4					3	19	1					1	1			4		37
12	4				4	4	3				4		4	1	3			1			37
13	6	5			4	4				3									1		32
14	21	3				10							4						1		40
15	3	1			3		1	3		3	4					7	1	8			39
16	1		5							2			2			4	4				25
17	3	4			4	3		4								4				3	34
18	2	4	5		2	1		2			4								2		40
19					4	2		6								4				9	25
20		4			4			3									12		7		31
$\hat{d}_j^{(2)}$	40	30	20	10	30	40	20	30	20	20	20	10	20	10	10	20	20	10	40	30	

TABLE 27. Comparison on computation result and runtime (in second) for our four algorithms and generic solver LINGO to solve each of six specified instances.

Instance name	ZO	ZL	t-A1	t-A2	t-A1bs	t-A2bs	t-L	A2/A1	A2bs/A1bs	A1bs/A1	A2bs/A2
6×4 problem 1	13	24	0.032034	**0.018919**	0.023323	0.022833	0.70	59.06%	97.90%	72.81%	**120.69%**
6×4 problem 2	319	685	0.03819	**0.020414**	0.035851	0.02673	0.75	53.45%	74.56%	93.88%	**130.94%**
10×5 problem 1	10	49	0.115766	**0.038508**	0.073076	0.051012	3.68	33.26%	69.81%	63.12%	**132.47%**
10×5 problem 2	151	240	0.228574	**0.096516**	0.164895	0.112312	4.11	42.23%	68.11%	72.14%	**116.37%**
20×10 problem	149	280	71.46616	**9.397827**	26.22227	13.25411	42.25	13.15%	50.55%	36.69%	**141.03%**
20×20 problem	210	779	460.6666	**14.26211**	45.03909	36.44326	39.66	3.10%	80.91%	9.78%	**255.53%**

TABLE 28. The parameter interval for a randomly produced instance of distinct size.

Interval		Parameter				
		$\hat{s}_i$	$\hat{d}_j^{(1)}$	$\hat{d}_j^{(2)}$	$\hat{t}_{ij}$	$\hat{u}_{ij}$
Size ($\tilde{m} \times \tilde{n}$)	6×4	$[21, 40]$	$[1, 20]$	$[10, 30]$	$[1, 400]$	$[1, 80]$
	10×5	$[21, 40]$	$[1, 20]$	$[10, 30]$	$[1, 400]$	$[1, 80]$
	10×10	$[21, 40]$	$[1, 20]$	$[10, 20]$	$[1, 400]$	$[1, 80]$
	20×10	$[21, 40]$	$[1, 20]$	$[10, 30]$	$[1, 400]$	$[1, 80]$
	20×20	$[21, 40]$	$[1, 20]$	$[10, 20]$	$[1, 400]$	$[1, 80]$
	30×30	$[31, 60]$	$[1, 30]$	$[10, 30]$	$[1, 900]$	$[1, 120]$

Believably, this is because C2SD-A2 has fast descent characteristics, and application of binary search to C2SD-A2 also has computation time consumption not counteracted by possible time reduction derived from binary search just like the saying "going too far is as bad as not going far enough".

Also, the practice is the unique criterion to test the truth. For this, the practical performance of our four algorithms is further validated in the following by abundant computation experiments.

5. Computation experiments

For further validation of the practical performance of our four algorithm (C2SD-A1, C2SD-A2, C2SD-A1bs, C2SD-A2bs), total sixty randomly produced C2SD instances of distinct size ($\tilde{m} \times \tilde{n}$) are solved in this section similarly to 6×4 problem 1 in Section 3.1 by applying our four algorithms and generic solver LINGO. The sixty instances are composed of ten 6×4, 10×5, 10×10, 20×10, 20×20 and 30×30 instances, for each of which known parameters $\hat{s}_i$, $\hat{d}_j^{(1)}$, $\hat{d}_j^{(2)}$, $\hat{u}_{ij}$ and $\hat{t}_{ij}$ are integer taken from a uniform distribution on interval given in Table 28 at random.

As a consequence of experiments, a comparison on computation result and runtime for solving any of the sixty randomly produced instances composed of ten instances with specified name and size as 6×4, 10×5, 10×10, 20×10, 20×20, and 30×30 by our four algorithms and generic solver LINGO, is given in Table 29. Explanations for items in Table 29 are presented below. Item ZL denotes the duration of a feasible solution found for an instance of given size by LINGO, and item t_L denotes the runtime (in second) for finding a feasible solution to a given size instance by LINGO. Items Z1, Z2, Z1bs and Z2bs denote the duration of the optimum solution found for an instance of given size by C2SD-A1, C2SD-A2, C2SD-A1bs and C2SD-A2bs respectively. Items t_A1, t_A2, t_A1bs and t_A2bs represent the runtime (in second) for finding the optimum solution to a given size instance by C2SD-A1, C2SD-A2, C2SD-A1bs and C2SD-A2bs respectively. Also, the one with performance best in our four algorithms is represented in bold runtime in Table 29, where all the known parameters for each instance are not given but provided if required.

Table 29. Comparison on computation result and runtime (in second) for solving any of randomly produced sixty instances by our four algorithms and generic solver LINGO.

Size ($\tilde{m} \times \tilde{n}$)	Instance name	Z1	Z2	Z1bs	Z2bs	ZL	t_A1	t_A2	t_A1bs	t_A2bs	t_L
	Data_C2SD_6 × 4_1	245	245	245	245	746	0.030117	**0.019536**	0.024090	0.022118	0.53
	Data_C2SD_6x4_2	300	300	300	300	718	0.015269	**0.010645**	0.013459	0.013415	0.42
	Data_C2SD_6x4_3	339	339	339	339	780	0.020574	**0.009307**	0.015125	0.010715	0.35
	Data_C2SD_6x4_4	315	315	315	315	780	0.018074	**0.010477**	0.016772	0.013275	0.42
6×4	Data_C2SD_6x4_5	342	342	342	342	790	0.033009	**0.012880**	0.020817	0.018642	0.42
	Data_C2SD_6x4_6	223	223	223	223	766	0.029300	**0.013004**	0.022674	0.018189	0.46
	Data_C2SD_6x4_7	384	384	384	384	754	0.016029	**0.009078**	0.015855	0.015104	0.51
	Data_C2SD_6x4_8	360	360	360	360	796	0.029272	**0.010895**	0.023819	0.019538	0.45
	Data_C2SD_6x4_9	240	240	240	240	744	0.013893	**0.007033**	0.011478	0.007969	0.67
	Data_C2SD_6x4_10	136	136	136	136	676	0.013505	**0.006101**	0.011357	0.008068	0.35
	Data_C2SD_10 × 5_1	206	206	206	206	782	0.083917	**0.043784**	0.076233	0.066447	0.51
	Data_C2SD_10x5_2	198	198	198	198	788	0.073708	**0.047859**	0.079906	0.073342	0.65
	Data_C2SD_10x5_3	180	180	180	180	796	0.050241	**0.029961**	0.047627	0.040209	0.55
	Data_C2SD_10x5_4	207	207	207	207	800	0.068176	**0.040595**	0.061498	0.052361	0.66
10×5	Data_C2SD_10x5_5	142	142	142	142	780	0.141690	**0.060319**	0.095260	0.084171	0.52
	Data_C2SD_10x5_6	300	300	300	300	792	0.188978	**0.056822**	0.103852	0.092292	0.66
	Data_C2SD_10x5_7	167	167	167	167	792	0.176187	**0.055200**	0.111285	0.086520	0.89
	Data_C2SD_10x5_8	167	167	167	167	743	0.103765	**0.047655**	0.074326	0.068178	1.27
	Data_C2SD_10x5_9	140	140	140	140	794	0.087902	**0.044809**	0.080490	0.071945	0.72
	Data_C2SD_10x5_10	272	272	272	272	794	0.158602	**0.072757**	0.144700	0.073142	0.71
	Data_C2SD_10x10_1	300	300	300	300	794	1.657762	**0.402516**	0.710430	0.604220	2.38
	Data_C2SD_10x10_2	331	331	331	331	790	2.301369	**0.444727**	0.757950	0.500470	2.59
	Data_C2SD_10x10_3	248	248	248	248	797	1.278262	**0.399909**	0.765460	0.658674	2.95
	Data_C2SD_10x10_4	158	158	158	158	800	2.488263	**0.406582**	0.939522	0.557949	2.03
10×10	Data_C2SD_10x10_5	306	306	306	306	798	1.294363	**0.348400**	0.550247	0.531800	1.82
	Data_C2SD_10x10_6	305	305	305	305	792	3.291840	**0.510145**	1.125019	0.696613	2.63
	Data_C2SD_10x10_7	338	338	338	338	796	1.811436	**0.351413**	0.776370	0.715481	2.04
	Data_C2SD_10x10_8	216	216	216	216	768	2.923634	**0.517502**	1.039841	0.825305	5.86
	Data_C2SD_10x10_9	356	356	356	356	794	0.610541	**0.236482**	0.553499	0.538444	1.40
	Data_C2SD_10x10_10	248	248	248	248	798	2.381878	**0.487420**	0.977888	0.736628	2.26
	Data_C2SD_20x10_1	182	182	182	182	798	10.285116	**2.054059**	4.597100	2.612543	2.70
	Data_C2SD_20x10_2	185	185	185	185	788	4.037503	**1.158703**	2.447689	1.974728	4.17
	Data_C2SD_20x10_3	164	164	164	164	793	6.401986	**1.514595**	2.583678	2.075967	2.60
	Data_C2SD_20x10_4	178	178	178	178	798	7.007786	**2.099220**	3.858150	2.974978	3.12
20×10	Data_C2SD_20x10_5	137	137	137	137	796	7.133177	**1.437154**	3.394423	2.334266	4.57
	Data_C2SD_20x10_6	172	172	172	172	788	5.106498	**1.037280**	2.596766	2.118321	1.91
	Data_C2SD_20x10_7	237	237	237	237	792	4.133355	**0.887119**	2.042052	1.817932	2.23
	Data_C2SD_20x10_8	152	152	152	152	800	6.858834	**1.074613**	2.475990	1.990825	2.79
	Data_C2SD_20x10_9	162	162	162	162	786	8.533525	**1.360159**	3.458472	2.835161	4.09
	Data_C2SD_20x10_10	130	130	130	130	794	8.746146	**1.482042**	3.715760	2.677717	4.46
	Data_C2SD _20x20_1	170	170	170	170	796	453.581678	**22.667051**	68.704729	44.768942	24.34
	Data_C2SD_20x20_2	140	140	140	140	796	527.804368	**48.085788**	78.980147	49.252845	21.35
	Data_C2SD_20x20_3	156	156	156	156	793	527.964476	**24.124798**	71.788833	51.350597	14.71
	Data_C2SD_20x20_4	197	197	197	197	786	435.962742	**24.222059**	65.447244	45.815448	79.88
20×20	Data_C2SD_20x20_5	148	148	148	148	794	293.373495	**18.809960**	54.295659	35.717468	50.03
	Data_C2SD_20x20_6	198	198	198	198	769	223.833140	**20.340004**	49.322102	29.760562	37.34
	Data_C2SD_20x20_7	159	159	159	159	787	586.399501	**20.293146**	62.392006	46.552720	25.94
	Data_C2SD_20x20_8	204	204	204	204	795	691.706036	**21.732497**	59.647263	44.116433	20.13
	Data_C2SD_20x20_9	179	179	179	179	796	302.754050	**21.134841**	53.144711	37.776092	75.72
	Data_C2SD_20x20_10	132	132	132	132	788	254.238883	**30.710952**	98.805502	78.383210	11.70
	Data_C2SD_30x30_1	394	394	394	394	1796	5210.838574	**445.80231**	708.477441	662.251597	75.55
	Data_C2SD_30x30_2	356	356	356	356	1790	5342.999618	**231.495027**	747.290381	555.35303	2249.77
	Data_C2SD_30x30_3	213	213	213	213	1799	12885.1674	**319.47058**	1230.267443	756.639025	125.09
	Data_C2SD_30x30_4	228	228	228	228	1778	6220.812404	**284.892491**	953.02014	693.247276	156.72
30×30	Data_C2SD_30x30_5	256	256	256	256	1793	10583.258398	**353.036189**	1151.54023	539.782931	96.42
	Data_C2SD_30x30_6	239	239	239	239	1786	8135.286779	**338.933234**	1053.313899	803.975089	2154.31
	Data_C2SD_30x30_7	300	300	300	300	1798	8161.758903	**216.344764**	662.64802	492.631117	63.06
	Data_C2SD_30x30_8	185	185	185	185	1794	15852.03112	**543.752754**	1636.871422	1029.736061	452.32
	Data_C2SD_30x30_9	323	323	323	323	1789	9004.831168	**281.384263**	1028.394158	773.902942	2265.77
	Data_C2SD_30x30_10	440	440	440	440	1787	3690.674013	**209.560024**	615.394461	508.729612	681.01

TABLE 30. Comparison on average runtime (in second) for solving an instance of given size by our four algorithms and generic solver LINGO.

Size ($\tilde{m} \times \tilde{n}$)	avt_A1	avt_A2	avt_A1bs	avt_A2bs	avt_L	A2/A1	A2bs/A1bs	A1bs/A1	A2bs/A2	A2/L
6×4	0.021904	**0.010896**	0.017545	0.014703	0.458	49.74%	83.80%	80.10%	**134.94%**	2.38%
10×5	0.113317	**0.049976**	0.087518	0.070861	0.714	44.10%	80.97%	77.23%	**141.79%**	7.00%
10×10	2.003935	**0.41051**	0.819623	0.636558	2.596	20.49%	77.66%	40.90%	**155.07%**	15.81%
20×10	6.824393	**1.410494**	3.117008	2.341244	3.264	20.67%	75.11%	45.67%	**165.99%**	43.21%
20×20	429.7618	**25.21211**	66.25282	46.34943	36.114	5.87%	69.96%	15.42%	**183.84%**	69.81%
30×30	8508.766	**322.4672**	978.7218	681.6249	832.002	3.79%	69.64%	11.50%	**211.39%**	38.76%

It is easily observed from Table 29 that all of our four algorithms efficiently seek out the optimum solution for any of the sixty instances produced at random, but generic solver LINGO always finds only a feasible solution far from the optimal for any of that in an acceptable computation time. Also, the runtime for solving an instance with size as 6×4, 10×5, 10×10, 20×10, 20×20 and 30×30 by C2SD-A1 (or C2SD-A2 or C2SD-A1bs or C2SD-A2bs or LINGO) is at most 0.033009 (or 0.019536 or 0.024090 or 0.022118 or 0.67), 0.188978 (or 0.072757 or 0.144700 or 0.092292 or 1.27), 3.291840 (or 0.517502 or 1.125019 or 0.825305 or 5.86), 10.285116 (or 2.099220 or 4.597100 or 2.974978 or 4.57), 691.706036 (or 48.085788 or 98.805502 or 78.383210 or 79.88) and 15852.03112 (or 543.752754 or 1636.871422 or 1029.736061 or 2265.77) s, respectively. Clearly, in computation time, C2SD-A2 significantly excels C2SD-A2bs, C2SD-A2bs vastly outdoes C2SD-A1bs, C2SD-A1bs has overwhelming preponderance over C2SD-A1, and C2SD-A1 rivals generic solver LINGO. Also, in solving quality, our four algorithms with ability of finding the optimal for C2SD perform the same, and overwhelmingly outperform LINGO. Totally, it indicates that for an instance of any size, C2SD-A2 does best, C2SD-A2bs next, C2SD-A1bs third, C2SD-A1 fourth, and LINGO worst.

From Table 29, a comparison (see Tab. 30) on average runtime for solving an instance of size 6×4, 10×5, 10×10, 20×10, 20×20 and 30×30 by our four algorithms and generic solver LINGO is derived. Interpretations for items in Table 30 are given as follows. Items avt_A1, avt_A2, avt_A1bs, avt_A2bs and avt_L denote the average runtime for solving an instance of given size by C2SD-A1, C2SD-A2, C2SD-A1bs, C2SD-A2bs and LINGO respectively. Items A2/A1, A2bs/A1bs, A1bs/A1, A2bs/A2, and A2/L denote the percentage ratio of average runtime for solving a given size instance by C2SD-A2 to that by C2SD-A1, that by C2SD-A2bs to that by C2SD-A1bs, that by C2SD-A1bs to that by C2SD-A1, that by C2SD-A2bs to that by C2SD-A2, and that by C2SD-A2 to that by LINGO respectively. Additionally, the one with performance best in our four algorithms is shown in bold average runtime in Table 30, and the entries for A2/A1, A2bs/A1bs, A1bs/A1, A2bs/A2 and A2/L to surpass one are shown in bold blue in Table 30.

From Table 30, a chart (see Fig. 3) of the performance trend comparison for solving an instance of size 6×4, 10×5, 10×10, 20×10, 20×20 and 30×30 by our four algorithms and generic solver LINGO is derived. In Figure 3, the vertical and horizontal axis represent the average runtime and the solved instance size respectively for solving an instance of given size by one of our four algorithms or generic solver LINGO. Also, from Table 30, a chart (see Fig. 4) of the trend for A2/A1, A2bs/A1bs, A1bs/A1, A2bs/A2 and A2/L varying with size is derived. In Figure 4, the horizontal axis represents the solved instance size, and the vertical axis represents the percentage value for A2/A1, A2bs/A1bs, A1bs/A1, A2bs/A2 and A2/L.

Tables 29 and 30 and Figures 3 and 4 firmly confirm that our four algorithms (C2SD-A1, C2SD-A2, C2SD-A1bs, C2SD-A2bs) are exact solution approach to C2SD with high efficiency and robustness, and successfully overcome the defect of generic solver LINGO with poor optimization effect. It is revealed that in computation time, C2SD-A1bs is vastly superior to C2SD-A1, but C2SD-A2bs is significantly inferior to C2SD-A2, and C2SD-A2 and C2SD-A2bs significantly excel C2SD-A1 and C2SD-A1bs respectively. Especially in computation time, the bigger the size of instance solved, the bigger the superiority for C2SD-A2 to C2SD-A1, C2SD-A2bs to C2SD-A1bs, C2SD-A1bs to C2SD-A1, and C2SD-A2 to C2SD-A2bs. Also, it is notable that C2SD-A2, as

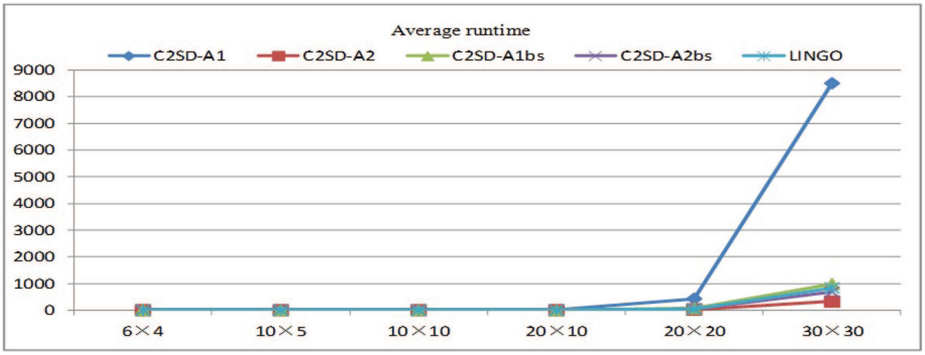

FIGURE 3. The chart of the performance trend comparison for solving an instance of C2SD by our four algorithms and generic solver LINGO.

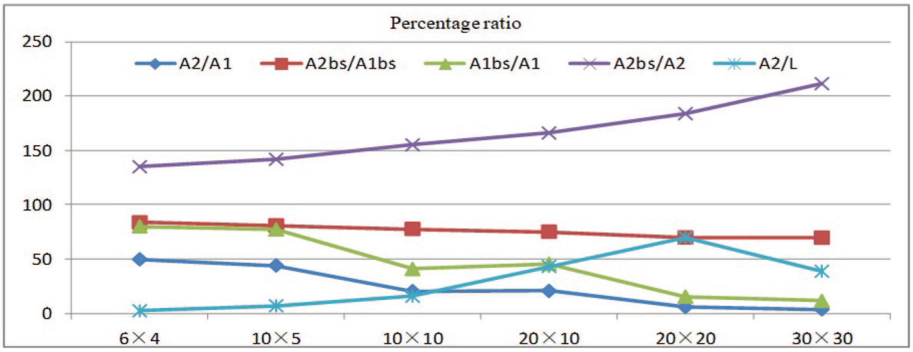

FIGURE 4. The chart of the trend for A2/A1, A2bs/A1bs, A1bs/A1, A2bs/A2 and A2/L varying with size.

the best performing in our four algorithms, overwhelmingly outperforms generic solver LINGO in both solving quality and computation time, especially for large size instances.

Consequently, our four algorithms C2SD-A1, C2SD-A2, C2SD-A1bs and C2SD-A2bs are robust and efficient exact solution technique for C2SD as compared to generic solver LINGO, where the most recommendable is C2SD-A2, next C2SD-A2bs, third C2SD-A1bs, last C2SD-A1, especially for large size instances.

6. CONCLUSIONS

Capacitated multistage time minimization transportation problem with destination-stage demand specified (CMTMTPSD) is an important optimization problem arising in industries due to resource finiteness and multistage shipment pervasiveness and time vital importance and destination-stage demand timely satisfaction necessity. CMTMTPSD aims at finding a multistage shipment plan with minimum duration to satisfy any stage demand at each destination subject to the restrictions on origin available supply and route shipping capacity, on condition that the identical stage shipment is parallel conducted, current stage shipment starts after previous stage shipment is completed. Due to the super intractability of CMTMTPSD, even CMTMTPSD's simplest but crucial case, the CMTMTPSD with two stages, *i.e.*, C2SD, has never been investigated in literature so far. C2SD is intimately related to but intrinsically different from C2TMTP investigated in Xie and Li [43] and Sharma *et al.* [31]. As C2TMTP's variant, C2SD is more conformable to the reality and more extendable as compared to C2TMTP. This paper has explored C2SD, and developed high efficiency C2SD solving technique.

Based on reduction of C2SD to a series of search for feasible flow in the constructed network by creating mathematical model C2SDM along with auxiliary models and calling Xie and Li [43]'s feasible flow algorithm FFn-A, four iterative algorithms, including two named C2SD-A1 and C2SD-A2, and another two named C2SD-A1bs and C2SD-A2bs derived respectively from application of binary search to C2SD-A1 and C2SD-A2, were proposed to resolve C2SD efficiently. It was theoretically proved that the optimum solution is found for C2SD by any of our four algorithms (C2SD-A1, C2SD-A2, C2SD-A1bs, C2SD-A2bs) in a polynomial time. Owing to full exploitation of the network structure of C2SD, our four algorithms hold the merits of high computational efficiency, no memory overflow, easy computer implementation, and easy extension to the CMTMTPSD with more than two stages. Distinct size examples were given as illustration on the application value and performance difference of our four algorithms as compared with generic solver LINGO, and analyses on causes for incurring their performance difference were presented.

Computation experiments on randomly produced instances of distinct size were conducted to further verify the practical performance of our four algorithms (C2SD-A1, C2SD-A2, C2SD-A1bs, C2SD-A2bs), as compared with generic solver LINGO applied to C2SDM model. It was indicated that our four algorithms are robust and efficient exact solution approach to C2SD with ability of serving as powerful tool to resolve other relevant complicated optimization problems. It was validated that in computation time, C2SD-A2 significantly excels C2SD-A2bs, C2SD-A2bs vastly outperforms C2SD-A1bs, and C2SD-A1bs has overwhelming preponderance over C2SD-A1. Also, it was validated that generally in computation time, the bigger the size of instance solved, the bigger the advantage of C2SD-A2 over C2SD-A2bs, C2SD-A2bs over C2SD-A1bs, C2SD-A1bs over C2SD-A1, and C2SD-A2 over C2SD-A1. In particular, all of our four algorithms overwhelmingly excel LINGO in solving quality and rival LINGO in computation time when applied to solving an instance of C2SD, successfully overcoming the defect of generic solver LINGO with poor optimization effect. Accordingly, as compared to generic solver LINGO, our four algorithms are exact solution technique with high efficiency and robustness for C2SD, where the most recommendable is C2SD-A2, next C2SD-A2bs, third C2SD-A1bs, and last C2SD-A1, especially for large size instances.

As prospect, interesting and challenging work in the future is as follows. The first is extension of our C2SD solving approach to solving the CMTMTPSD with more than two stages in high efficiency. The second is integration of our approach with heuristics including genetic algorithm for efficiently tackling other related more complex optimization problems. The third is application of our developed theoretical approach to resolving some practical application instances arising in industries.

Acknowledgments

We are in sincere appreciation of all the supports. We also thank Professor Ya-xiang Yuan, CAS, for his support in completion of this work. We sincerely thank the anonymous referees for their valuable suggestions which have led to substantial improvement in the presentation of this paper.

Funding

This work is supported by State Key Laboratory of Scientific and Engineering Computing, Chinese Academy of Sciences (CAS); Scientific Research Foundation for Talent Introduction in Sichuan University of Science and Engineering under Grant 2016RCL32; Natural Science Foundation of Jiangxi Province under Grant 20151BAB201022; Natural Science Foundation of Sichuan Province under Grant 2023NSFSC1021.

Data availability statement

The data that support the findings of this study are available on request from the corresponding author.

References

[1] R.K. Ahuja, Algorithm for the min–max transportation problem. *Nav. Res. Logistics Q.* **33** (1986) 725–739.

[2] R.K. Ahuja, T.L. Magnanti and J.B. Orlin, Networks Flows: Theory, Algorithms, and Applications. Prentice-Hall, Englewood Cliffs, NJ (1993).

[3] S. Arora and M.C. Puri, On lexicographic optimal solutions in transportation problem. *Optimization* **39** (1997) 383–403.

[4] A.N. Balaji, J.M. Nilakantan, I.E. Nielsen and S.G. Ponnambalam, Solving fixed charge transportation problem with truck load constraint using metaheuristics. *Ann. Oper. Res.* **273** (2019) 207–236.

[5] S. Bansal and M.C. Puri, A min–max problem. *ZOR* **24** (1980) 191–200.

[6] H.L. Bhatia, K. Swarup and M.C. Puri, A procedure for time minimizing trans-portation problem. *Indian J. Pure Appl. Math.* **8** (1977) 920–929.

[7] K. Dahiya and V. Verma, Capacitated transportation problem with bounds on rim conditions. *Eur. J. Oper. Res.* **178** (2007) 718–737.

[8] S.K. Das and S.K. Roy, Effect of variable carbon emission in a multi-objective transportation-p-facility location problem under neutrosophic environment. *Comput. Ind. Eng.* **132** (2019) 311–324.

[9] S.K. Das, S.K. Roy and G.W. Weber, Application of type-2 fuzzy logic to a multi-objective green solid transportation-location problem with dwell time under carbon tax, cap and offset policy: fuzzy vs. non-fuzzy techniques. *IEEE Trans. Fuzzy Syst.* **8** (2020) 2711–2725.

[10] H. Ding and F. Xie, A solution technique for capacitated two-level hierarchical time minimization transportation problem. *Comput. Oper. Res.* **151** (2023) 106125.

[11] E. Garajová and M. Rada, Complexity of computing the worst optimal value of interval transportation problems. *Cent. Eur. J. Oper. Res.* **33** (2025) 819–834.

[12] R.S. Garfinkel and M.R. Rao, The bottleneck transportation problem. *Nav. Res. Logistics Q.* **18** (1971) 465–472.

[13] R. Geng, S.A. Mansouri and E. Aktas, The relationship between green supply chain management and performance: a meta-analysis of empirical evidences in Asian emerging economies. *Int. J. Prod. Econ.* **183** (2017) 245–258.

[14] S. Ghosh, S.K. Roy, A. Ebrahimnejad and J.L. Verdegay, Multi-objective fully intuitionistic fuzzy fixed-charge solid transportation problem. *Complex Intell. Syst.* **7** (2021) 1009–1023.

[15] M. Gurupada, S.K. Roy and V.J. Luis, Time variant multi-objective interval-valued transportation problem in sustainable development. *Sustainability* **11** (2019) 6161.

[16] P.L. Hammer, Time minimization transportation problem. *Nav. Res. Logistics Q.* **18** (1969) 345–357.

[17] R. Hassain and E. Zemel, Probabilistic analysis of the capacitated transportation problem. *Math. Oper. Res.* **13** (1988) 80–89.

[18] H. Issermann, Linear bottleneck transportation problem. *Asia Pac. J. Oper. Res.* **1** (1984) 38–52.

[19] E. Jain, K. Dahiya, A. Sharma and V. Verma, An improved algorithm for two stage time minimization assignment problem. *J. Comb. Opt.* **37** (2019) 713–736.

[20] E. Jain, K. Dahiya and V. Verma, Three-phase time minimization transportation problem. *Eng. Opt.* **53** (2021) 461–473.

[21] P. Kaur, A. Sharma, V. Verma and K. Dahiya, A priority based assignment problem. *Appl. Math. Modell.* **40** (2016) 7784–7795.

[22] P. Kaur, V. Verma and K. Dahiya, Capacitated two-stage time minimization transportation problem with restricted flow. *RAIRO Oper. Res.* **51** (2017) 1169–1184.

[23] P. Kaur, A. Sharma, V. Verma and K. Dahiya, An alternate approach to solve two-level hierarchical time minimization transportation problem. **4OR** (2022) 23–61.

[24] S. Kaur, E. Jain, A. Sharma and K. Dahiya, An efficient algorithm for two-stage capacitated time minimization transportation problem with restricted flow. *RAIRO Oper. Res.* **58** (2024) 2733–2766.

[25] J.B. Mazzola and A.W. Neebe, An algorithm for the bottleneck generalized assignment problem. *Comput. Oper. Res.* **20** (1993) 355–362.

[26] S. Midya, S.K. Roy and V.F. Yu, Intuitionistic fuzzy multi-stage multi-objective fixed-charge solid transportation problem in a green supply chain. *Int. J. Mach. Learn. Cybern.* **12** (2021) 699–717.

[27] S. Parkash, On minimizing the duration of transportation. *Proc. Indian Acad. Sci.* **91** (1982) 53–57.

[28] S.T. Rachev and I. Olkin, Mass transportation problems with capacity constraints. *J. Appl. Prob.* **36** (1999) 433–445.

[29] ResearchGate, https://www.researchgate.net/publication/390740067/ (2025).

[30] V. Sharma, K. Dahiya and V. Verma, A note on two stage interval time minimization transportation problem. *ASOR Bull.* **27** (2008) 12–18.

[31] V. Sharma, K. Dahiya and V. Verma, Capacitated two-stage time minimization transportation problem. *Asia Pac. J. Oper. Res.* **27** (2010) 457–476.

[32] A. Sharma, V. Verma, P. Kaur and K. Dahiya, An iterative algorithm for two level hierarchical time minimization transportation problem. *Eur. J. Oper. Res.* **246** (2015) 700–707.

[33] H.D. Sherali, Equivalent weights for lexicographic multi objective programs characterization and computation. *Eur. J. Oper. Res.* **11** (1982) 367–379.

[34] G. Singh and A. Singh, Extension of Particle Swarm Optimization algorithm for solving two-level time minimization transportation problem. *Math. Comput. Simul.* **204** (2023) 727–742.

[35] Sonia and R. Malhotra, A polynomial algorithm for a two stage time minimization transportation problem. *Opsearch* **39** (2002) 251–266.

[36] Sonia and M.C. Puri, Two level hierarchical time minimizing transportation problem. *TOP* **12** (2004) 301–330.

[37] Sonia and M.C. Puri, Max–min sum minimization transportation problem. *Ann. Oper. Res.* **143** (2006) 265–275.

[38] Sonia and M.C. Puri, Two-stage time minimizing assignment problem. *Omega* **36** (2008) 730–740.

[39] Sonia, R. Malhotra and M.C. Puri, Two stage interval time minimization transportation problem. *Bull. Aust. Soc. Oper. Res.* **23** (2004) 2–14.

[40] F. Xie and R. Jia, On capacitated transportation problem with bounds on rim conditions, in 2009 International Conference on New Trends in Information and Service Science (NISS2009). IEEE Computer Society Press, Beijing (2009) 131–138.

[41] F. Xie and R. Jia, Nonlinear fixed charge transportation problem by minimum cost flow-based genetic algorithm. *Comput. Ind. Eng.* **63** (2012) 763–778.

[42] F. Xie and Z. Li, An iterative solution technique to minimize the average transportation cost of capacitated transportation problem with bounds on rim conditions. *Asia-Pac. J. Oper. Res.* **37** (2020) 2050024.

[43] F. Xie and Z. Li, An iterative solution technique for capacitated two-stage time minimization transportation problem. *4OR-A Q. J. Oper. Res.* **20** (2022) 637–684.

[44] F. Xie, M.M. Butt, Z. Li and L. Zhu, An upper bound on the minimal total cost of the transportation problem with varying demands and supplies. *Omega* **68** (2017) 105–118.

[45] F. Xie, M.M. Butt and Z. Li, A feasible flow-based iterative algorithm for the two-level hierarchical time minimization transportation problem. *Comput. Oper. Res.* **86** (2017) 124–139.

[46] F. Xie, A. Sharma and Z. Li, An alternate approach to solve two-level priority based assignment problem. *Comput. Opt. App.* **81** (2022) 613–656.

[47] W. Zhang and F. Xie, An efficient solution approach to capacitated three-level hierarchical time minimization transportation problem. *Expert Syst. App.* **249** (2024) 123405.

[48] N. Zhang and F. Xie, An efficient solution approach to capacitated bilevel time minimizing transportation problem. *Discrete Opt.* **58** (2025) 100915.

[49] Z. Zou and F. Xie, A solution approach to minimal average cost flow problem with lower and upper bounds on flows. *Expert Syst. App.* **187** (2022) 115988.

Appendix A. Usable notations

Explanation table for usable notations

Notation	Explanation
C2SDM	The mathematical model systematically formulating C2SD, *i.e.*, main model
$\{\text{AMSD}(h) \mid h = 1,2,3,4\}$	The four auxiliary models of C2SDM model, in which the former two AMSD(1) and AMSD(2) do not have controllable parameter, but the latter two AMSD(3) and AMSD(4) have controllable parameter.
$\{\text{CUAM}(g) \mid g = 1,2\}$	Two commonly used auxiliary models CUAM(1) and CUAM(2) respectively usable directly for solving C2SD's former one auxiliary model AMSD(1) and latter three auxiliary models $\{\text{AMSD}(h) \mid h = 2,3,4\}$ treatable as their special cases
h	Index of the four auxiliary models of C2SDM model, $h \in \{1,2,3,4\}$
AMSD(h)	The h-th auxiliary model of C2SDM model, where $h \in \{1,2,3,4\}$
g	Index of two commonly used auxiliary models, or index of two stages, $g \in \{1,2\}$
CUAM(g)	The g-th commonly used auxiliary model, where $g \in \{1,2\}$
$\tilde{m}$	The number of all the origins, *i.e.*, origin number
$\tilde{n}$	The number of all the destinations, *i.e.*, destination number
$I = \{1,2,\ldots,\tilde{m}\}$	The set of all the origins, *i.e.*, origin set
$J = \{1,2,\ldots,\tilde{n}\}$	The set of all the destinations, *i.e.*, destination set
$I \times J = \{(i,j) \mid i \in I, j \in J\}$	The set of all the routes from origin $i \in I$ to destination $j \in J$, *i.e.*, route set
(i,j)	The route from origin $i \in I$ to destination $j \in J$, $(i,j) \in I \times J$
r or i	Origin index, $r \in I$ or $i \in I$
k or j	Destination index, $k \in J$ or $j \in J$
$\hat{s}_r$ or $\hat{s}_i$	The available supply at origin $r \in I$ or $i \in I$
$\hat{d}_k^{(1)}$ or $\hat{d}_j^{(1)}$	The first stage demand at destination $k \in J$ or $j \in J$
$\hat{d}_k^{(2)}$ or $\hat{d}_j^{(2)}$	The second stage demand at destination $k \in J$ or $j \in J$
$\hat{t}_{ij}$	The shipping time of route $(i,j) \in I \times J$
$\hat{u}_{ij}$	The shipping capacity of route $(i,j) \in I \times J$
$u_{ij}^{(1)}$	The first stage route capacity controllable parameter for CUAM(1) or CUAM(2), *i.e.*, the parameter for controlling the first stage shipping capacity of route $(i,j) \in I \times J$ in CUAM(1) or CUAM(2)
$u_{ij}^{(2)}$	The second stage route capacity controllable parameter for CUAM(1) or CUAM(2), *i.e.*, the parameter for controlling the second stage shipping capacity of route $(i,j) \in I \times J$ in CUAM(1) or CUAM(2)
$x_{ij}^{(1)}$ or $x_{ij}^{(b1)}$ or $x_{ij}^{(\triangle 1)}$ or $x_{ij}^{(\delta 1)}$	The first stage shipment volume of route $(i,j) \in I \times J$
$x_{ij}^{(2)}$ or $x_{ij}^{(b2)}$ or $x_{ij}^{(\triangle 2)}$ or $x_{ij}^{(\delta 2)}$	The second stage shipment volume of route $(i,j) \in I \times J$
$x_{ij}^{(*1)}$ or $x_{ij}^{(o1)}$	The first stage optimum shipment volume of route $(i,j) \in I \times J$
$x_{ij}^{(*2)}$ or $x_{ij}^{(o2)}$	The second stage optimum shipment volume of route $(i,j) \in I \times J$
Tu	The upper bound on the first stage shipping times of some feasible solutions to C2SDM
$z^{(b)}$	The given upper bound on the first or second stage shipping time of a feasible solution to C2SDM
$z_1; z^{(1)}$	The first stage shipping time of a feasible solution to C2SDM
$z_2; z^{(2)}$	The second stage shipping time of a feasible solution to C2SDM
$x_{ij}^{(g)}$	The g-th stage shipment volume of route $(i,j) \in I \times J$, where $g \in \{1,2\}$
$z^{(g)}$	The g-th stage shipment duration, where $g \in \{1,2\}$
$\hat{s} = (\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_{\tilde{m}})$	The origin supply vector
$\hat{d}^{(1)} = (\hat{d}_1^{(1)}, \hat{d}_2^{(1)}, \ldots, \hat{d}_{\tilde{n}}^{(1)})$	The first stage destination demand vector
$\hat{d}^{(2)} = (\hat{d}_1^{(2)}, \hat{d}_2^{(2)}, \ldots, \hat{d}_{\tilde{n}}^{(2)})$	The second stage destination demand vector

Notation	Explanation
$\hat{T} = (\hat{t}_{ij})_{\tilde{m}\times\tilde{n}} = \{\hat{t}_{ij} \mid (i,j) \in I \times J\}$	The route shipping time matrix
$\hat{u} = (\hat{u}_{ij})_{\tilde{m}\times\tilde{n}} = \{\hat{u}_{ij} \mid (i,j) \in I \times J\}$	The route shipping capacity matrix
$U^{(1)} = (u_{ij}^{(1)})_{\tilde{m}\times\tilde{n}}$	The first stage route shipping capacity controllable matrix for CUAM(1) or CUAM(2)
$U^{(2)} = (u_{ij}^{(2)})_{\tilde{m}\times\tilde{n}}$	The second stage route shipping capacity controllable matrix for CUAM(1) or CUAM(2)
$x^{(1)} = (x_{ij}^{(1)})_{\tilde{m}\times\tilde{n}} = \{x_{ij}^{(1)} \mid (i,j) \in I \times J\}$; $x^{(b1)}$; $x^{(\Delta 1)}$; $x^{(\delta 1)}$	The first stage route shipment volume matrix, *i.e.*, the first stage shipment plan
$x^{(2)} = (x_{ij}^{(2)})_{\tilde{m}\times\tilde{n}} = \{x_{ij}^{(2)} \mid (i,j) \in I \times J\}$; $x^{(b2)}$; $x^{(\Delta 2)}$; $x^{(\delta 2)}$	The second stage route shipment volume matrix, *i.e.*, the second stage shipment plan
$x^{(*1)} = (x_{ij}^{(*1)})_{\tilde{m}\times\tilde{n}} = \{x_{ij}^{(*1)} \mid (i,j) \in I \times J\}$; $x^{(o1)}$	The first stage optimum route shipment volume matrix, *i.e.*, the first stage optimum shipment plan
$x^{(*2)} = (x_{ij}^{(*2)})_{\tilde{m}\times\tilde{n}} = \{x_{ij}^{(*2)} \mid (i,j) \in I \times J\}$; $x^{(o2)}$	The second stage optimum route shipment volume matrix, *i.e.*, the second stage optimum shipment plan
ω	The number of distinct values in all the route shipping times, *i.e.*, the number of distinct values in $\{\hat{t}_{ij} \mid (i,j) \in I \times J\}$
$\alpha(1), \alpha(2), \ldots, \alpha(\omega)$	Rigorously descent time sequence, denoted as $\{a(p)\}$, *i.e.*, the rigorously descent sequence produced by sorting all the elements in $\{\hat{t}_{ij} \mid (i,j) \in I \times J\}$ and treating equal ones as the identical
$\Omega = \{1, 2, \ldots, \omega\}$	The set of the rigorously descent time sequence index values
p or q	Rigorously descent time sequence index, $p \in \Omega$ or $q \in \Omega$
$\alpha(p)$ or $\alpha(q)$	The p-th or q-th maximal time in rigorously descent time sequence
$[q_1, q_2]$	Binary search interval, *i.e.*, the binary search interval of rigorously descent time sequence
q_1	Binary search interval left endpoint, *i.e.*, the left endpoint of binary search interval
q_2	Binary search interval right endpoint, *i.e.*, the right endpoint of binary search interval
q_{2s}	Recently previous binary search interval right endpoint, *i.e.*, the right endpoint of recently previous binary search interval
$T_1^p(k)$; $T_1(k)$	The first stage shipping time of the k-th pair in a finite sequence of the first and second stage shipping time pair corresponding to a feasible solution to C2SDM
$T_2^p(k)$; $T_2(k)$	The second stage shipping time of the k-th pair in a finite sequence of the first and second stage shipping time pair corresponding to a feasible solution to C2SDM
$T_{1\min}$ or $T_{2\min}$	The first or second stage minimum feasible shipping time for C2SDM.

Appendix B. Proofs of Lemmas 1–3 and Theorems 1 and 2 as well as LINGO program of C2SD instance

They can be found on the link `https://www.researchgate.net/publication/390740067`.