# Real-Time Decentralized Framework for Technology-Enabled Intermodal Freight Transport

**Final Report**

by

**Nathan Huynh**, Ph.D., University of South Carolina
Telephone: 803-777-8947, Fax: 803-777-0670
Email: huynhn@cec.sc.edu

**William Ferrel,** Ph.D.
**Bhavya Padmanabhan**

**October 2023**

**Center for Connected Multimodal Mobility (C²M²)**

*200 Lowry Hall, Clemson University*
*Clemson, SC 29634*

# DISCLAIMER

# ACKNOWLEDGEMENT

*This study is partially supported by the Center for Connected Multimodal Mobility (C2M2) (USDOT Tier 1 University Transportation Center) headquartered at Clemson University, Clemson, SC. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of C2M2, and the official policy or position of the USDOT/OST-R, or any State or other entity, and the U.S. Government assumes no liability for the contents or use thereof.*

## Technical Report Documentation Page

| 1. Report No. | 2. Government Accession No. | 3. Recipient's Catalog No. |
|---|---|---|
| 4. Title and Subtitle<br>Real-time Decentralized Framework for Technology-Enabled Intermodal Freight Transport | | 5. Report Date<br>October 2023<br>6. Performing Organization Code |
| 7. Author(s)<br>Nathan Huynh, Ph.D, ORCID: 0000-0002-4605-5651<br>William Ferrell, Ph.D, ORCID: 0000-0002-6578-522X<br>Bhavya Padmanabhan, ORCID: 0000-0002-7087-5334 | | 8. Performing Organization Report No. |
| 9. Performing Organization Name and Address<br>University of South Carolina<br>300 Main St, Columbia, SC 29208 | | 10. Work Unit No.<br>11. Contract or Grant No. |
| 12. Sponsoring Agency Name and Address<br>Center for Connected Multimodal Mobility ($C^2M^2$)<br>USDOT Tier 1 University Transportation Center<br>Clemson University<br>200 Lowry Hall, Clemson<br>Clemson, SC 29634 | | 13. Type of Report and Period Covered<br>14. Sponsoring Agency Code |
| 15. Supplementary Notes | | |

**16. Abstract**

This study addresses the less than truckload pickup and delivery problem where carriers collaborate in a decentralized and dynamic manner. It is assumed that carriers do not have any collaboration agreement with one another; however, they collaborate indirectly due to the need to outsource jobs and/or the opportunity to bid for jobs to increase profit. In the context of this study, trucks operate independently and make decisions on outsourcing and insourcing of jobs on their own. To address this problem, an approach based on a multi-round combinatorial auction (CA) is proposed. The novelty of the proposed approach is that in each round of the auction, trucks can bid for multiple bundles in contrast to the commonly used approach of a single bundle. The multiple bundles consist of the most profitable bundle that is determined by each truck and all or some subsets of this bundle. For the proposed multi-round CA approach, two integer programming models are developed: 1) for trucks to select the bundle of jobs to bid in real-time, and 2) for the central clearinghouse to select the winning trucks. Numerical experiments are conducted using an actual transportation network with up to 50 hypothetical jobs to be completed by five carriers. The results indicate that the proposed auction method provides a higher profit in all instances tested.

| 17. Keywords | 18. Distribution Statement |
|---|---|
| Carrier collaboration, decentralized, dynamic pickup and delivery problem, and combinatorial auction | No Restrictions. |

| 19. Security Classif. (of this report) | 20. Security Classif. (of this page) | 21. No. of Pages | 22. Price |
|---|---|---|---|
| Unclassified | Unclassified | 41 | NA |

# Table of Contents

## LIST OF TABLES



## LIST OF FIGURES

# EXECUTIVE SUMMARY

This project focuses on developing a real-time, decentralized framework for less than truckload carrier (LTL) collaboration. The collaboration is assumed to be indirect, meaning carriers do not have any collaboration agreement with one another; however, they collaborate due to the need to outsource jobs and/or the opportunity to bid for jobs to increase profit. It is assumed that carriers receive jobs from their customers and determine optimal truck routes to serve these jobs. If trucks cannot fulfill the jobs within the time window, they can outsource those jobs in real-time. Also, trucks can bid for the jobs outsourced by the other trucks in real-time. To achieve this job exchange among the carriers, an iterative combinatorial auction (CA) mechanism is adapted. In the proposed iterative CA, trucks determine the most profitable bundle of jobs and submit bids that include some or all subsets of the most profitable bundle. Mathematical models are developed: 1) for trucks to select the most profitable bundle of jobs to bid in real-time (*dynamic bundle selection model*) and 2) to resolve conflict when multiple trucks bid for the same jobs and to select the winning truck(s) (*conflict resolution model*). Solution methods based on large neighborhood search (LNS) is developed to solve the *dynamic bundle selection model*. Numerical experiments showed that the combinatorial auction approach with multiple bundle (includes the profitable subsets of the most profitable bundle) bidding provides a higher profit and a higher number of jobs served in less time compared to single bundle (includes only the most profitable bundle) bidding. That means the benefit from multiple bundle bidding approach is significantly high with respect to the small effort required to determine the profitable subsets of the most profitable bundle which comprise the multiple bundles. If trucks have the most profitable bundle and the optimal route to serve the jobs in the bundle, they can determine the marginal profit from any subset in two steps: 1) remove the job(s) that is not included in the subset from the route, and 2) recalculate the profit of the new route. In multiple bundle bidding, keeping the value of the required minimum increases in profit (α) when deciding on which bundle of jobs to bid between 5 to 10% is more likely to provide a higher increase in profit for the participating trucks. In addition, if all trucks' α values are less than 20%, the process of resolving the conflicts is expedited because there are more bundle combinations in the bids submitted by the trucks. The multiple bundle bidding approach is also effective when approximately 50% of the trucks of the collaborating carriers are not available to participate; this finding is conducive to actual practice. Moreover, the higher the number of outsourced jobs or higher the number of nodes in the transport network, the better the multiple bundle bidding approach will perform in terms of higher number of jobs served and a higher profit in less time than the single bundle approach. For example, when the number of outsourced jobs is 2 times, the difference in increase in profit is 12 times whereas when the number of outsourced jobs is 4 times, the difference in increase is profit is 28 times. Similarly, when the number of outsourced jobs is 2 times, the difference in the number of jobs served is 14 times whereas when the number of outsourced jobs is 4 times, the difference in number of jobs served is 56 times.

# CHAPTER 1

# Introduction

The less-than-truckload (LTL) carriers that are the focus of this study have trucks that pickup and deliver jobs to customers in a rather compact region. A single truck handles both transactions for a job so a critical aspect of operations is using the least-cost routes to serve all jobs that are assigned. Customer expectation and competition are making “faster and cheaper” a nonnegotiable part of success; hence, they are seeking carriers that can deliver with ever-shorter lead times and at the lowest possible cost. In this environment, large carriers have a competitive advantage over small carriers due to their market power. For small-to-medium-sized LTL carriers to stay in business, they must improve their efficiency and lower cost. A potential strategy to achieve these goals is to collaborate with other carriers (Padmanabhan et al., 2023). This type of collaboration is known as horizontal collaboration because organizations at the same echelon in a supply chain collaborate with one another ( Padmanabhan et al., 2023, Ferrell et al., 2019). In collaborating, carriers would exchange and perform certain jobs for each other to make more profit. There are some real-world examples of horizontal collaboration in the trucking industry. For instance, the Belgian posts reduced total miles driven by 25% and emissions by 90% through *collaborative urban logistics & Transport (CULT)* ( Padmanabhan et al., 2023, *Driving Reduced by 25% and Emissions by 90% through Combined City Deliveries in Antwerp*, n.d.). In CULT, parcels from multiple carriers are consolidated at warehouses outside city centers to provide last miles deliveries. Despite the benefits, the trucking industry has yet to adopt horizontal collaboration on a large scale ( Padmanabhan et al., 2023, Krajewska et al., 2008).

Most of the previous LTL carrier collaboration studies such as (Berger & Bierwirth, 2010; Dai et al., 2014; Padmanabhan et al., 2020) addressed the static version of the problem in which there is a collaboration agreement and carriers make job exchange decisions before their trucks are dispatched from the depot. They assumed that the carriers receive jobs from their customers and then share some or all of the jobs that are better served by their coalition members. These studies aimed to find the optimal reassignment of shared jobs to the trucks and optimal routes for trucks to serve these jobs so that both the coalition profit and individual carriers’ profits are maximized. However, some trucks may not be able to serve all the allocated jobs within the available time window due to unexpected delay at customer locations or on roadways. This limitation of the static framework can be overcome by considering a dynamic framework in which the trucks can outsource the jobs they are unable to fulfill. In addition, the dynamic framework has the advantage that the new jobs that the carriers may receive from the customers in real-time can be considered. Thus, the carriers can allocate new jobs with high marginal profit to its truck(s) and/or outsource the nonprofitable new jobs. In this study, a dynamic LTL carrier collaboration problem is considered in which trucks of participating carriers operate independently and make real-time decisions on job outsourcing and insourcing. It is assumed that there are multiple carriers interested in collaborating with each other indirectly and each carrier has multiple trucks interested in the real-time job exchange due to the need to outsource jobs and/or the opportunity to bid for jobs to increase profit. This proposed framework of collaboration is called dynamic indirect collaboration hereafter.

There are four scenarios in which a truck may need or want to outsource some jobs after leaving the depot. First, when there is a disruption to a truck's current route (e.g., severe congestion due to a traffic accident**)**, the truck is forced to take an alternate route that can prevent the truck from being able to serve some jobs within the time window and/or the total travel time may exceed the driver's maximum hours of service. Second, the enroute truck may encounter a breakdown due to engine failure or crash. In this scenario, the truck must outsource all unserved jobs. Third, a truck may receive new jobs that are more profitable than some of its assigned jobs. In this scenario, the truck would outsource the least profitable current job(s) to meet all deadlines and restrictions. Fourth, the carrier itself receives new jobs that are not as profitable. In this scenario, it directly outsources the new jobs instead of assigning them to its trucks.

In this study, it is assumed that auction is the preferred mechanism for competing carriers/trucks to bid for the outsourced jobs in real-time, and there is a central clearinghouse or auctioneer that oversees the bidding process. To model the proposed problem, the combinatorial auction (CA) approach, which has been well studied in the energy sector (e.g., (*Uber's Self-Driving Truck Startup Otto Makes Its First Delivery | WIRED*, n.d.; Zaidi et al., 2018, 2021)) is applied. As such, multiple trucks are allowed to bid for multiple bundles; in CA, although jobs are sourced individually, they are combined into bundles by the clearinghouse/bidder, because the value of a set of jobs might be higher for a truck than the sum of the values of all individual jobs. Single round CA involves one-time submission of bids, i.e., each bidder should provide information about its valuation for all $2^m$-1 (where *m* is the number of goods) combination of the goods to the auctioneer, and the winner is selected based on this information. On the other hand, multi-round CA involves multiple iterations of the submission of bids. In each iteration, bidders bid for a bundle of jobs of their choice and may change their bid in the next iteration based on current bids. The decision on whether to change the bid or not is made based on the "mode" of the multi-round auction. There are two modes in multi-round CA: quantity-setting based and price-setting based. In each round of the quantity setting-based auctions, the bidders send the price that they are willing to pay for the items they want to buy. The auctioneer makes the provisional allocation based on the price of the bid, and then the bidders adjust the price of their bid in the iteration. In each round of the price- setting based auctions, the bidders send their selection of the items they want to buy at the given prices. The auctioneer resolves the conflict in the bids submitted, makes provisional allocation, and adjust the price of the goods. In carrier collaboration problem, pickup and delivery jobs are considered as goods and the carriers as bidders. Note that in this study, the *trucks* are considered as bidders rather than the carriers. The price-setting based combinatorial auction is adopted in this study as this is more appropriate for the proposed problem (Dai et al., 2014). This is because, in this study, it is assumed that each truck selects its most profitable bundle of jobs by solving the dynamic vehicle routing problem with pickup and delivery and time windows.

The novelty and contributions of this study are as follows: 1) a mixed integer linear programming model for trucks to select the most profitable bundle of jobs to bid in real-time (this model is hereafter called the dynamic bundle selection model), and 2) an integer programming model to resolve conflict when multiple trucks bid for the same jobs and to select the winning truck(s) (this model is hereafter called the conflict resolution model). The dynamic bundle selection model and the conflict resolution model are used in a multi-round CA framework that allows for bidding of multiple bundles instead of a single bundle. The multiple bundles include one most profitable bundle and all profitable subsets of this bundle.

The next chapter (Chapter 2) presents a literature review of related work. Chapter 3 describes the Methodology. Chapter 4 presents the Computational study. Chapter 5 presents the results and discussion. Lastly, Chapter 6 presents this study's summary and conclusion/

# CHAPTER 2

## Literature Review

The following review is focused on horizontal collaboration among LTL carriers using the decentralized approach and auction as a job exchange mechanism. Readers are referred to the work of Gansterer and Hartl (2017) for a detailed review of prior work on LTL centralized, and non-auction based decentralized collaborative planning.

The five steps of CA for job exchange among carriers in a decentralized environment proposed by Berger and Bierwirth (2010) are as follows: 1) selection of pickup and delivery jobs to outsource, 2) generation of bundles of outsourced jobs, 3) determination of marginal profit, 4) determination of the winner, and 5) profit sharing. Previous studies on decentralized carrier collaboration considered some or all of these steps.

Gansterer and Hartl (2016) focused on generating candidate jobs to outsource for a carrier. According to the authors, to maximize profit, the carriers should try to outsource the jobs that are valuable to other collaborating carriers. They proposed two strategies for the selection of jobs to be outsourced: 1) geographically-based, and 2) profit-based. They found that the geographically-based approach outperformed the profit-based one. Schopka and Kopfer (2017) also investigated the generation of candidate jobs that a carrier should outsource. They proposed that a carrier select the jobs based on the approximate potential to increase the profit. The jobs with the highest potential are incorporated into existing routes to be fulfilled by the carrier and the rest are released for job exchange. Ruther and Rieck (2022) addressed the auction-based decentralized collaboration problem but focused on bundle generation by the auctioneer. They proposed a scenario-based bundle selection approach in combination with two pre-selection techniques: cluster-based and neural network-based. Their approach was found to be 6.82% to 10.71% more profitable than the previously used bundle selection techniques.

Krajewska et al. (2008) was the first to propose an auction-based approach for job exchange between carriers. They developed a methodology for decentralized carrier collaboration based on a single-round CA. Their numerical results showed that collaboration can yield considerable cost decrease and efficient profit allocation is possible by using cooperative game theory. Berger and Bierwirth (2010) proposed a two-job exchange mechanisms for the decentralized carrier collaboration based on a single round CA: single job exchange with another carrier and bundle job exchange with another carrier. They compared the results of the two mechanisms and found that bundle job reassignment provided a higher benefit than the single job reassignment. Li et al. **(**2015) investigated the single job auction approach for the carrier collaboration problem to improve the overall profit of the alliance. They proposed two bundle selection models and one bundle exchange model. They stated that their proposed methodology is superior to the other four existing methods.

Nicola (2022) compared four different job exchange mechanisms in auction based carrier collaboration. The iterative CA mechanism was found to be better than the single round. Individual-auction mechanism and centralized-auction mechanism provided similar results and both outperformed the posted price mechanism. The results also showed that the difference in the improvement comes at the cost of providing more information. Gansterer et al. (Gansterer et al., 2020) found that sharing information among the carriers reduced the number of bundles generated by up to 80%. They assessed the benefits of sharing information in an auction-based decentralized carrier collaboration. They considered two types of information sharing: 1) information that can enhance the selection of transportation jobs to submit to the job pool (aggregate information), and 2) information that can enhance the packaging of transportation jobs into bundles (partial information). They found that sharing aggregate information resulted in a 20% increase in profit than sharing partial information. Conversely, sharing more detailed information decreased the profit.

One of the limitations of the single round CA is the difficulty in managing a large number of bundles set. Several studies attempted to overcome this limitation. For instance, Dai et al. (Dai et al., 2014) developed a methodology for carrier collaboration based on iterative CA and Lagrangian relaxation. In iterative CA, each carrier submits its most profitable bundle in each round; not all bundles are evaluated in a single round, and thus, the computational complexity is reduced. The auctioneer adjusts the prices of the jobs based on the previous bids. They proposed and compared three different price adjustment methods for iterative CA. Numerical experiments on randomly generated instances were performed to demonstrate the effectiveness of the proposed approach. Lyu et al. (Lyu et al., 2019) also proposed an iterative request exchange mechanism for LTL carrier collaboration based on CA. They compared their approach to the methods proposed by Berger and Bierwirth (Berger & Bierwirth, 2010) and Gansterer and Hartl (Gansterer & Hartl, 2016). They concluded that their method provided 11.80% more profit than the other two methods. Li et al. (Y. Li et al., 2016) addressed a bid generation problem in which the carriers select the most profitable bundle of jobs to bid. Their contribution is an adaptive large neighborhood search heuristic to solve the bid generation problem.

The studies in the foregoing review considered the static version of the decentralized carrier collaboration problem. The studies discussed in this paragraph considered the dynamic carrier collaboration problem. Dai and Chen (2011) proposed a multi-agent and auction-based framework. In their study, there is no central auctioneer; a carrier acts as an auctioneer when it wants to outsource a job and acts as a bidder when it intends to acquire a job. They compared their proposed framework against centralized collaborative planning. Los et al. (Los et al., 2020) proposed a platform based dynamic carrier collaboration problem and investigated the value of information sharing. They used an auction-based multi-agent system to solve the problem and investigated nine different information sharing policies. The value of information in each of the policies was discussed in detail. Hernández et al. (Hernández et al., 2011) proposed a deterministic dynamic carrier collaboration problem in which a single carrier of interest tries to minimize the cost of transportation by acquiring capacities from the collaborating partners. Their problem is dynamic because the collaborative capacities are time dependent and the actual holding costs encountered by a load depend on the number of intervals it is held at a transfer location. They found that as the degree of collaboration increases, the relative attractiveness of acquiring collaborative capacity also increases compared to the short-term leasing option, leading to increased capacity utilization.

The collaboration problems considered in Los et al. (Los et al., 2020) and Hernández et al. (Hernández et al., 2011) are different from this study because the former considered a platform based carrier collaboration problem and the latter considered the perspective of a single carrier. There are several differences between this study and that of Dai et al. (Dai & Chen, 2011). First, they assumed that the carriers collaborate by sharing their jobs dynamically just to make more profit. The carriers selected the jobs to outsource by solving an outsourcing job selection problem. In contrast, this study assumes that there is no prior agreement between the carriers; they collaborate indirectly due to the need to outsource jobs and/or the opportunity to bid for jobs in real-time to increase profit. Second, in their work, the auction is asynchronous because they assumed that an auction starts when a carrier announces its outsourced jobs. In an asynchronous auction, there is no order for job outsourcing, bid generation, and price adjustment. In this study, an auction starts at the end of each defined period when a CC announces the jobs outsourced during that period. Thus, the auction is synchronous, meaning the outsourcing of the jobs happens first, followed by bidding and then price adjustment. Third, they described the dynamic behavior of the carrier collaboration by defining different states of each job: before auction, in auction, auctioned, in execution and executed. In this study, the dynamic behavior of the carrier collaboration problem is captured by using the decision variables that can be tracked in every time period. Fourth, they assumed that each carrier bids for the most profitable bundle of jobs; however, this study assumes that each truck bids for multiple bundles of jobs. The multiple bundles consist of the most profitable bundle that is determined by each truck and all or some subsets of this bundle. Fifth, in their study, conflict resolution is only implemented after the auction is completed. They assumed that after the completion of the iterative process, if multiple carriers are interested in a single job, the winner will be the carrier who announced the bid earliest. However, in this study, a conflict resolution model is developed and implemented after every auction round.

# CHAPTER 3
# Methodology

## 3.1 Problem description

In the proposed problem, an indirect collaboration among LTL carriers is considered. The previously discussed dynamic framework is employed. Carriers only exchange jobs when there is a need to outsource jobs and/or when there is an opportunity to bid for jobs to increase profit. Each carrier f = {1, …, d}, (where d is the number of carriers) initially receives mf jobs directly from customers and independently finds optimal truck routes (static decisions) to serve these jobs to maximize profit. Each job has a pickup point and a delivery point, and each such point is assumed to be a node in the network. Each truck can make the job outsourcing and bidding decisions on its own. In a dynamic setting, this approach expedites the operational process since trucks do not need to wait for instructions from the carrier. It is assumed that in maximizing its profit, a truck also maximizes its carrier's profit. There are two scenarios in which a truck may need or want to outsource its jobs after leaving the depot and there are two scenarios in which a carrier itself outsources new jobs in real-time. These four scenarios are explained in the introduction.

A truck can submit a request to outsource a job at any time along with the outsourcing price of that job. However, the auction process will take place between periods, so jobs outsourced during a period are auctioned (made known to other trucks) only at the end of that period. A period is defined to be 15 minutes. Figure 3.1 illustrates the auction process. The intent of this illustration is to convey the dynamic nature of job outsourcing and bidding and the auction process used to manage them. There are three carriers, and each carrier has one truck enroute. The truck of carrier 1 is truck 1, the truck of carrier 2 is truck 2, and the truck of carrier 3 is truck 3. Each truck serves two jobs. Figure 3.1a shows the initial routes of the trucks that were determined by the respective carrier; these routes are determined before the trucks departed from the depot. Recall that a truck or a carrier can outsource a job at any time and a carrier can receive a new job from its customers at any time. However, in this example, it is assumed that carrier 3 received a job at 6 minutes into the first period (the new job's location is shown in Figure 3.1b) and truck 2 outsourced one of its assigned jobs at 8 minutes into the first period (the outsourced job and the new route of truck 2 is shown in Figure 3.1c). Note that the new jobs from customers go to carriers, not trucks. In this instance, Carrier 3 would check if it would be profitable for one of its trucks (truck 3 in this example) to serve the new job. If it is not, Carrier 3 would outsource the job. At the end of the 15-minute period, the auctioneer announces the availability of two outsourced jobs to all trucks. Trucks independently determine the most profitable bundle of jobs and then bid for it. The CC resolves the conflicts and determines provisional winner(s). The CC also adjusts the outsourcing price of the jobs if the stopping criterion has not yet been met. This iterative process of bidding and price updating continues until the stopping criterion is satisfied and the final winner is announced. The jobs are then transferred to the winning trucks along with the outsourcing price. This final outsourcing price is the payment for the truck that serves the job. The trucks involved in the exchange process subsequently update their routes as shown in Figure 3.1d. In this example, the auction process took 1 minute.

In the first auction, the job outsourced by carrier 3 (the new job) is won by truck 1, and the job outsourced by truck 2 is won by truck 3. The auctioning dynamics described for the first 15-minute period is repeated until the end of the planning horizon.

After the completion of each auction, there are two possible outcomes: 1) all outsourced jobs are assigned to the trucks that won the bids, or 2) some outsourced jobs are not selected by any of the trucks. If the first outcome occurs, no further action is required because the results are favorable to all parties. If the second outcome occurs, it is assumed that the job is returned to the carrier of the truck that tried to outsource the job; that carrier is ultimately responsible for taking the necessary action to serve the job. The reason why the job will not go back to the truck that outsourced it is that the truck may have accepted some other jobs during the auction process. For this reason, this study assumes that the carriers will have the responsibility of reassigning the unbid job to one of its trucks.

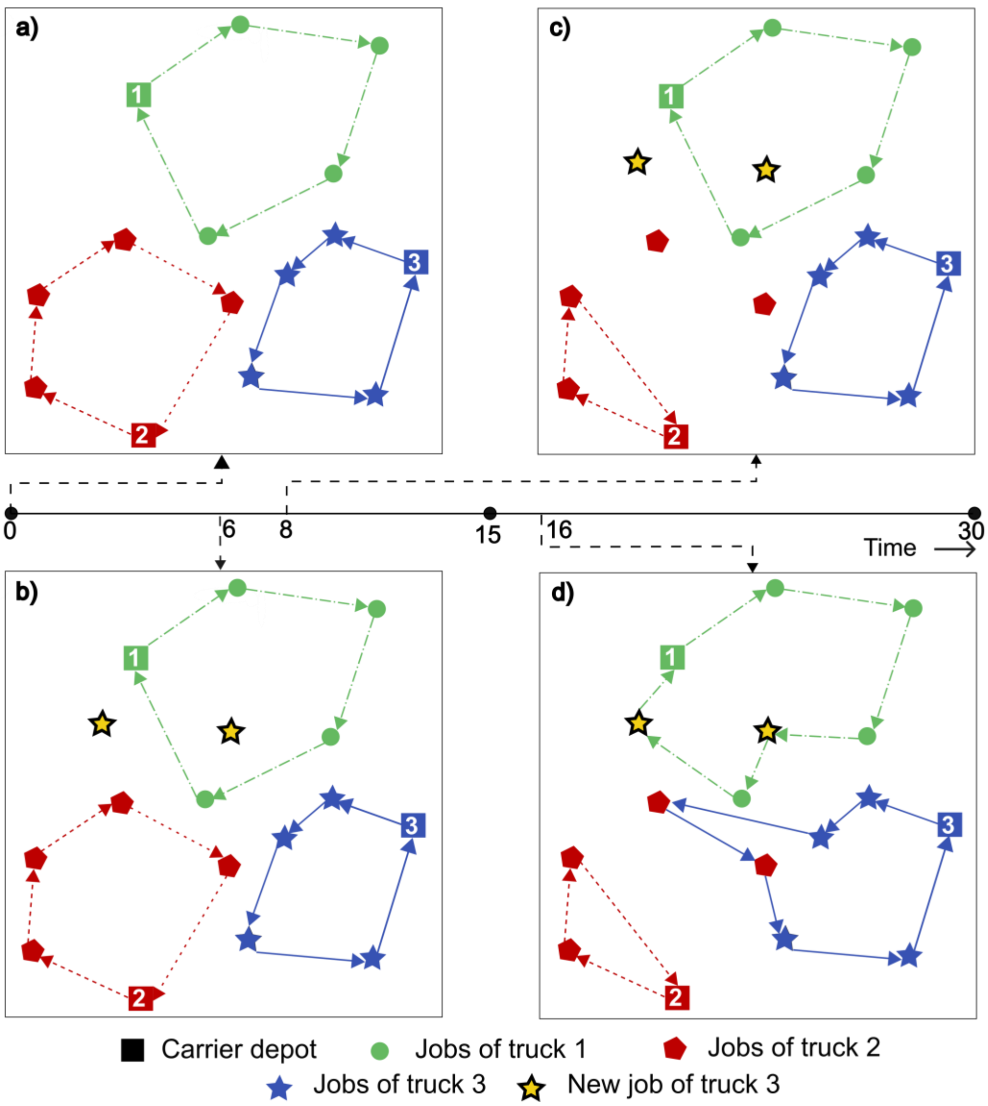


**Figure 0.1 Illustrative example of the iterative combinatorial auction process**

## 3.2 Sets and parameters

The sets and parameters used in this study are provided below.

**Sets and Description**

| **Sets** | | **Description** |
|---|---|---|
| $D$ | = | $\{1, . . ., d\}$, set of depots or set of all carriers and $d$ is the number of carriers participating indirect collaboration. |
| $PK$ | = | $\{d + 1, …, d + m\}$, set of pickup nodes, where $m$ is the total number of jobs received by all carriers |
| $DL$ | = | $\{d + m + 1, …, d + 2m\}$, set of delivery nodes |
| $O$ | = | $PK \cup DL$, set of all customer nodes |
| $N$ | = | $D \cup O$, set of all nodes in the network |
| $V_f$ | = | Set of trucks at depot $f \in D$ |
| $V$ | = | $V_1 \cup V_2 \cup ,…, \cup V_d$, set of all trucks |
| $TK$ | = | Set of all trucks enroute of all collaborating carriers |
| $PKv$ | = | For each truck $v$, union of the set of pickup nodes of current jobs (excluding the job outsourced by that truck) and the set of pickup nodes of the outsourced jobs of other trucks |
| $DLv$ | = | For each truck $v$, union of the set of delivery nodes of current jobs (excluding the job outsourced by that truck) and the set of delivery nodes of the outsourced jobs of other trucks |
| $Ov$ | = | For each truck $v$, $PKv \cup DLv$ |
| $Nv$ | = | For each truck $v$, $Ov \cup \{depot\}$, where *depot* is the depot node from where the truck started the route (i.e., its carriers depot) |
| $PK_0$ | = | For each truck, when an auction starts, the set of pickup nodes of current jobs that are yet to be visited |
| $DL_0$ | = | For each truck, when an auction starts, the set of delivery nodes of current jobs that are yet to be visited |
| $N_0$ | = | For each truck, $PK_0 \cup DL_0 \cup \{depot\}$, |
| $DL_3$ | = | For each truck, set of delivery nodes of the jobs whose pickup nodes are already visited |
| $O_c$ | = | For each truck, pickup and delivery nodes of the current job |
| $N_c$ | = | $Oc \cup \{depot\}$ |
| $O_1$ | = | For each truck, set of pickup and delivery nodes that are already visited |
| $N_1$ | = | $O_1 \cup \{depot\}$ |
| $N_2$ | = | For each truck, union of the pickup and delivery nodes of its outsourced jobs |
| $K_r$ | = | The set of bundles of jobs bid by truck $r$ |
| $R$ | = | Set of all trucks enroute |
| $B$ | = | Set of all outsourced jobs of all trucks in a single period |

**Parameters and Description**

| **Parameters** | | **Description** |
|---|---|---|
| $d$ | = | Total number of carrier depots which is equal to the total number of carriers in the coalition |

| | | |
|---|---|---|
| $m$ | = | The total number of jobs received by all carriers in the coalition |
| $m_f$ | = | Number of jobs received by carrier $f = \{1,\ldots,d\}$ |
| $p_i$ | = | Revenue for serving the job $i = \{1, \ldots, m\}$(price paid by the customer to serve job $i$) |
| $K$ | = | Vehicle capacity |
| $H$ | = | 14 hours (Maximum service hours allowed in one route) |
| $c_{ij}$ | = | Cost of travel from node i $\in N$ to node $j \in N$, $i \neq j$ |
| $q_i$ | = | Demand/supply at node i $\in O$ (positive sign represents a pickup and negative sign represents a drop off) |
| $t_{ij}$ | = | The time required to traverse the arc connecting node $i$ and node $j$, i $\in N$, $j \in N$ , $i \neq j$ |
| $a_i$ | = | The earliest acceptable pickup/delivery time at node i $\in O$ |
| $b_i$ | = | The latest acceptable pickup/delivery time at node i $\in O$ |
| $M$ | = | Big number |
| $tk$ | = | Total number of all trucks of all collaborating carriers |
| $\beta$ | = | For each truck, the node at which the truck is currently located or heading to at the end of the period (e.g., t=15, 30, 45 etc.) in which the outsourced jobs are considered for reassignment |
| *depot* | = | For each truck, its carrier's depot from where it originally started the route |
| $\alpha$ | = | Percentage of minimum increase in profit set by the trucks to include a bundle in the bid |
| $\delta_r^k$ | = | Increase in profit of truck $r \in TK$ from bundle $k \subseteq B$ |

### 3.3 Multiple bundle iterative combinatorial auction for the dynamic job exchange among the trucks of multiple LTL carriers

Dai et al. (2014) and Dai and Chen (Dai & Chen, 2011) proposed an iterative CA based framework for carrier collaboration in which, in each round of the auction, each carrier determines the most profitable bundle of jobs to bid by solving a pickup and delivery problem. Based on the bid submitted by the trucks, the outsourcing price of the jobs is adjusted in every auction round. This approach reduces the complexity of analyzing all possible combinations of outsourced jobs while bidding (Dai et al., 2014; Dai & Chen, 2011). However, they assumed that the auction process terminates if: 1) there is no conflict in the bid submitted; each outsourced job is bid by exactly one carrier, 2) the given number of auction rounds is achieved, or 3) the solution is not improved for a given number of rounds of auction. There are some limitations to their approach: 1) no guarantee that the conflict will be resolved and get a feasible solution even after the termination of the auction because the conflict resolution procedure depends upon how the carriers change their bids according to the price adjustment of the outsourced jobs after every round of the auction (Dai et al., 2014; Dai & Chen, 2011), 2) if there are a large number of bidders, the outsourcing price of a job will decrease proportional to the number of bidders interested in that job. If the outsourcing price of a job is too low, then all carriers might drop that job from their bid. In such situations, more rounds of auction might be required to get a feasible solution which takes a large amount of time. To overcome these limitations, this study proposes an approach that allows each bidder to submit some or all profitable subsets of the most profitable bundle. The proposed approach is called multiple bundle bidding (MBB) and the single bundle bidding approach is called SBB hereafter. The flowchart in Figure 3.2 shows the general steps of the proposed approach and each step is explained in detail in the subsequent paragraphs.

### *1.1.1. Carriers determine truck routes*

Each carrier determines the routes for their trucks to serve the jobs received from its customers. This is a static decision. The mathematical model and heuristic developed by Padmanabhan et al. (Padmanabhan et al., 2020) are used to determine the truck routes for each carrier.

### *1.1.2. Trucks outsource jobs, and carriers receive new jobs*

The trucks can outsource their assigned jobs at any time if they cannot serve them. In addition, at any time, the carriers may receive new jobs. When a carrier receives a new job from its customers, it will solve a dynamic pickup and delivery problem to determine if the new job is profitable when added to one of its trucks' routes. The *bundle selection model* (see section 3.3.4) developed in this study is used to achieve this.

When a carrier wants to outsource a new job it received or a truck wants to outsource one of its current jobs, it provides the CC with the job's initial and maximum outsourcing price. The initial outsourcing price is determined based on the minimum profit requirement of the carriers, as shown in Equation 1. The maximum outsourcing price is decided by the truck/carrier that outsourced the jobs.

$$op_m = p_m - \phi * p_m \tag{1}$$

Where $op_m$ is the initial outsourcing price of job $m$, $p_m$ is the price paid by the customer to serve job $m$, and $\phi$ is the minimum profit margin of the carriers. Since the carriers need not share the value of $op_m$ and $\phi$, the other carriers will not know the confidential information such as the price a carrier collects from its customer and the profit made by the carrier.

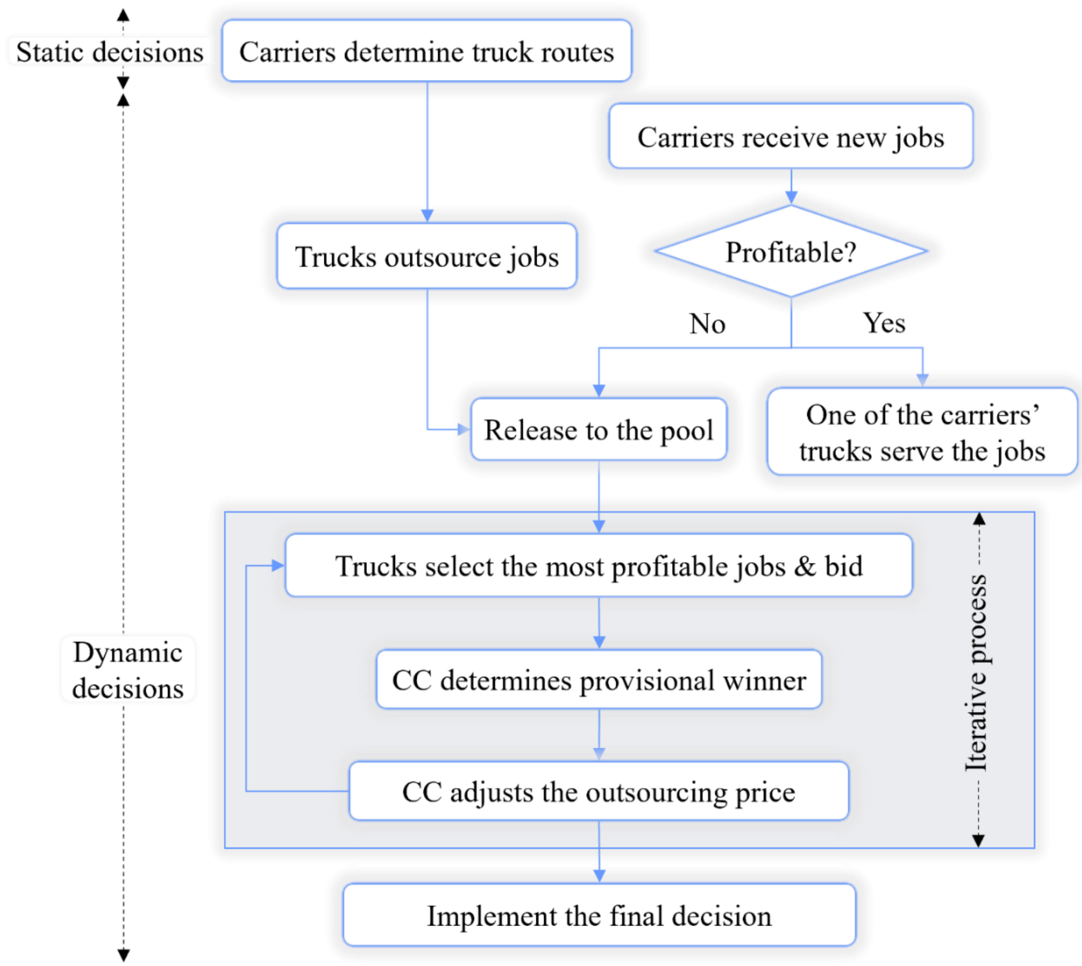


**Figure 0.2.Proposed multiple bundle CA framework**

### 1.1.3. *Release to the pool*

At the end of each period, the CC releases the list of jobs that have been outsourced during that period. This list provided to the trucks of all carriers who wish to participate in the indirect collaboration. Note that the auction is assumed to be periodic.

### 1.1.4. *Trucks select the most profitable jobs and bid*

Each truck independently determines the most profitable bundle of jobs from the list of outsourced jobs by solving the *dynamic bundle selection model* which is provided below. Note that the trucks enroute consider the current location as the starting point to solve the *dynamic bundle selection model,* not from the original starting point (depot).

*Decision variables*

$y_i^t$ = 1, if job $i$ is selected to serve during the time period that starts at $t$; 0 otherwise. Where $i$ is the job that has pickup location at $i \in PK_v$.

$x_{ij}^t$ = 1, if the truck travels from node $i$ to node $j$ during the time period that starts at $t$; 0 otherwise ($i \in N_v$, $j \in N_v$ ).

$Q_{ij}^t$ = Quantity transported across arc ($i \in N_v$ , $j \in N_v$ ) during the time period that starts at $t$.

$S_i^t$ = The time at which the truck begins the service at node $i \in N$ during the time period that starts at $t$.

*objective function*

$$Max\ Z = \sum_{i \in N_v} p_i y_i^t - \sum_{i \in N_v} \sum_{j \in N_v} c_{ij} x_{ij}^t \tag{2}$$

*subject to*

$$\sum_{j \in N_v \setminus N_1, j \neq \beta} x_{\beta j}^t = 1 \tag{3}$$

$$\sum_{i \in N_v} \sum_{j \in N_v} t_{ij} x_{ij}^t \leq H \tag{4}$$

$$\sum_{i \in N_0 \setminus \{depot\}, i \neq j} x_{ij}^t = 1 \quad \forall \ j \in DL_3 \tag{5}$$

$$\sum_{i \in N_v} x_{ij}^t \leq 1 \ \forall \ j \in N_v \tag{6}$$

$$Q_{ij}^t \leq K x_{ij}^t \quad \forall \quad i \in N_v, j \in N_v, i \neq j \tag{7}$$

$$\sum_{j \in N_v \setminus \{depot\}} \left[ Q_{ij}^t - Q_{ji}^t \right] = q_i y_i^t \quad \forall \quad i \in N_v \setminus \{depot\} \tag{8}$$

$$\sum_{j \in N_v, j \neq i} x_{ji}^t - \sum_{j \in N_v, j \neq i} x_{ij}^t = 0 \quad \forall \quad i \in N_{v \setminus \{depot\}} \tag{9}$$

$$S_j^t \geq S_i^t + t_{ij} x_{ij}^t - M(1 - x_{ij}^t) \ \forall \ i \in N_v, j \in N_v \setminus \{depot\}, i \neq j \tag{10}$$

$$a_i \leq S_i^t \leq b_i \quad \forall \ i \in N_v \setminus \{depot\} \tag{11}$$

$$S_i^t \leq S_{i+m}^t \quad \forall \ i \in PK_v \tag{12}$$

$$\sum_{i \in N_v, i \neq j+m, i \neq j} x_{ij}^t - \sum_{i \in N_v, i \neq j+m} x_{ij+m}^t = 0 \ \forall \ j \in PK_v \tag{13}$$

$$\sum_{j \in N_v} x_{ij}^t = y_i^t \ \forall \ i \in PK_v \tag{14}$$

$$\sum_{i \in N_v} x_{ij}^t = 1 \ \forall \ j \in N_c / O_1 \tag{15}$$

$$\sum_{i \in N_v} x_{ij}^t = 0 \ \forall \ j \in N_2 \tag{16}$$

$$Q_{ij}^t \geq 0, S_i^t \geq 0, \ y_i^t = \{0,1\}, \ x_{ij}^t = \{0, 1\} \quad \forall \quad i \in N_v, j \in N_v \tag{17}$$

The objective function (2) maximizes the profit of the truck. Note that in this study, it is assumed that maximizing the profit of each truck effectively increases the profit of the carrier. Constraints (3) ensure that the node at which a truck is currently located (or heading to) is the starting point in the dynamic framework. Constraints (4) restrict the maximum number of hours traveled in one route. Constraints (5) ensure that if the truck has already visited a job's pickup node (before time *t*), then its delivery node must be visited. Constraints (6) ensure that a customer node cannot be visited more than once. Constraints (7) are the vehicle capacity constraints that ensure the truck's capacity is not exceeded. Constraints (8) balance the flows across each node. “They guarantee that the difference between the incoming and the outgoing products flow in a node will be equal to the supply or demand at that node (Huynh et al., n.d.)” . Constraints (9) ensure that a vehicle arriving at a node must leave the node and a vehicle leaving a node must come back to the node. Constraints (10) and (11) ensure the vehicles satisfy the time window constraints. Constraints (10) give time ($S_j^t$) at which the truck begins service at node $j \in N$ if the truck came from node $i$. Constraints (10) also ensure that the trucks originally start and end only at the carrier depots, and thus, eliminates sub tour formation. Constraints (11) ensure that the time at which a truck arrives/starts from a node $i \in N$ is within the allowable time window; that is, after the earliest pickup/delivery time and also before the latest time for that node. In this study, the service time is assumed to be zero; thus, the arrival time and starting time of a truck at a node are the same. Constraints (12) ensure that the pickup node is visited before the corresponding delivery node. Constraints (13) ensure that if the pickup node of a job is visited, the delivery node of the same job must be visited. Constraints (14) guarantee that assigned jobs are picked up; if a job is not selected, the truck does not visit that job's pickup node. Constraints (15) ensure that the pickup and delivery locations of the current jobs that are not already visited by the truck must be visited. Constraints (16) ensure that a truck will not serve its outsourced jobs. Constraints (17) show the nature of the decision variables.

As mentioned, in this study, trucks are allowed to bid for multiple bundles of jobs rather than bidding for a single bundle of jobs as done in previous studies (Dai et al. 2014; Dai & Chen 2011). Recall that the set of multiple bundles consists of the most profitable bundle and some or all profitable subsets of the most profitable bundle. For each truck, the *dynamic bundle selection model* gives the most profitable bundle and the route to serve the current jobs (excluding the jobs already served) and those in the most profitable bundle. Determining the profitable subsets of the most profitable bundle does not require additional solving processes and the reason is provided below. The selection process of the subsets of the most profitable bundle to include in the bid is explained using Figure 3.3. Figure 3.3a shows the current route of a truck; there are three jobs in this route, J1, J2 and J3. The pickup location of a job *i* is shown as JiP and the delivery location of a job *i* is shown as JiD, $i = \{1, 2, 3, 4, 5\}$. The most profitable bundle of the truck is {J4, J5} and the most profitable route determined by using the *dynamic bundle selection model* is shown in Figure 3.3b. To determine the marginal profit of subset {J4}, remove J5 from the route shown in Figure 3.3b and reroute it as the truck directly goes from node J4D to the depot (shown in Figure 3c). Since the cost to traverse between any two nodes is known, the marginal profit of subset {J4} is the difference between the profit from the route shown in Figure 3.3c and Figure 3.3a. Similarly, to determine the marginal profit of the subset {J5}, remove J4 from the route shown in Figure 3.3b to obtain the route shown in Figure 3.3d and then calculate the marginal profit of the obtained route. If the ratio of marginal profit and initial profit of a subset is higher than the minimum value ($\alpha$) set by the truck, that subset will be included in the bid.

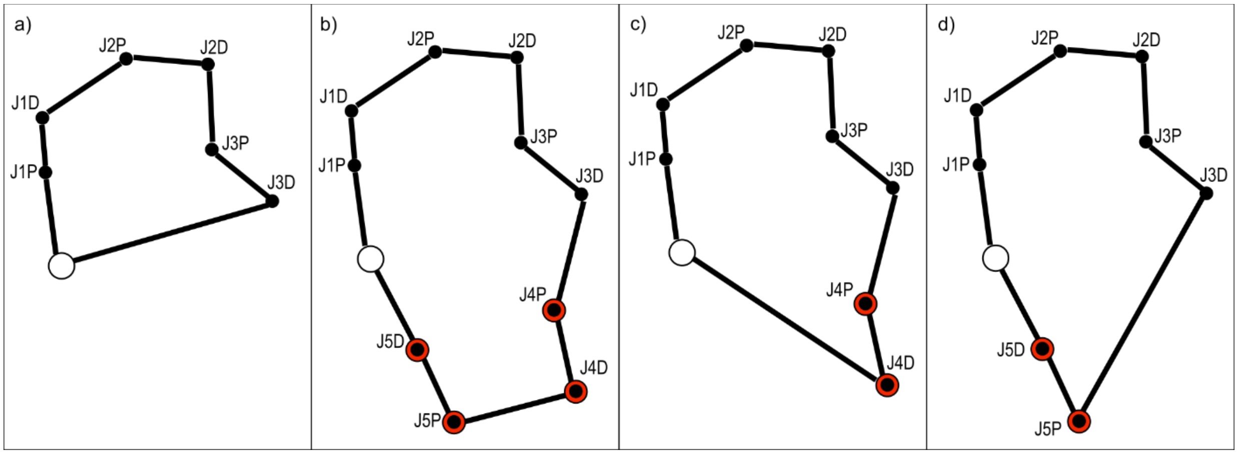


**Figure 0.3. Illustration of finding profitable subsets: (a) current truck route, (b) truck route with the most profitable bundle {J4, J5}, (c) truck route with subset {J4}, d) truck route with subset {J5}**

### *1.1.5. CC determines provisional winner*

In each round of the auction, the CC resolves any conflicts that exist in the bids submitted by the trucks and determines the provisional winner based on two objectives: 1) maximize the number of jobs served (primary objective), and 2) maximize the total increase in profit (secondary objective). The *conflict resolution model* is shown below.

*Decision variables*

$$\varepsilon_r^k = \begin{Bmatrix} 1 \text{ if truck } r \text{ wins bundle } k \\ 0 \text{ otherwise} \end{Bmatrix} r \in R, k \subseteq B$$

$$\mu_k^b = \begin{Bmatrix} 1 \text{ if job } b \text{ from bundle } k \text{ is served} \\ 0 \text{ otherwise} \end{Bmatrix} b \in B, k \subseteq B$$

*objective function*

$$Max\, Z = w_1 * \sum_{r \in R} \sum_{k \in K_r} |k| * \varepsilon_r^k + w_2 * \sum_{r \in R} \sum_{k \in K_r} \varepsilon_r^k * \delta_r^k \quad \textbf{(18)}$$

*subject to*

$$\sum_{k \in K_r} \varepsilon_r^k \le 1 \; \forall \; r \in R \quad \textbf{(19)}$$

$$\sum_{r \in R} \varepsilon_r^k \le 1 \; \forall \; k \subseteq B \quad \textbf{(20)}$$

$$\sum_{k \subseteq B} \mu_k^b \le 1 \; \forall \; b \in B \quad \textbf{(21)}$$

$$\varepsilon_r^k \le \mu_k^b \quad \forall \; r \in R, k \in K_r, b \in k \quad \textbf{(22)}$$

Note the bi-criteria objective function of the *conflict resolution model* (18) that seeks to maximize the number of jobs served and the total increase in profit of the participating trucks. In this model, it is assumed that the primary objective is the number of jobs served (first term in Equation 18), and the secondary objective is the total increase in the profit of the winning trucks (second term in Equation 18). A weighted aggregation (Ngatchou et al. 2005) method is adopted to solve the bi-criteria problem. In Equation

(18), $w_1$ is the weight of the primary objective and $w_2$ is the weight of the secondary objective. To ensure that the model selects the winners (trucks) such that the number of jobs served is maximized, it is assumed that $w_1 > w_2$. Constraints (19) ensure that a truck cannot win more than one bundle (among the bundles it submitted). Constraints (20) ensure that not more than one truck can win a bundle. Constraints (21) ensure that a job cannot be served more than once. Constraints (22) define the relationship between $\mu$ and $\epsilon$.

#### 1.1.6. *CC adjusts the outsourcing price*

The iterative CA is stopped when any of the following criteria is met: 1) all outsourced jobs are served, or 2) the solution does not improve for a given number of iterations. If the stopping criterion is not satisfied, CC adjusts the outsourcing price of the jobs. Suppose a job is not included in the bid submitted by any of the trucks. In that case, the outsourcing price of that job is increased based on Equation 23. On the other hand, if all outsourced jobs are present in at least one of the bundles submitted by the trucks, but all jobs are not served (note that the iteration stops if all jobs are served), the outsourcing price of the conflicting jobs is decreased based on Equation 23. If a job is present in three trucks' most profitable bundle, then the term $\sum_{r\in R} \varepsilon_r^{k*} * \mu_{k*}^b$ is equal to three. Thus, according to Equation (23), the outsourcing price of that job is decreased by twice of the step size, $t_w$.

$$OP_b^{w+1} = OP_b^w - t_w\left(\sum_{r\in R} \varepsilon_r^{k*} * \mu_{k*}^b - 1\right) \quad \forall \ b \in B \tag{23}$$

Where $OP_b^{w+1}$ is the outsourcing price of job $b$ in the $(w+1)^{th}$ iteration, $t_w$ is the step size at the $w^{th}$ iteration and $k^*$ is the most profitable bundle of a truck. Equation 23 is adopted from the work of Dai et. al. (2014). They proposed three methods to set up the step size in each round of the auction. This study adopts their first method that is based on Walrasian tatonnement theory. Initially, $t_w$ is set to a constant number $t_0$. After a given number of iterations, $t_w$ is halved, i.e., $t_{w+1}= t_w/2$. This step size will be halved again after this same number of iterations has been executed.

#### 1.1.7. *Implement the final decision*

The iterative process stops if either one of the two stopping criteria (mentioned above) is satisfied. After the iteration is stopped, the model checks whether bidders have selected all jobs to serve or not. If all jobs have been selected, the CC transfers the jobs and outsourcing prices to the winning truck(s). If they are not, the jobs will be returned to the respective carriers of the trucks that outsourced the jobs; the carrier has the ultimate responsibility of completing the jobs given by its customers. As done in previous indirect decentralized collaboration studies, it is assumed that the winning trucks get all the profit from the jobs they served.

## 3.4 Solution method

The *dynamic bundle selection model* belongs to the class of Nondeterministic Polynomial-time Complete (NPC) problems because this model is an extension of the vehicle routing problem with pickup and delivery (Solomon 1987). "Thus, all known algorithms that yield an optimal solution require exponentially increasing computational time as the number of jobs increases (Padmanabhan et al., 2023)”. Therefore, a solution methodology based on the LNS heuristic is used to solve the model. The underlying principle of the LNS heuristic is to remove some jobs from the current solution and then reinsert them in a better position to improve the objective function value (Padmanabhan et al., 2023, Shaw 1998 ). This process is repeated until the stopping criterion is met; the stopping criterion used in this study is the maximum number of iterations. The LNS heuristic developed by Padmanabhan et al. (2023) is adapted in this study with some modifications. The modifications are needed because the problem to be solved in this study is different from theirs. In their problem, a carrier receives jobs from its customers and then finds optimal trucks route to serve these jobs. However, in this study, an enroute truck should select the most profitable bundle of jobs from the list of outsourced jobs by finding the optimal route to serve both the current jobs and the jobs in the most profitable bundle.

The LNS algorithm is provided in Figure 3.4. The inputs required for the modified LNS heuristic are: an initial solution, the number of jobs to be removed (*q*) from the current solution, initial temperature, and cooling rate. LNS heuristic consists of a job removal heuristic and a job insertion heuristic. The removal heuristic, the insertion heuristics, and the criteria to select the number of jobs to be removed (*q*) are the same as that of the previously developed LNS heuristic (Padmanabhan et al. 2020, 2023). How the initial solution is determined is explained in the next paragraph.

```
Input: Initial solution, Number of jobs to be removed, Initial temperature, Cooling rate
Current solution, s_cur = Initial solution
Best solution, s_best = Initial solution
T = Initial temperature
α = Cooling rate
q = Number of jobs to be removed
Q = {1, …, q}
while the stopping criteria is not met do
        s = s_cur
        remove q jobs from s
        reinsert the removed jobs q into s
        if f(s) ≥ f(s_best) do
                s_best = s
            s_cur = s
                else if f(s) ≥ f(s_cur) do
                s_cur = s
                else
                        Generate a random value r from the uniform distribution in the range (0, 1)
                        Probability to accept solution s, p = exp (-( f(s_cur) - f(s) )/T)
            if p > r do
                s_cur = s
                T = T*α
            end
        end
end
return s_best
```

**Figure 0.4. Pseudocode of LNS heuristic with simulated annealing acceptance criteria (Padmanabhan et al., 2023)**

The modifications made to the LNS heuristic are explained using an illustrative example shown in Figure 3.5. In this example, there is a truck enroute called truck 1 and it is trying to select its most profitable bundle from the set of outsourced jobs. Truck 1 has two current jobs: job 1 with pickup location at J1P and delivery location at J1D and job 2 with pickup location at J2P and delivery location at J2D. There are two outsourced jobs: job 3 with pickup location at J3P and delivery location at J3D and job 4 with pickup location at J4P and delivery location at J4D. It is assumed that at time *t*, truck 1 just departed from its depot. In the modified version, the initial solution consists of two separate routes: 1) a truck's current route and 2) a dummy route that consists of all outsourced jobs. The initial dummy route is constructed by assuming that the truck starts and ends at its depot and serves all outsourced jobs such that an outsourced job's delivery location is visited immediately after visiting its pickup location. Also, the dummy route is formed without considering any other constraints, such as the time window, hours of service, etc. In this example, the dummy route is *Depot- J3P- J3D - J4P - J4D- Depot*. When applying the removal heuristics, jobs can be removed either from truck 1's route or from the dummy route. In iteration 1, job 1 is removed from truck 1's route and job 3 is removed from the dummy route. The jobs removed from truck 1's route must be reinserted back into the same route during the reinserting step. However, the jobs removed from the dummy route can be reinserted into truck 1's route if it is profitable; otherwise, they will reinsert back into the dummy route. In this example, job 3 is not profitable for truck 1, thus it goes back to the dummy route. Note that all the constraints such as time window, hours of service, etc. must be satisfied for a job to enter into truck 1's route. In iteration n, job 1 is removed from truck 1's route and job 4 is removed from the dummy route. In this example, job 4 is profitable to serve by truck 1, thus, it is inserted into truck 1's route. If the solution does not improve when the iteration stops, this solution is considered as the best solution and the most profitable bundle is the bundle with job 4.

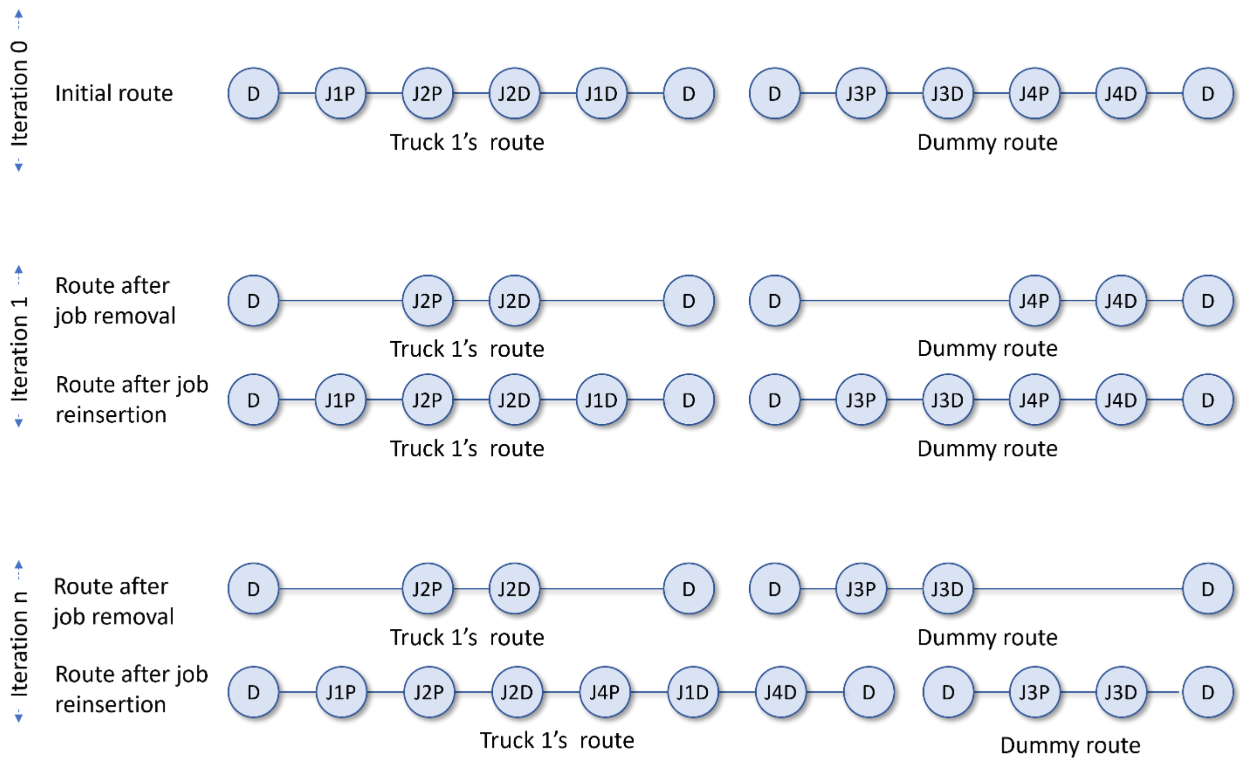


**Figure 0.5. Job removal and reinsertion process in the modified LNS heuristic**

# CHAPTER 4

## Computational Study

Six numerical experiments are used to demonstrate some of the insights that can be gleaned from this framework. The first experiment aims to establish that the modified LNS heuristic provides the correct solution for the *dynamic bundle selection model*. The second experiment shows the difference between MBB and SBB in regard to how bids are submitted and resulting job allocation by the CC and increase in profit for the participating carriers. The third experiment assesses the performance of MBB with different numbers of outsourced jobs in a period. The fourth experiment analyzes how MBB performs if all trucks are not available to collaborate. The fifth experiment assesses the effect of the value of the minimum percentage of increase in profit ($\alpha$) from a bundle on the solution from MBB. The sixth experiment analyzes the effect of the network size on the performance of MBB. The results from experiments 3, 4, 5, and 6 are also used to compare the performance of the MBB against SBB. In addition, the efficiency of MBB is also analyzed using the results from these four experiments. If all outsourced jobs are selected by the trucks through the bidding process, the bidding process is deemed 100% efficient. To quantify this, efficiency is defined to be the number of outsourced jobs served at the end of the auction divided by the total number of jobs that were outsourced. The proposed CA approach and the LNS heuristic were coded in Julia (version 1.8.3). To solve the *dynamic bundle selection model* and the *conflict resolution model,* the Gurobi solver (version 9.0.3) and JuMP (version 1.3.0) were used. The experiments were conducted on a laptop computer equipped with a 3 GHz processor and 32 GB of RAM.

There are no benchmark instances available for the proposed problem; hence, all six experiments were performed using instances generated for this study. All test instances were randomly generated from a set of 446 nodes that represent 446 actual cities and towns in South Carolina. The actual distance and travel time between any two locations were obtained from nextbillion.ai, an industry-leading spatial data and artificial intelligence platform (*AI-Powered Mapping Platform for Enterprises - Map APIs, SDKs & Map Tools | NextBillion.Ai*, n.d.). The *cost* to traverse the arc between any two nodes was assumed to be equal to the actual distance between them. The locations of carrier depots were manually selected from the 446 nodes and were chosen in such a way that they were located a significant distance from each other. The depots were fixed for all instances. The total number of nodes in a network is the sum of carrier depots and customer locations. If there are *m* jobs, then the total number of customer locations will be equal to *2*m* because each job consists of a pickup location and a delivery location.

For the first experiment, 9 networks were considered that vary in size from 19 nodes to 35 nodes. Only one instance of customer nodes was generated (recall that they are selected randomly) for each network size since the goal was to compare the solutions from LNS heuristics to the exact solution. For the second experiment, a 44-node network was considered. Similar to experiment 1, only one instance of customer nodes was generated for the 44-node network since the goal was to compare the MBB and SBB.

For the third, fourth, and fifth experiments, a network with 4 carriers and 60 jobs (each carrier received 15 jobs from its customers) was considered and 10 instances of this network were generated. In the third experiment, all 10 instances were tested with nine different numbers of outsourced jobs. The different number of outsourced jobs considered were 4, 5, 6, 7, 8, 9, 10, 15, and 20; to thoroughly observe the trend, four to ten outsourced jobs were considered, and to ensure that the trend continues for larger numbers, 15 and 20 outsourced jobs are considered. Experiment 2 was limited to no more than 20 outsourced jobs because it seems unlikely that a truck would outsource a significant number of jobs in a 15-minute window. Therefore, it is assumed the maximum is 33.3% (33.3% of 60 is 20). For the third and fourth experiments, it was assumed that the number of outsourced jobs is 6.

In the fourth experiment, it was assumed that all trucks may not be available to collaborate in all time periods. Thus, this experiment evaluates how MBB performs in such situations. Each instance was tested with four different percentages of available trucks: 25%, 50%, 75% and 100%. In the fifth experiment, the performance of MBB was evaluated under a scenario where each truck has different values for the percentage of minimum increase in profit ($\alpha$) from a bundle. It is not practical to evaluate all possible combinations of trucks' $\alpha$ values. Thus, for each instance, a preliminary experiment was conducted with all trucks' $\alpha$ set to 5%. The truck that obtained the highest increase in profit in the preliminary experiment is referred to as *selected truck* and all other trucks are referred to as *other trucks*. Five different levels were considered, with each level representing a different $\alpha$ value for the *other trucks*: level 1 with 5%, level 2 with 10%, level 3 with 15%, level 4 with 20%, and level 5 with 25%. In each level, five different $\alpha$ values are considered for the *selected truck*: 5%, 10%, 15%, 20%, and 25%. The fifth experiment enables the observation of the change in the *selected truck's* profit and number of jobs served with respect to the change in the $\alpha$ values of both the *selected truck* and the *other trucks*. In the sixth experiment, four different network sizes were considered: 1) 62 nodes, 2 carriers and 30 jobs; 2) 93 nodes, 3 carriers and 45 jobs; 3) 124 nodes, 4 carriers and 60 jobs; and 4) 155 nodes, 5 carriers and 75 jobs. Note that each carrier received 15 jobs from its customers in all four network sizes. It was assumed that the number of outsourced jobs is proportional to the total number of jobs (approximately 14%).

All six experiments were conducted for a single period (the first period, between 0 and 15 minutes) because the main goal of the experiments is to test the efficiency of the proposed approach and compare it against SBB. For the price adjustment of outsourced jobs, to increase the price, the initial time step $t_m$ was assumed to be 50, and to decrease the price, the initial time step $t_m$ was assumed to be 25. In both cases, the step size was halved after every five rounds of auction. The iterative process was stopped if the solution did not improve in four consecutive rounds. The reason for selecting four consecutive rounds as a stopping criterion is twofold: 1) from preliminary experiments, it was found that if the solution does not improve in four consecutive rounds, then it is highly unlikely that a solution will improve in the subsequent rounds, and 2) a solution must be obtained as quickly as possible since it is a dynamic problem (specifically, within 14 minutes because the duration of a single period is assumed to be 15 minutes). The maximum outsourcing price of a job is assumed to be the actual price paid by the customers to serve the job. The weights in Equation (18), $w_1$ and $w_2$ were assumed to be 0.999 and 0.001 respectively.

The assumptions and parameter values used in all six experiments are as follows. All jobs consist of transporting 50 units of goods from their origin to the destination and the capacity of all vehicles is 300 units. The time window and the amount paid by the customer are determined exactly as in the work of Padmanabhan et. al. (2023) and it is as follows. “The earliest arrival time at *pickup* locations was randomly generated from a uniform distribution in the range of [0, 8] hours, where 0 corresponds to 12 AM. The earliest arrival time at the *delivery* locations was calculated by adding a random number between 3 and 4 hours to the earliest arrival time at the job's pickup location. The latest departure time at the *pickup* locations was randomly generated from a uniform distribution in the range of [8, 16] hours. The latest departure time at the *delivery* locations was calculated by adding a random number between 3 and 4 hours to the latest departure time at the job's pickup location. Preliminary experiments were conducted to ensure that there are feasible schedules/routes with the specified time windows. It was assumed that there is no restriction on the time when a truck must return to the depot (Padmanabhan et al., 2023).”

“The amount paid by a customer to a carrier for fulfilling a job was calculated via Equation (24).

**Price to be paid by a customer to the carrier for fulfilling a job =** $\zeta_1 + \zeta_2 * l$ **(24)**

Where $\zeta_1$ is the fixed price, $\zeta_2$ is the variable price and $l$ is the distance between the pickup and delivery location. The values of $\zeta_1$ and $\zeta_2$ were obtained by collecting the price to serve the pickup and delivery jobs from 13 LTL carriers in South Carolina. The fixed price, $\zeta_1$ is the average price for pickup and delivery of the goods, and the variable price, $\zeta_2$ includes fuel surcharge and line haul charges. The average fixed price and variable price from the 13 LTL carriers were used to conduct the experiments; that is $\zeta_1 = 100$ and $\zeta_2 = 2.5$ (Padmanabhan et al., 2023).”

# CHAPTER 5

# Results and Discussion

## 5.1 Experiment 1: Evaluation of LNS heuristic compared to Gurobi solver

Table 5.1 shows the results of experiment 1. Column 1 shows the network size. Columns 2 and 3 show the total profit obtained and computation time taken by the Gurobi solver, respectively. Columns 4 and 5 show the total profit obtained and computation time taken by the LNS heuristic, respectively. The last column shows the percentage of deviation between the Gurobi solver solution and LNS heuristic solution. When the LNS heuristic obtained a solution equal to the optimal solution obtained from the Gurobi solver, the objective function value is reported with an asterisk. When the Gurobi Solver is unable to obtain the optimal solution within 24 hours, the best solution (incumbent solution) is reported with an 'i' next to it. For networks with up to 27 nodes, Gurobi found an optimal solution that was replicated by the LNS heuristic developed for the *dynamic bundle selection model*. For networks with 29, 31, 33, and 35 nodes, Gurobi did not find an optimal solution within 24 hours and the LNS heuristic yielded an equal or superior solution than the incumbent solution. The results shown in Table 5.1 demonstrate that the LNS heuristic provides an equivalent or superior solution to the Gurobi solver in all experiments with significantly less computational time.

**Table 0.1.Evaluation of the performance of LNS heuristic for the *dynamic bundle selection model* against Gurobi Solver**

| Network size (# of nodes) | Gurobi Solver (24-hours time limit) | | LNS heuristic (1000 iterations) | | % deviation |
|---|---|---|---|---|---|
| | Profit | Time (minutes) | Profit | Time (minutes) | |
| 19 | 384.77 | 12 | 384.77* | 1.5 | 0 |
| 21 | 556.02 | 25 | 556.02* | 1.7 | 0 |
| 23 | 424.92 | 40 | 424.92* | 1.7 | 0 |
| 25 | 507.03 | 192 | 507.03* | 1.7 | 0 |
| 27 | 552.10 | 1020 | 552.10* | 1.8 | 0 |
| 29 | 1915.23[i] | 1440 | 2002.36 | 1.8 | -4.4 |
| 31 | 1542.38[i] | 1440 | 1542.38 | 2.0 | 0 |
| 33 | 1809.33[i] | 1440 | 1809.33 | 2.1 | 0 |
| 35 | 1881.74[i] | 1440 | 2178.99 | 2.1 | -13.6 |

[i] incumbent solution, N/A - No feasible solution is found within the specified time limit, * LNS solution = Gurobi optimal solution
% deviation = (Gurobi solution – LNS solution)*100/ LNS solution

## 5.2 Experiment 2: Comparison of MBB and SBB performance

Tables 5.2 and 5.3 show the results from a network with 44 nodes and 4 participating carriers with each carrier having 5 jobs each using MBB and SBB, respectively. In this experiment, there are three outsourced jobs. Column 1 shows the trucks of participating carriers; truck 1 belongs to carrier 1, truck 2 belongs to carrier 2, truck 3 belongs to carrier 3, and truck 4 belongs to carrier 4. The number of trucks per carrier and the number of jobs per truck are determined using the models described in section 3.3.1. Column 2 shows the bundle(s) submitted by the trucks. Column 3 shows the provisional winners and the selected bundles. The last column shows the sum of the increase in profit of all participating trucks. In both approaches, each truck determined the most profitable bundle. However, in MBB, each truck also determined some profitable subsets (based on the minimum percentage of increase in profit) of the most profitable bundle and bid for multiple bundles (see column 3 of Table 5.3). For both approaches, all three jobs were selected to serve in the first round of the auction. However, MBB solution had a higher profit than SBB because it could piece together smaller bundles of jobs to get more profit and still be feasible.

**Table 0.2. Results from a 44-node network using SBB CA (first round of auction)**

| Trucks | Bids submitted | Winner | Increase in profit |
|---|---|---|---|
| Truck 1 | [25, 26, 24] | [25, 26, 24] | |
| Truck 2 | [25, 26, 24] | | 362.32 |
| Truck 3 | [24, 26] | | |
| Truck 4 | [24, 26, 25] | | |

**Table 0.3. Results from a 44-node network using MBB CA (first round of auction)**

| Trucks | Bids submitted | Winner | Increase in profit |
|---|---|---|---|
| Truck 1 | [25, 26, 24], [24, 26], [24, 25], [25, 26] | [24, 26] | |
| Truck 2 | [25, 26, 24], [24, 26], [24, 25], [25, 26] | | 380.4 |
| Truck 3 | [24, 26], [26], [24] | | |
| Truck 4 | [24, 26, 25], [25, 26], [24, 25], [24, 26], [25] | [25] | |

## 5.3 Experiment 3: Effect of numbers of outsourced jobs on performance of MBB

Figure 5.1 shows the results from experiment 3, and it can be seen that MBB provided better results in less time than SBB, regardless of the number of outsourced jobs. Figure 5.1a shows that MBB served a higher number of jobs than SBB, for all instances. The difference between the number of jobs served from MBB and SBB increases as the number of outsourced jobs increased (see inset of Figure 5.1a). These results show that MBB is more capable of finding a higher number of jobs served than SBB when there are a large number of outsourced jobs. There are two possible explanations for the increase. First, there are more conflicting bids with SBB, making the conflict resolution even more difficult. Second, the cases where a truck submits a bid that consists of all outsourced jobs are very rare unlike when there are a small number of outsourced jobs. If a truck bids for all outsourced jobs, there is at least one solution without conflict in SBB and this will not occur when there are a large number of outsourced jobs.

Figure 5.1b shows that MBB used fewer auction rounds than SBB. In MBB, the number of auction rounds is less than 1.2 up to 8 outsourced jobs. Beyond that, it increased gradually because the number of bundles in the bids submitted also increases as the number of outsourced jobs increases. More specifically, MBB efficiently resolved the conflict in less than three rounds of auction, even when more than 30% of the total jobs (the case with 20 outsourced jobs) were outsourced. In SBB, the number of auction rounds increased rapidly, up to 8 outsourced jobs. Beyond that, it increased gradually. The reason for the rapid increase is that similar to MBB, the number of bundles in the bids submitted also increases as the number of outsourced jobs increases. However, SBB took more iterations compared to MBB as its conflict resolution is not as efficient as MBB. It is also observed that when the number of outsourced jobs is 8 or less, in most of the instances, SBB resolved the conflict after some auction rounds; average auction round is 1.3 for 4 outsourced jobs and 5.2 for 8 outsourced jobs. However, when the number of outsourced jobs is 10 or higher, the iterative process is stopped using the second stopping criterion: stop the iteration if the solution does not improve in four consecutive iterations. That means the iterative process will not continue until the conflict is completely resolved. Thus, the number of auction rounds to get a collaboration decision is comparatively less. This is the reason for the gradual increase in the number of auction rounds when there are 10 or more number of outsourced jobs. Note that when the iteration stops using the second criterion in SBB, some jobs will be unserved.

Figure 5.1c shows that MBB reached the solution in less time than SBB. In MBB, the computation time increases gradually as the number of outsourced jobs increased. This is because the number of nodes a truck should consider while solving the *bundle selection model* increases proportionally as the number of outsourced jobs increased. However, in SBB, the computation time increases rapidly as the number of outsourced jobs increased. This is because in addition to the increase in the number of nodes (as mentioned above), the number of auction rounds (discussed previously for Figure 5.1b) also increased, which further increases the computation time. The difference between auction rounds from MBB and SBB, and the difference between the computation time from MBB and SBB are shown in the inset of figures 5.1b and 5.1c, respectively.

Figure 5.1d shows that MBB provided a higher increase in profit than SBB. The increase in profit from MBB increases exponentially as the number of outsourced jobs increased. This is because more profit will be obtained if there are more jobs. In some instances, MBB provided a higher increase in profit even though the number of jobs served was the same as SBB. The reasons are twofold. First, MBB determines the winners such that the jobs are served using the least cost routes compared to SBB. Second, in SBB, to resolve the conflict, the outsourcing price of the conflicting jobs gets reduced after each auction round, which reduces the profit of the trucks serving them. However, in MBB, the outsourcing price of a job only gets reduced if there is a situation where all jobs are included in the bid, but not all jobs are served. Thus, the trucks get a lower profit increase using SBB than MBB even if they serve the same number of jobs because of the lower outsourcing price in SBB in the later rounds of the auction. The difference between the increase in profit from MBB and SBB is likely to increase as the number of outsourced jobs is increased based on results shown in the inset of Figure 5.1d.

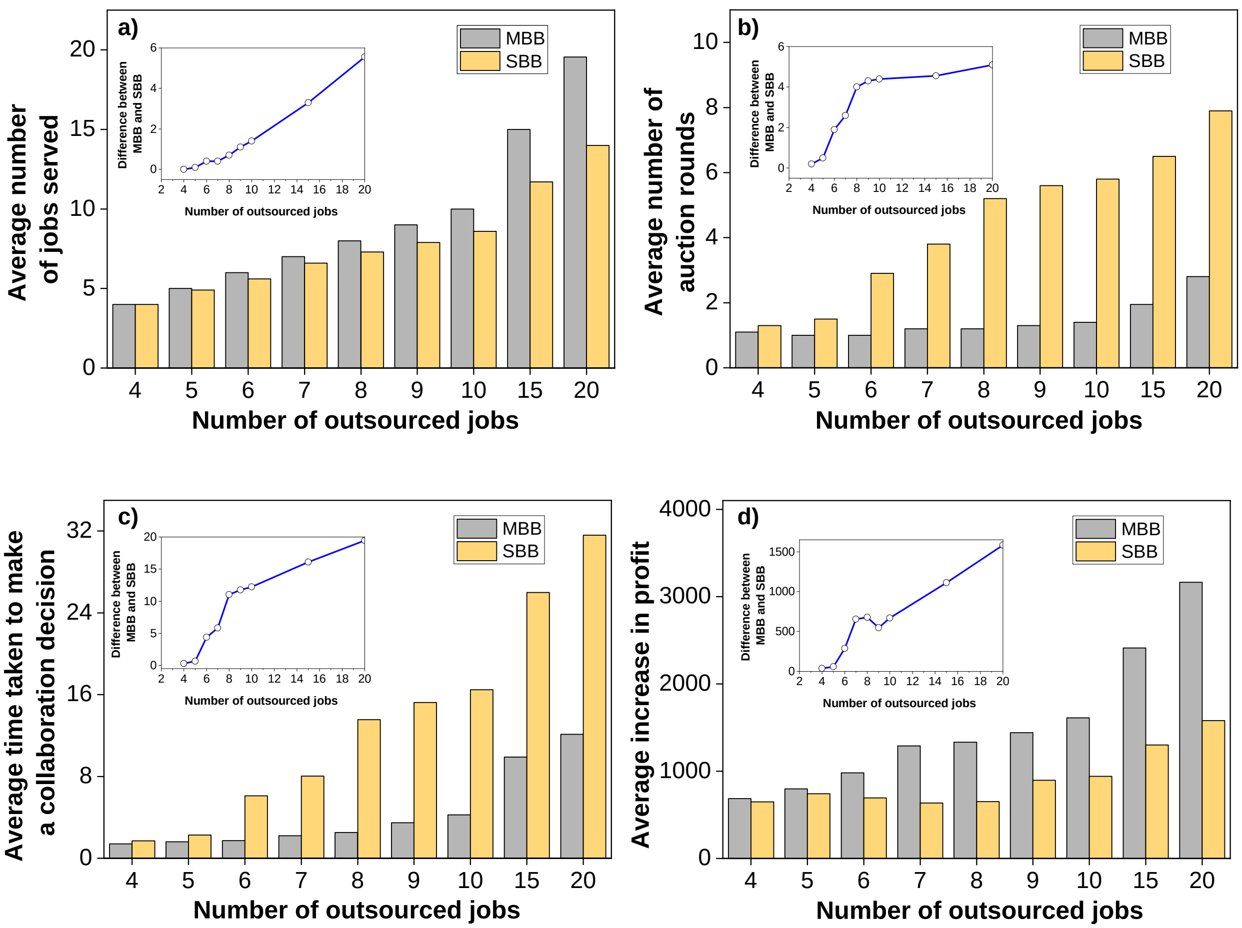


**Figure 0.1. Comparison of MBB and SBB for the different numbers of outsourced jobs a) number of jobs served, b) the number of iterations, c) time taken to make a collaboration decision, and d) average increase in profit.**

## 5.4 Experiment 4: Effect of different percentages of trucks available on the performance of MBB

Figure 5.2 shows the results from experiment 4 which indicates that MBB always outperforms SBB even when only a portion of the trucks are available to collaborate. Recall that MBB outperforms SBB when all trucks are available (section 6.3).

In MBB, if at least 50% of the trucks are available, the system performs nearly as well as when all trucks are available. All jobs are served in all ten instances when the percentages of trucks available are 50, 75, and 100, as shown in Figure 5.2a. When 100% trucks were available, it took only one auction round to get all jobs served. However, the number of auction rounds and the computation time are slightly higher when the percentages of trucks available are 50 or 75 (see Figures 5.2b and 5.2c). In addition, the increase in profit increases as the percentage of trucks available increased (see Figure 5.2d). Unlike the instances when more than 50% of the trucks are available, when only 25% of the trucks are available the average number of jobs served is less, and the number of auction rounds and computation time are higher.

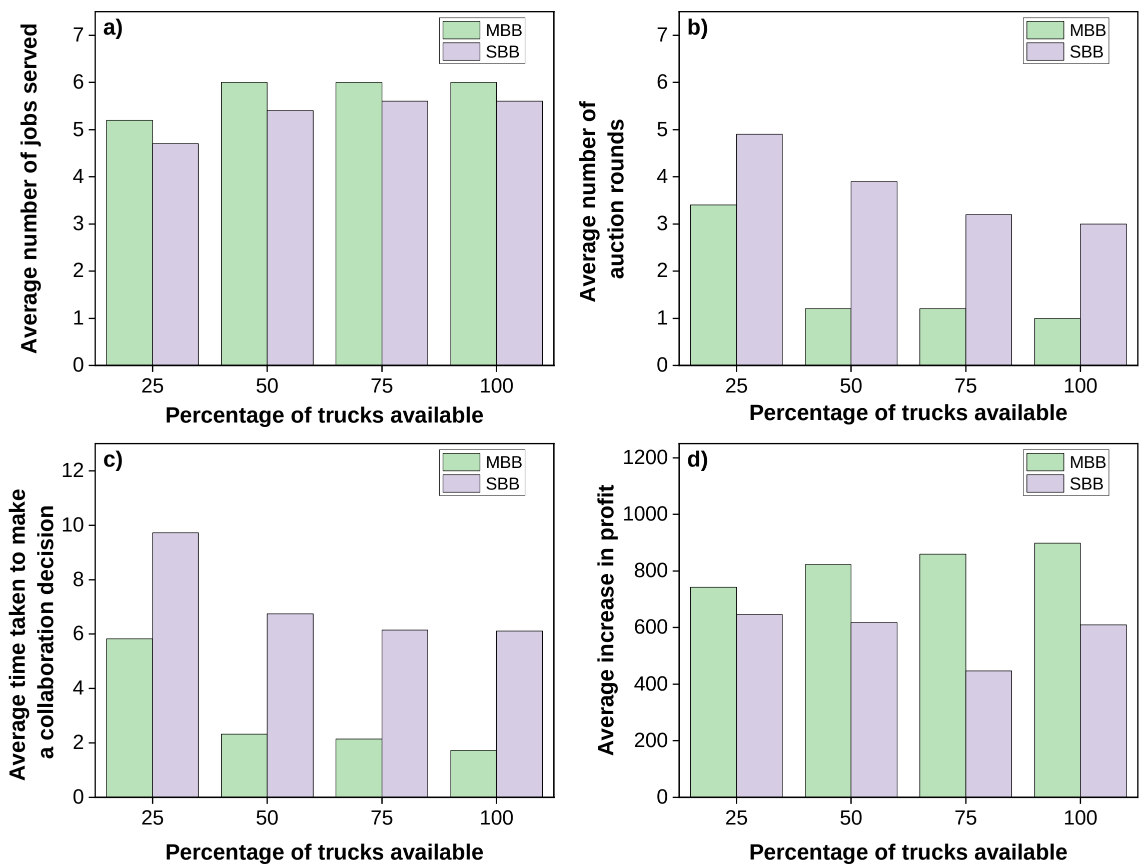


**Figure 0.2. Comparison of MBB and SBB for the different percentages of trucks available a) number of jobs served, b) number of iterations, c) time taken to make a collaboration decision, and d) average increase in profit.**

In SBB, as the percentage of available trucks decreases, the average number of jobs served decreased, and the average rounds of auction and computation time increased. The average increase in profit is lowest when the availability of trucks is 75%. The reason for this can be understood through an example shown in Table 5.4. In Table 5.4, when the percentage of available trucks is 75, all jobs are served after three auction rounds. Thus, the best solution is from auction round 3; 6 jobs are served with an increase in profit of 273.43. When the percentage of trucks available is 50% or 25%, only four jobs are served even after five auction rounds. Thus, the best solution is from auction round one; 4 jobs are served with 679.61. The profit increase is considerably less when 75% of the trucks are available compared to 50 or 25%, but more jobs are served. Recall that the primary objective of this study is to serve more jobs. In six out of ten instances, when 75% of the trucks are available, SBB resolved the conflict after some rounds of auction, and thus, all jobs are served but with less increase in profit, as discussed above. This is the reason for the dip in the increase in profit when the availability of trucks is 75%.

**Table 0.4. An example showing the difference in the increase in profit as the percentage of trucks available changes.**

| Percentage of the trucks available | Iteration number | Number of jobs served (current) | Number of jobs served (best) | Increase in profit |
|---|---|---|---|---|
| **100** | 1 | 6 | 6 | 982.56* |
| **75** | 1 | 5 | 5 | 729.99 |
| | 2 | 5 | 5 | 404.99 |
| | 3 | 6 | 6 | 273.43* |
| **50** | 1 | 4 | 4 | 679.61* |
| | 2 | 4 | 4 | 504.61 |
| | 3 | 4 | 4 | 354.61 |
| | 4 | 4 | 4 | 261.58 |
| | 5 | 4 | 4 | 249.99 |
| **25** | 1 | 4 | 4 | 679.61* |
| | 2 | 4 | 4 | 661.584 |
| | 3 | 4 | 4 | 611.584 |
| | 4 | 4 | 4 | 579.61 |
| | 5 | 4 | 4 | 550.92 |

## 5.5 Experiment 5: Effect of the percentage of minimum increase in profit of trucks on the performance of MBB

Figure 5.3 shows the results from experiment 5. Recall that in MBB, trucks select the subsets of the most profitable bundle to include in the bid based on their *α* value. If *α* is too low, more subsets of the most profitable bundle will qualify to be included in the bid; the number of subsets in the bid submitted decreases as the *α* value increases. Experiment 5 evaluated how the value of *α* affects individual carriers' increase in profit. When the selected truck' *α* is between 5 and 10, it has the highest increase in profit (see Figure 5.3b) and the highest number of jobs to serve (see Figure 5.3a) irrespective of the other carriers' *α* values. When *α* is greater than 10, the truck with the highest increase in profit and more jobs to serve lost its position as the winner. This trend is consistent irrespective of the other trucks' *α* values. In Figure 5.3b, the selected trucks' profit decreases as the other trucks' *α* values increased. This is because the set of bundles that each truck submits changes with the change in *α* values. These results show that the final collaboration results depend on each truck's *α* value, and it cannot be concluded that a particular *α* value will ensure winning the auction. However, if all trucks' *α* values are less than 20%, the process of resolving the conflicts is expedited because there are more bundle combinations in the bids submitted by the trucks.

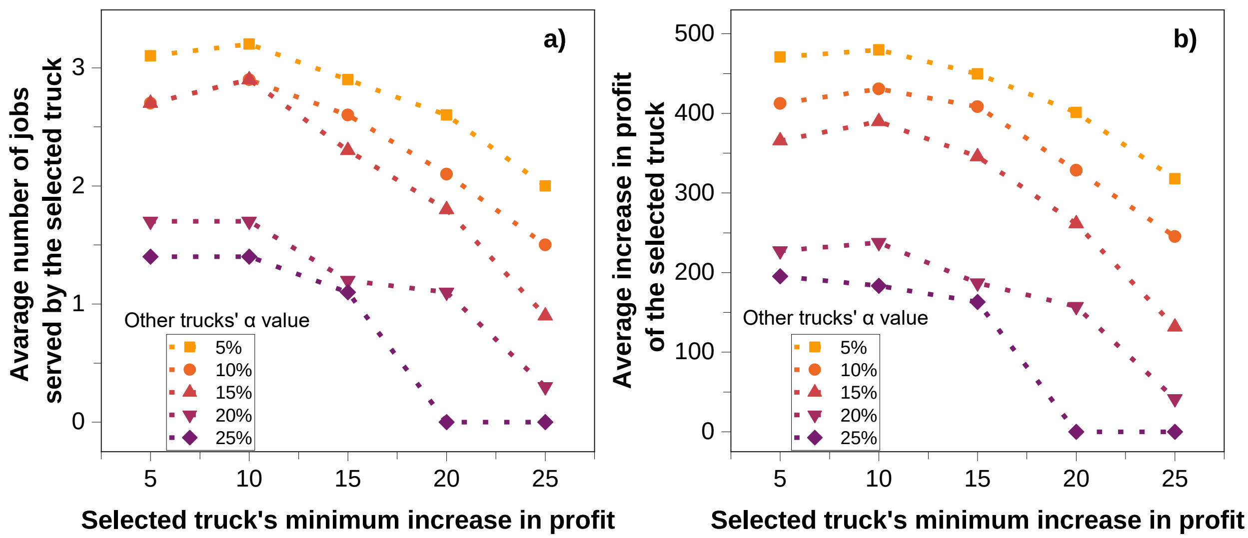


**Figure 0.3. Effect of minimum increase in profit on a) number of jobs served by the selected truck and b) increase in profit by the selected truck**

## 5.6 Experiment 6: Effect of network size on the performance of MBB

Figure 5.4 shows the results of experiment 6. In Figure 5.4a, using MBB, all jobs are served in all instances regardless of the network size. Whereas using SBB, in all network sizes considered, there were some instances in which all jobs are not served. In figure 5.4b, for all network sizes considered, the number of auction rounds is much less in MBB compared to SBB. In Figure 5.4c, when using MBB the time it takes to come to a collaboration decision increases gradually compared to a rapid increase with SBB. As explained in section 6.3, the reason is due to the increase in the number of nodes for MBB. For SBB, it is due to the increase in both the number of nodes and number of auction rounds. In Figure 5.4d, using MBB, the increase in profit increases exponentially whereas with SBB it increases gradually. The reason is the same as explained in section 6.3 (for Figure 5.4d). The graphs in the inset of Figure 5.4a to 5.4d show that the difference between the performance of MBB and SBB for all four attributes increases exponentially as the network size increases. A possible explanation for the significant increase is that the bids submitted by the trucks will be much larger as the network size increases. This makes the conflict resolution much more difficult for the single bundle approach compared to the multiple bundle approach. This in turn translates to a higher performance of MBB.

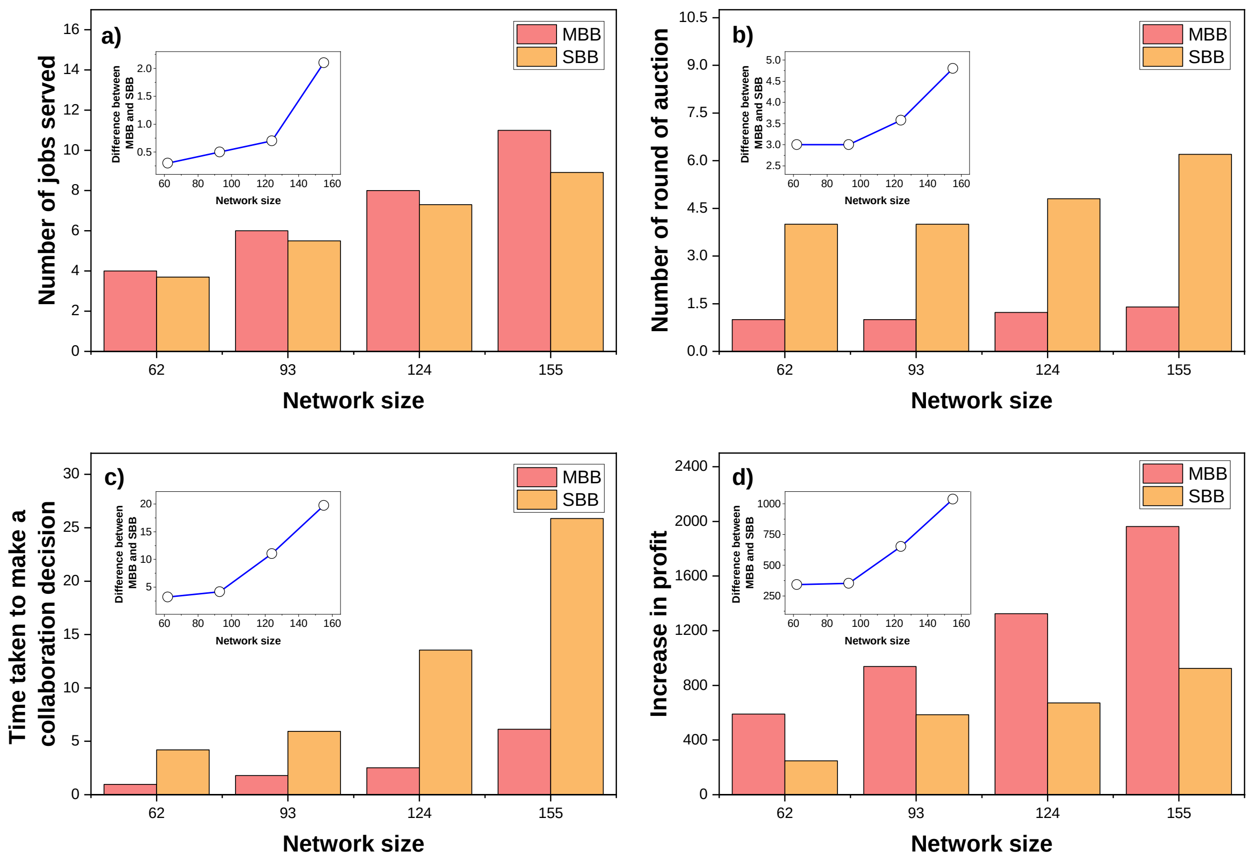


**Figure 0.4. Comparison of MBB and SBB for the different network sizes a) number of jobs served, b) number of iterations, c) time taken to make a collaboration decision, and d) average increase in profit.**

## 5.7 Effect of different factors on the efficiency of MBB

Results from experiments 3, 4, 5, and 6 are used to analyze the efficiency of the proposed indirect dynamic decentralized collaboration using MBB. From Figure 5.5, it can be seen that using MBB resulted in 100% efficiency for almost all instances. The three cases when 100% efficiency was not attained are shown in Figure 5.5a, 5.5b and 5.5c. In Figure 5.5a, 100% efficiency is not obtained when there are twenty outsourced jobs in an alliance with eight trucks; the trucks-to-outsourced jobs ratio is less than 0.4. The reason for not getting 100% efficiency in this case is that when there are fewer trucks and more outsourced jobs, there will be a large number of conflicting bids and resolving the conflict in all instances such that all jobs are served may not be practical. In Figure 5.5b, 100% efficiency is not obtained when only 25% of the total number of trucks are available to collaborate. Similar to the case in 9a, here the trucks-to-outsourced jobs ratio is less than 0.4, and thus, resolving the conflict in all instances such that all jobs are served may not be practical. In Figure 5.5c, 100% efficiency is not obtained when the minimum increase in profit from a bundle set by the trucks to bid for the bundle is too high (more than 20%). The reasons for not getting 100% efficiency in this case are: 1) some trucks did not submit a bid, and 2) some jobs were not included in the bid.

From Figure 5.5, it can be also seen that with SBB efficiency has a decreasing trend as the number of outsourced jobs increases, the percentage of trucks available decreases, and the network size increases. Thus, SBB's performance worsens as the complexity increases.

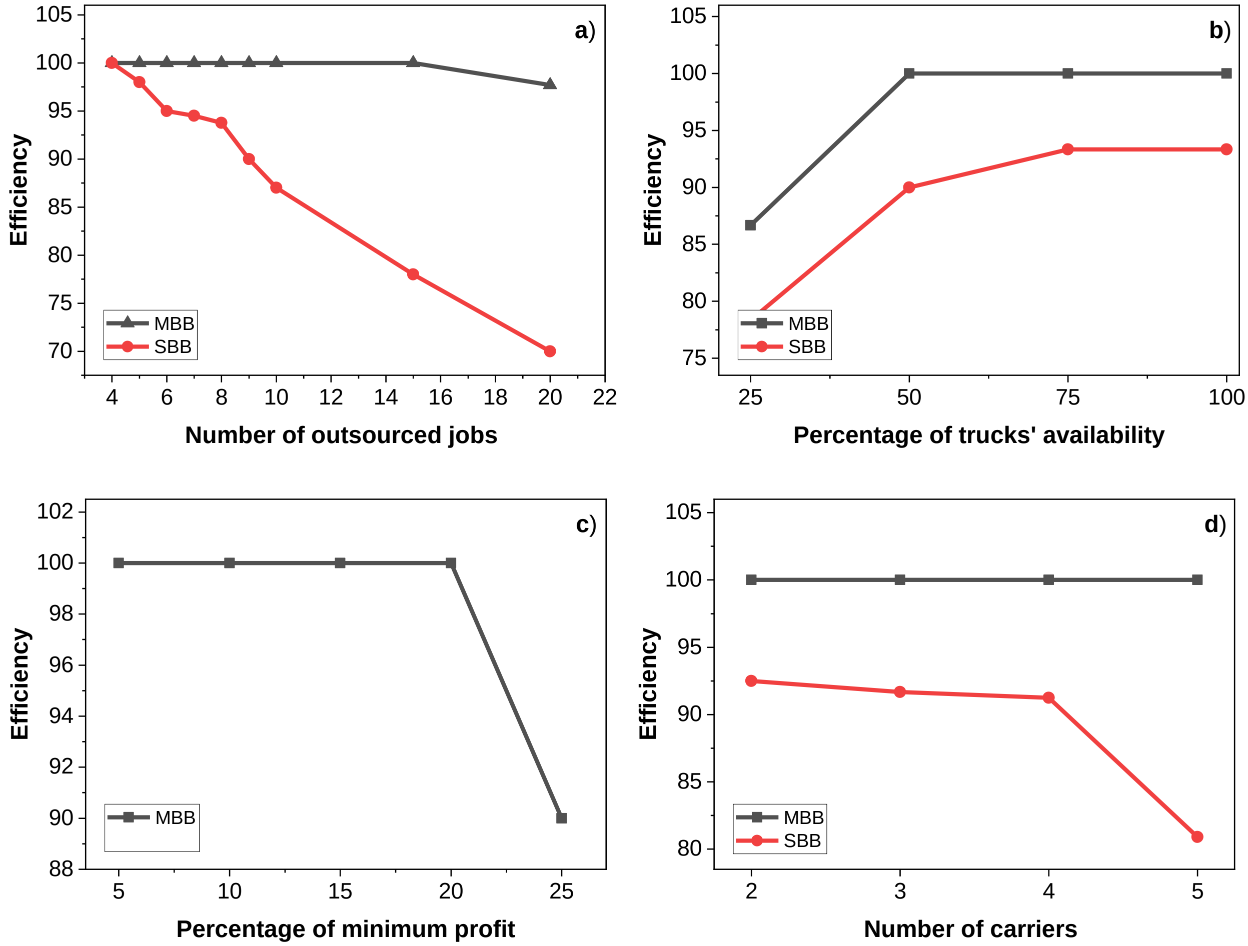


**Figure 0.5. Efficiency of MBB and SBB under a) number of jobs served, b) number of iterations, c) time taken to make a collaboration decision, and d) average increase in profit.**

# CHAPTER 6

## Summary and Conclusions

This study addressed a decentralized dynamic LTL pickup and delivery problem in which carriers collaborate indirectly due to the need to outsource jobs and/or the opportunity to bid for jobs to increase profit. Significant contributions of this study are: 1) a multi-bundle iterative CA framework is developed to solve the proposed problem, 2) a mixed integer linear programming model that can be used by each truck to select the most profitable bundle of jobs to bid in real-time is developed (*dynamic bundle selection model*), 3) a solution method based on LNS is developed to solve the *dynamic bundle selection model*, and 4) an integer programming model is developed (*conflict resolution model*) to resolve conflict when multiple trucks bid for the same jobs and to select the winning truck(s). Numerical experiments showed that the proposed MBB strategy in CA has the following benefits compared to SBB. First, all outsourced jobs are served except in some extreme situations, such as the trucks-to-outsourced jobs ratio being too low (less than 0.4) and the bidding requirement (the minimum increase in profit from a bundle to include in the bid) being more than a 20% increase in profit. On the other hand, if only the most profitable bundle is submitted, all jobs are served only when the trucks-to-outsourced jobs ratio is more than 2. Second, the number of auction rounds and the computation time required to make a collaboration decision is significantly less. Third, MBB determines the least cost routes compared to SBB, thus it provides a higher increase in profit for the participating carriers to serve the same sets of jobs. Lastly, for the participating trucks, considering profitable subsets of the most profitable bundle in the bid increases the chance of winning the auction as more bundle combinations are available. It was observed that for decentralized collaboration of LTL carriers, the CA approach with multiple bundle bidding and an exclusive conflict resolution model benefits more than having a CA approach with single bundle bidding and a conflict resolution just by reducing the outsourcing price of the jobs for the trucks to serve the same bundle of jobs. To allow for the investigation of much larger networks and greater number of carriers, trucks per carrier, and jobs per truck, future work should consider the development of a more efficient metaheuristic/heuristic to solve the *dynamic bundle selection model*.

## REFERENCES

AI-Powered Mapping Platform for Enterprises - Map APIs, SDKs & Map Tools | NextBillion.Ai." n.d. Accessed May 31, 2022. https://nextbillion.ai/.

Berger, Susanne, and Christian Bierwirth. 2010. "Solutions to the Request Reassignment Problem in Collaborative Carrier Networks." *Transportation Research Part E: Logistics and Transportation Review* 46 (5): 627–38.

Dai, Bo, and Haoxun Chen. 2011. "A Multi-Agent and Auction-Based Framework and Approach for Carrier Collaboration." *Logistics Research* 3 (2–3): 101–20.

Dai, Bo, Haoxun Chen, and Genke Yang. 2014. "Price-Setting Based Combinatorial Auction Approach for Carrier Collaboration with Pickup and Delivery Requests." *Operational Research* 14 (3): 361–86.

"Driving Reduced by 25% and Emissions by 90% through Combined City Deliveries in Antwerp." n.d. Accessed May 23, 2022. https://press.bpost.be/driving-reduced-by-25-and-emissions-by-90-through-combined-city-deliveries-in-antwerp.

Ferrell, William, Kimberly Ellis, Phil Kaminsky, and Chase Rainwater. 2019. "Horizontal Collaboration: Opportunities for Improved Logistics Planning." *International Journal of Production Research*, August, 1–18. https://doi.org/10.1080/00207543.2019.1651457.

Gansterer, Margaretha, and Richard F Hartl. 2016. "Request Evaluation Strategies for Carriers in Auction-Based Collaborations." *OR Spectrum* 38 (1): 3–23.

Gansterer, Margaretha, and Richard F Hartl. 2017. "Collaborative Vehicle Routing: A Survey." *European Journal of Operational Research*.

Gansterer, Margaretha, Richard F Hartl, and Martin Savelsbergh. 2020. "The Value of Information in Auction-Based Carrier Collaborations." *International Journal of Production Economics* 221: 107485.

Hernández, Salvador, and Srinivas Peeta. 2014. "A Carrier Collaboration Problem for Less-than-Truckload Carriers: Characteristics and Carrier Collaboration Model." *Transportmetrica A: Transport Science* 10 (4): 327–49.

Hernández, Salvador, Srinivas Peeta, and George Kalafatas. 2011. "A Less-than-Truckload Carrier Collaboration Planning Problem under Dynamic Capacities." *Transportation Research Part E: Logistics and Transportation Review* 47 (6): 933–46.

Huynh, N., Ferrell, W., Padmanabhan, B., & Badyal, V. (n.d.). "Tool to Assess Effectiveness of Intermodal Facility Location and Carrier Collaboration (U. of S. Carolina & C. University (Eds.))." *https://rosap.ntl.bts.gov/view/dot/60928*

Klaus, P. 2003. "Die ''TOP 100" Der Logistik: Berichtszeitraum 2001/2002."

Krajewska, Marta Anna, Herbert Kopfer, Gilbert Laporte, Stefan Ropke, and Georges Zaccour. 2008. "Horizontal Cooperation among Freight Carriers: Request Allocation and Profit Sharing." *Journal of the Operational Research Society* 59 (11): 1483–91.

Leskelä, Riikka Leena, Jeffrey Teich, Hannele Wallenius, and Jyrki Wallenius. 2007. "Decision Support for Multi-Unit Combinatorial Bundle Auctions." *Decision Support Systems* 43 (2): 420–34. https://doi.org/10.1016/J.DSS.2006.10.009.

Li, Junsong, Gang Rong, and Yiping Feng. 2015. "Request Selection and Exchange Approach for Carrier Collaboration Based on Auction of a Single Request." *Transportation Research Part E: Logistics and Transportation Review* 84: 23–39.

Li, Yuan, Haoxun Chen, and Christian Prins. 2016. "Adaptive Large Neighborhood Search for the Pickup and Delivery Problem with Time Windows, Profits, and Reserved Requests." *European Journal of Operational Research* 252 (1): 27–38.

Los, Johan, Frederik Schulte, Matthijs T J Spaan, and Rudy R Negenborn. 2020. "The Value of Information Sharing for Platform-Based Collaborative Vehicle Routing." *Transportation Research Part E: Logistics and Transportation Review* 141: 102011.

Lyu, Xiaohui, Haoxun Chen, Nengmin Wang, and Zhen Yang. 2019. "A Multi-Round Exchange Mechanism for Carrier Collaboration in Less than Truckload Transportation." *Transportation Research Part E: Logistics and Transportation Review* 129: 38–59.

Ngatchou, Patrick, Anahita Zarei, and A El-Sharkawi. 2005. "Pareto Multi Objective Optimization." In *Proceedings of the 13th International Conference on, Intelligent Systems Application to Power Systems*, 84–91. IEEE.

Nicola, Daniel. 2022. "Comparison of Four Mechanisms for Request Exchange in Collaborative Transportation." *International Transactions in Operational Research*.

Padmanabhan, Bhavya, Nathan Huynh, William Ferrell, and Vishal Badyal. 2020. "Potential Benefits of Carrier Collaboration in Vehicle Routing Problem with Pickup and Delivery." *Transportation Letters*, 1–16.

Padmanabhan, Bhavya, Nathan Huynh, William Ferrell, and Vishal Badyal. 2023. "Evaluation of Three Collaboration and Profit Sharing Methods for Carriers in Pickup-and-Delivery Problems." *Multimodal Transportation* 2 (2): 100066. https://doi.org/https://doi.org/10.1016/j.multra.2022.100066.

Rüther, Cornelius, and Julia Rieck. 2022. "Bundle Selection Approaches for Collaborative Practical-Oriented Pickup and Delivery Problems." *EURO Journal on Transportation and Logistics* 11: 100087.

Schopka, Kristian, and Herbert Kopfer. 2017. "Pre-Selection Strategies for the Collaborative Vehicle Routing Problem with Time Windows." In *Dynamics in Logistics*, 231–42. Springer.

Shaw, Paul. 1998. "Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems." In *International Conference on Principles and Practice of Constraint Programming*, 417–31. Springer.

Solomon, Marius M. 1987. "Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints." *Operations Research* 35 (2): 254–65.