10-2017

Combinatorial auction for transportation matching service: Formulation and adaptive large neighborhood search heuristic

Baoxiang LI
Singapore Management University, bxli@smu.edu.sg

Hoong Chuin LAU
Singapore Management University, hclau@smu.edu.sg

DOI: https://doi.org/10.1007/978-3-319-68496-3_9

Published in Computational Logistics: 8th International Conference, ICCL 2017, Southampton, October 18-20, 2017: Proceedings, Pages 127-142
https://doi.org/10.1007/978-3-319-68496-3_9

A Combinatorial Auction for Transportation Matching Service: Formulation and Adaptive Large Neighborhood Search Heuristic

Baoxiang Li and Hoong Chuin Lau

Fujitsu-SMU Urban Computing and Engineering Corporate Lab, Singapore Management University, 71 Stamford Road, 178895, Singapore
{bxli,hclau}@smu.edu.sg

Abstract. This paper considers the problem of matching multiple shippers and multi-transporters for pickups and drop-offs, where the goal is to select a subset of group jobs (shipper bids) that maximizes profit. This is the underlying winner determination problem in an online auction-based vehicle sharing platform that matches transportation demand and supply, particularly in a B2B last-mile setting. Each shipper bid contains multiple jobs, and each job has a weight, volume, pickup location, delivery location and time window. On the other hand, each transporter bid specifies the vehicle capacity, available time periods, and a cost structure. This double-sided auction will be cleared by the platform to find a profit-maximizing match and corresponding routes while respecting shipper and transporter constraints. Compared to the classical pickup-and-delivery problem, a key challenge is the dependency among jobs, more precisely, all jobs within a shipper bid must either be accepted or rejected together and jobs within a bid may be assigned to different transporters. We formulate the mathematical model and propose an Adaptive Large Neighborhood Search approach to solve the problem heuristically. We also derive management insights obtained from our computational experiments.

Keywords: Pickup-and-Delivery Problem with Jobs Dependency, Winner Determination Problem, Logistics

1 Introduction

In this paper, we study the winner determination problem (WDP) for an online auction platform for B2B less-than-truckload transport matching. In such platforms, we have multiple shippers with job bundles and multiple transporters with a heterogeneous fleet participating in an auction market, and the platform operator (auctioneer) is to perform a match of jobs with vehicles that maximizes profits at periodic (say hourly) intervals. Such platforms are rapidly emerging in a sharing economy with the rise of Uber-like business models.

The problem we present in this paper arises from a real-world implementation for a large urban logistics platform operator. It is a variant of the standard pickup-and-delivery problem with time windows (PDPTW), with additional dependencies among jobs and a number of side constraints between cargo, locations and vehicle types, as well as profitability as the objective function. Each shipper's delivery request may include a group of pickup-and-delivery jobs, and such grouping is called a shipper bid, which must either be accepted or rejected together. There are three reasons for the grouping of delivery jobs as a shipper bid: firstly, some jobs may be unprofitable and very difficult to find a matching transporter (due to low profit margins). But if such low-profit jobs would combine with high-profit jobs as a bundle, then the bid could be more ready to find matching transporters; secondly, some shipper companies may like to bundle delivery jobs themselves based on their own consideration (e.g., reverse logistics); thirdly, some shippers may prefer a one-stop solution rather than having to manage separate delivery requests.

In addition, arising from grouped jobs, a shipper bid may be split in terms of deliveries; that is, the different jobs in the shipper bid can be served by more than one transporter bid (assuming one transporter bid includes one vehicle in this paper). Our goal is to maximize the profit, which is calculated as total revenues associated with served bids minus the total transportation costs incurred correspondingly.

2 Literature Review

Transportation auctions are considered in the context where shippers compete with each other in order to purchase transport services at the lowest possible price from transporters aiming to sell their service at the highest possible price (see [5]). [1] first proposed a transportation auction to reduce logistics costs. Subsequently, a good number of transportation auction papers were published, which mainly considered full truckload (FTL, i.e. one bid uses all available space in a vehicle auction), e.g., [13]. However, not all pickup/delivery jobs could be formed as FTL bids, and under such case, FTL auction cannot fulfill both shipper and transporter requirements. During the last decade, practitioners started to test the more challenging settings of less-than-truckload (LTL) auction platforms. [7] for example proposed an LTL transportation auction, where auctioneers generate bundles of shipper requests and offers them to the transporters, and transporters place their bids for the offered bundles.

The research topics for transportation auctions mainly focus on two aspects: the bundle generation problem and WDP. For example, [9] formulated the bundle generation problem as a PDPTW in an iterative bid generation auction problem. On winner determination, the interesting aspect is in coping with uncertainty. [8] presented a double auction model for transportation service procurement in a spot market with stochastic demand and supply. [12] proposed a tractable two-stage robust optimization approach to solve the WDP under shipment volume uncertainty. In addition to the standard desirable properties for auctions, transport logistics auction designers must deal with the specific challenge on the economic sustainability of the auction platform. [15] for instance discussed a bi-criteria auction mechanism to achieve environmental sustainability while ensuring economic sustainability.

The above-mentioned papers focus on improving the service quality of the auction, whether from the strategic viewpoint (e.g., [5]) or operational viewpoint (e.g., [9]). From the computational perspective, [6] addressed the concerns of transporters bidding on an exponentially large set of bundles, and solving the corresponding exponentially large WDP. In this paper, we focus on a computationally efficient solution for an online auction platform for LTL matching. Unlike past research, we allow multiple vehicles to serve one bundle (consisting of multiple jobs) and each vehicle may serve jobs from different bundles (i.e. many-to-many matching). Moreover, to avoid too many rounds of bidding (e.g., [10]), we propose a simple single-shot auction where each shipper submits the bundles, each transporter submits the truck availability and cost structure; and the auctioneer will decide the winning shippers and transporters.

3 Mathematical model

The G-PDPTW integrates the pickup and delivery problem (PDP) and the group bundle constraints, aims to select a subset of bundled shipper jobs (bids) and design service itineraries, and maximize the profit obtained from shipper revenue minus transporter cost, at same time respecting shipper and transporter constraints. We formulate the problem as a mixed-integer programming (MIP) model in this section. First, we present notations used throughout the paper as shown in Table 1. Jobs within a bid are defined by a set of nodes.

G-PDPTW can be defined on a complete undirected graph $\mathcal{G} = (V, E)$ where $V = V^p \cup V^d \cup \{0\} \cup \{2n + 1\}$. Subsets V^p and V^d correspond to pickup and delivery nodes, respectively, while nodes 0 and $2n + 1$ represent the dummy depots (distance to other nodes, service time, and weight/volume are all equal to 0). While in the real-life auction platform there is no tracking for each vehicle's/carrier's origin, and each vehicle/carrier needs to start serving the shipper jobs within the jobs time window. For ease of reference, we arrange all nodes in V in such a way that all origins precede all destinations, and the destination of each job can be obtained as its origin offset by a fixed constant n.

Let K be the set of transporter vehicles. Each vehicle $k \in K$ has a weight capacity Q_k and volume capacity H_k. The hourly cost of vehicle k is p_k. Let O be the set of shipper bids. The revenue (i.e. bid price) for delivering a shipper bid o is represented by r_o, while z_o is a binary decision variable indicating whether shipper bid o is served or not. Each shipper bid includes one or more jobs, each job is defined by two nodes (a pickup node and a delivery node). A time window $[e_i, l_i]$ is associated with node $i \in V$, where e_i and l_i represent the earliest and latest arrival time, respectively. Each edge $(i, j) \in E$ has a travel time t_{ij}. In addition, let λ_i be the loading/unloading time, w_i be the weight, and c_i be the volume of node i (for a given pickup and delivery pair, $w_i = -w_{i+n}$, and $c_i = -c_{i+n}$).

For each arc $(i, j) \in A$ and each vehicle $k \in K$, $x_{ij}^k = 1$ if vehicle k travels from node i directly to node j. For each node $i \in V$ and each vehicle $k \in K$, let τ_i^k be the time for which vehicle k begins to serve node i, and W_i^k/C_i^k be the weight/volume load of vehicle k after visiting node i. The integer variable y_k indicates the hours traveled by vehicle k.

Table 1: Parameters and variables for the G-PDPTW model

n	Number of jobs, one job includes two nodes (one origin and one destination)
K	Set of vehicles, $K = \{1, 2, \ldots, \lvert K\rvert\}$, and $k \in K$
V^p	Set of origins $V^p = \{1, 2, \ldots, n\}$
V^d	Set of destinations $V^d = \{n+1, n+2, \ldots, 2n\}$
V	Set of nodes $V = V^p \cup V^d \cup \{0\} \cup \{2n+1\}$
	{0} and {2n+1} represent the vehicle dummy origin and destination points, and $i \in V$
O	Set of bids (each bid includes a group of jobs), $O = \{1, 2, \ldots, \lvert O\rvert\}$, and $o \in O$
$\lvert O_o\rvert$	Number of jobs inside bid o
r_o	Revenue obtained from serving bid o
w_i	Weight of node i
c_i	Volume of node i
$[e_i, l_i]$	Time window for node i
λ_i	Service time at node i
t_{ij}	Travel time between nodes i and j
Q_k	Weight capacity of vehicle k
H_k	Volume capacity of vehicle k
$[\iota_k, \hbar_k]$	Time window associated with dummy depot for vehicle k
p_k	Hourly cost of vehicle k
x_{ij}^k	Binary decision variables indicating if vehicle k goes directly from node i to node j,
	it is equal to 0 if vehicle k does not travel from node i to node j direct
y_k	Integer variables indicating the number of hours traveled by vehicle k
z_o	Binary decision variables indicating if bid o is served; it is 0 if bid o is not served
τ_i^k	Time point when vehicle k leaves node i
W_i^k	Weight load of vehicle k after visiting node i
C_i^k	Volume load of vehicle k after visiting node i

Given these notations, the formulation of the G-PDPTW is as follows:

$$\max \sum_{o \in O} r_o z_o - \sum_{k \in K} p_k y_k \tag{1}$$

Subject to:

$$\sum_{i \in O_o} \sum_{j \in V} \sum_{k \in K} x_{ij}^k = |O_o| z_o, \quad \forall\ o \in O \tag{2}$$

$$\sum_{j \in V} \sum_{k \in K} x_{ij}^k \leq 1, \quad \forall\ i \in V^P \tag{3}$$

$$\sum_{i \in V} x_{0,i}^k = \sum_{i \in V} x_{i,2n+1}^k = 1, \quad \forall\ k \in K \tag{4}$$

$$\sum_{i \in V} x_{i,0}^k = \sum_{i \in V} x_{2n+1,i}^k = 0, \quad \forall\ k \in K \tag{5}$$

$$\sum_{j \in V} x_{ij}^k = \sum_{j \in V} x_{ji}^k, \quad \forall\ i \in V^p \cup V^d,\ k \in K \tag{6}$$

$$\sum_{i \in V} x_{i,j+n}^k = \sum_{i \in V} x_{ij}^k, \quad \forall\ j \in V^P,\ k \in K \tag{7}$$

$$\tau_j^k + t_{j,j+n} + \lambda_j \leq \tau_{j+n}^k, \quad \forall\, j \in V^P,\ k \in K \tag{8}$$

$$(\tau_i^k + t_{ij} + \lambda_i) x_{ij}^k \leq \tau_j^k, \quad \forall\, i,\ j \in V,\ k \in K \tag{9}$$

$$(W_i^k + w_j) x_{ij}^k \leq W_j^k, \quad \forall\, i,\ j \in V,\ k \in K \tag{10}$$

$$(C_i^k + c_j) x_{ij}^k \leq C_j^k, \quad \forall\, i,\ j \in V,\ k \in K \tag{11}$$

$$e_i \ \leq \ \tau_i^k \leq l_i, \quad \forall\, i \in V^p \cup V^d,\ k \in K \tag{12}$$

$$0 \leq W_i^k \leq Q_k, \quad \forall\, i \in V^p \cup V^d, k \in K \tag{13}$$

$$0 \leq C_i^k \leq H_k, \quad \forall\, i \in V^p \cup V^d, k \in K \tag{14}$$

$$\tau_0^k \geq \iota_k, \quad \forall\, k \in K \tag{15}$$

$$\tau_{2n+1}^k \leq \hbar_k, \quad \forall\, k \in K \tag{16}$$

$$(\tau_{2n+1}^k - \tau_0^k)/60 \leq y_k, \quad \forall\, k \in K \tag{17}$$

$$x_{ij}^k, \tau_i^k, W_i^k \in \mathbb{R}_+, \quad \forall\, i,\ j \in V,\ k \in K \tag{18}$$

$$z_o \in \{0,1\}, \quad \forall\, o \in O \tag{19}$$

$$y_k, \in \mathbb{Z}_+ \quad \forall\, k \in K \tag{20}$$

The objective function (1) maximizes the total profit that corresponds to the revenue obtained from bids minus the transporter costs. The cost is calculated based on travel time (with hourly unit) and is calculated as the difference between the departure time and return time at the dummy depot. The objective function is set for the auction platform operator, and the profit will be rebated to the platform owner, shipper and transporter after delivery based on various performance indicators.

Constraints (2) show that the nodes belong to the same bid o is considered to be a bundle, i.e., they must be served or reject together. Constraints (3) indicate that every node can be served at most once by one vehicle. Constraints (4) and (5) are imposed to fix the origin and destination points (which are dummy nodes, with distance to all nodes equal to 0) of vehicles. Note that an empty route will be represented by a path with 2 stops, which starts at 0 and ends at (2n+1). Every node except the origin and the destination of a vehicle must have same number of preceding and one succeeding node, which is defined in Constraints (6). Constraints (7) and (8) ensure that the job origin is visited before the destination. Constraints (9), (10) and (11) compute the travel times and loads of vehicles (both weight and volume dimension). The shipper node time window constraints are defined in (12). Constraints (13) and (14) represent the vehicle capacity constraint in both weight and volume dimension. The time window associated with dummy depot for each vehicle is defined in Constraints (15) and (16). Moreover, Constraints (17) define the vehicle travel time in hours (translate from minutes based to hourly based). Finally, Constraints (19)-(20) specify the domains of the variables.

4 ALNS Approach

4.1 The ALNS Framework

Our heuristic is based on the ALNS described in [4,14] with simulated annealing as the local search framework, and the pseudo-code is presented in Algorithm 1. In the algorithm, each iteration includes two subroutines: job selection and perturbation. In particular, the request in the ALNS are treated independently and not as a bundle in most of the job selection and perturbation operators.

Algorithm 1: Adaptive Large Neighborhood Search

Input: Initial solution s, solution $s_{best} := s$, initial probabilities associated with the operators

while *stopping criteria not reached* **do**
    $s' := s$
    Apply operator P1 for pre-process of neighborhood search
    Apply selection operator (R1-R3) to select jobs for removal
    Apply perturbation operator (I1-I5) to remove selected jobs from s' and reinsert as many unserved jobs as we can into s'
    if $f(s') > f(s_{best})$ **then**
        $s := s'$, $s_{best} := s'$
    else
        if $f(s') > f(s)$ **then**
            $s := s'$
        else
            $s := s'$ with probability $p(s', s)$ defined in Equation (21)
end while
Remove the bids that are partly served in s_{best}

Output: s_{best};

Let s be the current solution, s' be the new solution, and $f(s)$, $f(s')$ – the corresponding objective values. If $f(s')$ is worse than $f(s)$, we accept the solution s with probability $p(s', s)$:

$$p(s', s) = \min\{1, e^{(f(s') - f(s))/\overline{T}}\}, \tag{21}$$

where $\overline{T} \geq 0$ is the "temperature" that starts at $\overline{T_0}$ and decreases every iteration using the expression $\overline{T} := 0.9999 \cdot \overline{T}$, $\overline{T_0}$ is defined in such a way that objective value of the first iteration is accepted with a probability 0.5. The simulated annealing structure is the same as in [4]. The search continues until the stopping criteria is met (20000 iterations or no improvement for the last 2000 iterations).

4.2 Solution Evaluation

Two solution evaluation approaches are used for the ALNS:
(1) $ALNS_F$: only feasible solutions are allowed during the search;
(2) $ALNS_I$: infeasible solutions are considered and a penalty of the violated constraints is added to the objective.

Let $c(s)$ be the routing profit, The solution is evaluated by $c(s)$ plus the penalty of timw window violation $\bar{t}(s)$ and load violation $\bar{q}(s)$:

$$f(s) = c(s) + \alpha_t \bar{t}(s) + \alpha_q \bar{q}(s) \tag{22}$$

For the $ALNS_F$, $f(s) = c(s)$ holds, because all constraints must be satisfied and $\bar{t}(s)$ and $\bar{q}(s)$ are equal to zero.

At the end of each iteration, the values of the parameters α_t, and α_q are modified by a factor $1+\delta$, with $0 < \delta \leq 1$. If the current solution is feasible with respect to load constraints, the value of α_q is divided by $1 + \delta$. Otherwise, it is multiplied by $1 + \delta$.

To compute the profit of each route, we need to compute the revenue minus the cost of the route. However, since a shipper bid may be assigned to multiple vehicles, it is impossible to precisely calculate the revenue of a single route during search. As a heuristic, we break the bundles and split the price of a bid to the jobs according to their weights/volumes. For the route cost, which is a function of the route duration in hours, we first set the vehicle k to depart from the depot at ι_k, and compute the total waiting time $\overline{W}$ along the route. If no violation of time window can be found, we postpone the departure time by adding $\overline{W}$. After that, we check the feasibility of the route, once an upper time window violation is found, the departure time is adjusted by deducting the upper time window violation value. The algorithm iterates until no time window violations can be found. Finally, we recalculate the route duration. For details we refer to Algorithm 2.

Algorithm 2: Travel time duration calculation

Input: Route $R := (0, 1, \ldots, 2n + 1)$, departure time $\tau_0 := \iota_k$, whole route waiting time $\overline{W}$, index $m \leftarrow 1$, and postponed time $u_0 \leftarrow \overline{W} + 1,\ u_1 \leftarrow \overline{W}$

while $u_m < u_{m-1}$ **do**
    $\tau_0 \leftarrow \iota_k + u_m$
    for each $i \in R$ **do**
    $\tau_i \leftarrow \max\ (\tau_{i-1} + \lambda_{i-1} + t_{(i-1,i)}, e_i)$
    Until $\tau_i > \min(l_i, \hbar_k)$, $u_{m+1} \leftarrow (u_m - \tau_i + \min(l_i, \hbar_k))$
    $m \leftarrow m + 1$
end while
Output: $\tau_{2n+1} - \tau_0$;

4.3 Initial Solution

An initial solution is constructed by a basic greedy insertion heuristic. The heuristic randomly chooses a job, and inserts it to the best position in the routes (with the highest profit added). Afterwards, the ALNS heuristic is implemented to improve the initial solution. Simulated annealing is applied during the ALNS update process.

4.4 Adaptive weight adjustment procedure

The choice of the selection and perturbation heuristics is governed by a roulette wheel mechanism. We have three selection operators and six perturbation operators. On the one hand, we diversify the search by combining different operators. On the other hand, a good balance between the quality of the solution and the running time can be reached by choosing a suitable operator at every iteration.

We define P_d^t as the probabilities of choosing operator d at iteration t. Starting from a predefined value, they are updated as $P_d^{t+1} := P_d^t(1-\rho)+\rho\chi_i/\zeta_i$, where ρ is the roulette wheel parameter, χ_i is the score of operator i, and ζ_i is the number of times it was used during the last 200 iterations. The score of an operator is updated as follows. If the current iteration finds a new best solution, the scores related to the used operators are increased by π_1; if it finds a solution better than the previous one, their scores are increased by π_2; if it finds a non-improving yet accepted solution, their scores are increased by π_3. Every 200 iterations, new weights are calculated using the scores obtained, and all scores are reset to zero.

4.5 Pre-process for neighborhood search (P1)

Once a new best solution been found and without partly served bid, we optimize the route (s_{best} and $f(s_{best})$) by sequentially removing job from the route and reinserted in the best position so as to maximize the profit.

For some instances, there always exist some bids may never be fully served during all the iterations, remain them in the selection sets seldom lead to better served solution. Therefore, from the 100 iterations of the ALNS, there is a 50% of chance for low win probability bids involved for the next iteration search, the low win probability criteria is set as below: shipper bid that has been served less than 45 times in the last 100 iterations, and transporter bid (vehicle) that serve less than 2 jobs on average in the last 100 iterations.

4.6 Jobs Selection

At each iteration, jobs are selected and added to a perturbation set C (set C initially includes the unserved jobs). Three selection operators are used, details shown as follows:

- **Random job based (R1)**: This operator randomly selects a number of jobs.

– **Random bid based (R2)**: This operator random selects a number of bids.
– **Partly served bid based (R3)**: Let U be the set of all partly served bid jobs (only part of jobs in a bid been served by vehicles), then, this operator randomly selects $50\% - 70\%$ jobs from set U.

4.7 Jobs Perturbation

After the procedure of jobs selection, five perturbation heuristics have been implemented.

– **1-by-1 (I1)**: The selected jobs are sequentially removed one by one and reinserted into the best position (the highest improvement for the current objective value).
– **Global all-at-once (I2)**: The operator repeatedly inserts jobs in the best position of all the routes. The difference with **I1** is that all jobs are removed at once, then inserted again one by one.
– **Balanced all-at-once (I3)**: All jobs are removed out from the route at once. Then, for every job, we choose a route with the lowest profit value to insert. It tends to generate a relatively balanced solution.
– **Tabu 1-by-1 (I4)**: This operator implements a diversification strategy similar to the tabu search. Suppose that job i is removed from some route k, the job is then prohibited to be reinserted into route k. The ban can only be canceled if insertion into route k leads to a better routing profit compared to the best-known routing profit of route k with i inside. For the job that has never been served before, skip the removal step and only do the insertion.
– **Local all-at-once (I5)**: Suppose, job i is removed from some route k, it tries to insert the job i into the same route k again but in a better position.

5 Computational Experiments

In this section, we first test our algorithm on benchmark instances, and then analyse the result for instances of moderate size. Our ALNS approach is implemented in Java, and executed on an Intel Xeon E5-2667v4 8C/16T (3.2GHz) 16 core CPU 32 GB RAM machine. The parameters used in the ALNS are shown in Table 2, chosen by the tuning strategy proposed by [4]. Each time only one single parameter is adjusted, while the rest are fixed. The setting with the best average behavior (in terms of average deviation from the best-known solutions) is chosen. This process iterates through all parameters once.

5.1 Performance Comparison on PDPTW Benchmark instances

To evaluate the effectiveness of the proposed ALNS approach, we first apply it to solve the PDPTW benchmark instances.[1] For detailed descriptions, we refer the reader to [3]. Due to the difference between G-PDPTW and PDPTW, essential

[1] See https://www.sintef.no/projectweb/top/pdptw/li-lim-benchmark.

changes must made to the ALNS and instances: 1) we assume every job stands for a bid in the benchmark instances; 2) we change the objective to minimize the travel distances and the number of used vehicles; 3) $ALNS_I$ is used, penalty is added to the objective value to ensure all the jobs must be served. The overall performance of the ALNS shows as Table 3, the results show 0.55-7.85% gap (best results of 16 runs) to the best benchmark results. Main reason is that the ALNS is tailored for G-PDPTW, and slight changes of the model may lead to quite different solution, e.g., if we only minimize the travel distance (without minimizing the number of vehicles as benchmark instance settings), we observe 149 improved solutions.

Table 2: Parameters used in the ALNS

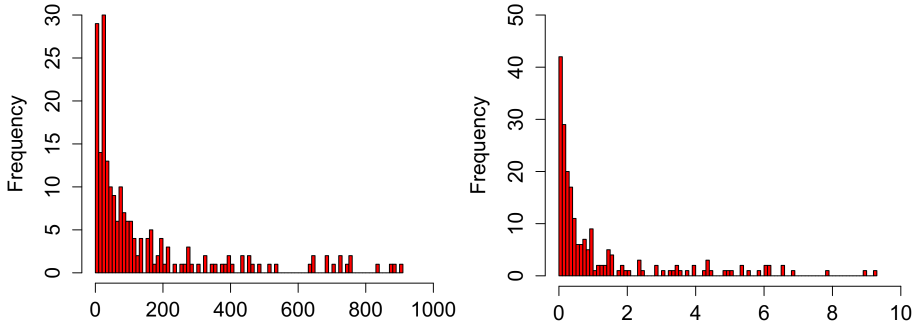
Description	values
Number of selection jobs	5%-25%
Roulette wheel parameter, ρ	0.50
Score of a global better solution, π_1	6.00
Score of a better solution, π_2	1.00
Score of a worse solution but accept, π_3	2.00
$P^0_{d_R}$ used for selection operators	0.33
$P^0_{d_I}$ used for perturbation operators	0.17

Table 3: Results comparison against the benchmark instances in [3]

# Nodes	Gaps	Running times (minutes)
100	0.55%	1
200	0.82%	3
400	3.09%	17
600	5.86%	42
800	6.87%	79
1000	7.85%	106

5.2 Relationship between bid features and win probability

Moving from computational performance, we next present our insights that give shippers and transporters some indication of the factors that may affect their probability of winning a bid. For this purpose, we run three groups of auctions. Each group includes 1000 instances. In the first group, one vehicle can serve 22 nodes on average. While in the second group, each vehicle may visit 11 nodes on average, and the ratio reduces to 7 for the third group. Bids with different prices, time windows, sizes, weights, and volumes are generated; for more details, we refer the reader to Table 4. All the test instances can be found at https://unicen.smu.edu.sg/pickup-and-delivery-problem-time-window-g-pdptw. Moreover, for the sake of notational consistency, we use “SBid”/“TBid” to represent shipper bid and transport “bid” (which is simply the capacity, availability and cost associated with a vehicle) respectively.



(a) Weight (b) Volume

Fig. 1: Histogram for weight (kg) and volume (m^3)

Table 4: Design of experimental instances

Number of instances	3000, 3 groups, each group includes 1000 instances, in the first group instances, the ratio between number of nodes and vehicles equals to 22, while the ratio equals to 11 and 7 for the second and third group instances, respectively
Shipper jobs time windows	Randomly choose from 1, 2, 3 and 9 hours
SBid size (number of jobs inside a SBid)	Randomly choose from (1-6)
Base price per SBid, ζ	Randomly chosen from \$15, \$30, \$45, \$60
SBid weight&volume	Weight value randomly choose from (0-1000) kg, then, find the corresponding volume, histogram graphs show as Figure 1a and 1b
Shipper job service time	15 minutes
Number of vehicles	Randomly choose a number from (2-21)
Vehicles capacity	2500 kg, 7 m^3
Vehicles available time period	9:00-18:00
Vehicles unit cost	Randomly choose from \$10, \$20, \$30, \$40 per hour

Additionally, we calculate the bid price using equation (23), which is mainly based on the number of jobs, and adjusted according to the weight/volume of the cargo. Let ζ and α be the per job price and number of jobs within a shipper bid, respectively. β, χ, and δ denote the number of small size (with cargo lighter than 10 kg), medium size (with cargo lighter than 100 kg but heavier than 10 kg), and large size (with cargo heavier than 100 kg) jobs inside a bid. In addition, the total weights of medium and large size cargo are represented as ϕ and φ.

$$price = \zeta\alpha - 1.71\beta - 4\chi + 4.7\delta + 0.09\phi + 0.007\varphi \tag{23}$$

Considering that running 3000 instances is time-consuming, we only apply the ALNS once with 5000 iterations for each instance, and $ALNS_F$ is applied. We analyze the effect of the shipper bid size (the number of jobs inside a bid), shipper bid unit price, shipper job time window, and transporter bid unit (hourly) cost on the win probability. By checking the win bids features, we calculate the relationship of the bid win probability and corresponding feature X by applying Bayes theorem:

$$P(win|X) = P(X|win)P(win)/P(X) \tag{24}$$

where $P(win|X)$ is the probability of a winning bid characterized by feature X, $P(win)$ is the prior probability of observing a win, and $P(X)$ represents the prior probability of observing X as a winning outcome. For a specific bid, we calculate its win probability based on a naive Bayes network, as shown in Figure 2.

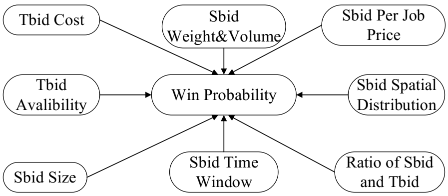


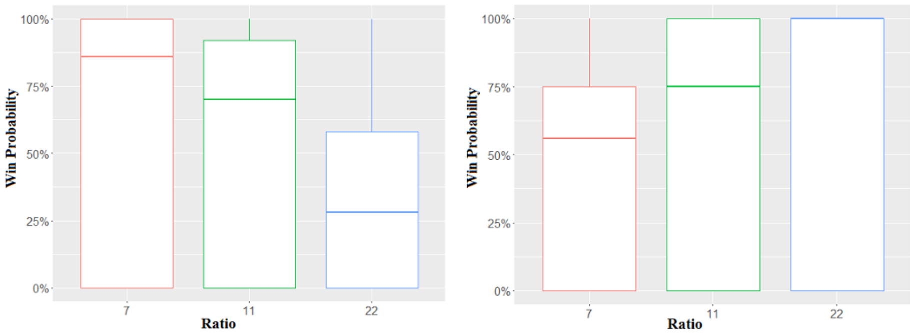
Fig. 2: Naive Bayes network

One can observe from Figure 3 that the SBid median win probability ranges from 25% to 87%, depends on the total number of SBid involved in an auction, the lower the ratio, the higher the win probability. On the contrary, the TBid median win probability varies from 56% to 100%, the higher the ratio, the higher the win probability.

In Figure 4a, we check the SBid win probability versus bid size, it addresses that the SBid include 1 or 2 jobs tend to win regardless other factors (e.g., price, time window, cost). However, if the ratio between SBid and TBid is high, the win probability is always low due to high demand and low supply. Assuming that the number of shipper bids is fixed, an unassigned SBid may win with an increasing the number of transporter bids.

Figure 4b shows the SBid win probability against the per job price of SBid (named “SBid Unit Price”), it indicates that the SBid win probability is not sensitive to per job price. Take “Nodes/Vehicles = 11” as an example, even if the price increases from 15 to 60, the SBid win probability only increases from 50% to 59%. The reason is that the SBid win probability is a determining by multiple factors, and the price is not a main factor. However, in real-life context, high bid price may attract more TBid to involve in the platform, subsequently, the SBid win probability will improve.

Figure 4c depicts the shipper job win probability against the time window. Suppose that the shipper submit a job with one hour time window for "Nodes/Vehicles = 7" group, the result is not so promising. However, if the time window width increases to 9 hours, the shipper bid win probability increases from 30% to around 40%. Moreover, the win probability of groups that "Nodes/Vehicles = 11" and "Nodes/Vehicles = 22" seems not affected by the time window. The reason is that different shipper jobs with different time windows are randomly bundled together, so the win probability not only depends on single job time window, but also relies on the groups time window. For instance, if the time window of all jobs in a given bid are 3 hours, then, the results are totally different from the situation that the half jobs time window equal to 1 hour and another half jobs time window equal to 5 hours.



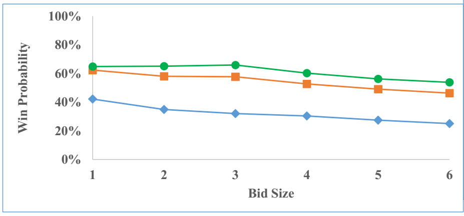
(a) SBid win probability

(b) TBid win probability

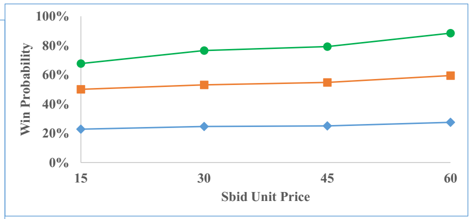
Fig. 3: Overall win probability

From Figure 4d, one can see the vehicles win probability seems not affected by unit cost under the case of "Nodes/Vehicles = 22". However, with the increasing of the vehicles number, the cheap vehicles (with low unit cost) have higher win probability compare to expensive vehicles (with high per unit cost). At the same time, even if the vehicles number is much lower than the shipper jobs number, the vehicle win probability does not reach to 100%, an explanation is that some vehicles cost is too high comparing to the shipper bid price, it is better to fail those bids as unprofitable to serve them.

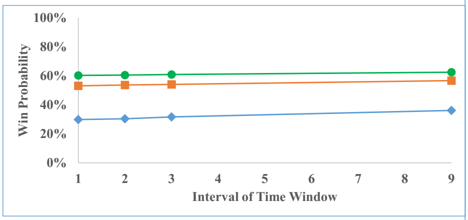
As the objective function is based on profit, we cannot use the win probability as criteria to evaluate the performance the algorithm. However, the win probability can be used to provide suggestions for both shippers and transporters, which is one means to improve the auction platform financial sustainability. For instance, a shipper bid with one job (time window equals 2 hours, unit price



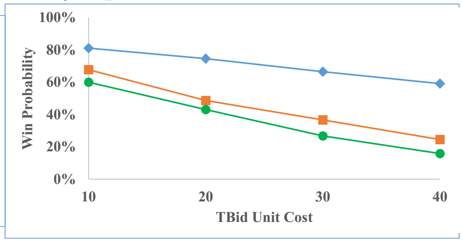
(a) SBid win probability versus SBid size



(b) SBid win probability versus average SBid job price



(c) Shipper job win probability versus shipper job time window



(d) TBid win probability versus TBid unit cost

Fig. 1. Win probability versus different factors, ● #Nodes/#Vehicles = 7, ■ #Nodes/#Vehicles = 11, ♦ #Nodes/#Vehicles = 22

equals to 15) win probability is approximately 50% (obtained from naive Bayes network). In contrast, by increasing the tight time window to 9 hours, and increase the price to $45 per job, the win probability can reach almost 100%.

In summary, the win probability depends on multiple factors drawn from both demands and supplies. Generally, most factors are independent of one another. Where some factor may depend on others, advanced machine learning techniques should be applied to predict the win probability.

6 Conclusion

In this paper, we investigate the winner determination problem with bundled jobs, which is an variant of the PDPTW. From the academic perspective, this work raises many new challenges. From a data analytics point of view, we find that the win probability may not be high when shippers/transporters randomly submit bids. Therefore, mechanisms should be properly designed to improve the win probability, such as moving from single-shot to multi-round iterative auctions, allowing the failed bid owner increase/decrease their bid price or relax some constraints. Besides that, we see the following broad areas for future research: 1) extending the current model by addressing heterogeneous vehicle routing with a mixture of cost structures (e.g., some traditional logistic companies prefer cost structure based on weight, volume, number of visited locations, or travel distance); 2) extending the current model to multi-objective, for example, the platform may need to achieve high match rate in addition to maximizing profit; 3) evaluating the impact of relaxing some constraints; for example, imposing a penalty cost for violating some rules instead of rejecting an order completely may benefit all stakeholders (shipper, transporter, and sharing platform owner); 4) profit sharing with the stakeholders in the form of rebates post-auction, which may incentivise more users to participate in the platform. In this regard, a fair and stable profit sharing mechanism was proposed in [11] that encourages coalition formation among multiple logistics providers for vehicle routing.

Acknowledgment

This research is funded by the National Research Foundation Singapore under its Corp Lab @ University scheme and Fujitsu Limited as part of the A*STAR-Fujitsu-SMU Urban Computing and Engineering Centre of Excellence.

References

1. Ledyard, J.O., Olson, M., Porter, D., Swanson, J.A. and Torma, D.P.: The first use of a combined-value auction for transportation services. Interfaces, **32**(5), 4–12 (2002)
2. Cordeau, J.F. and Laporte, G.: A tabu search heuristic for the static multi-vehicle dial-a-ride problem. Transportation Research Part B: Methodological, **37**(6), 579–594 (2003)
3. Li, H. and Lim, A.: A metaheuristic for the pickup and delivery problem with time windows. International Journal on Artificial Intelligence Tools, **12**(2), 173–186 (2003)
4. Ropke, S. and Pisinger, D.: An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transportation Science, **40**(4), 455–472 (2006)
5. Mes, M.: Sequential auctions for full truckload allocation. PhD thesis, University of Twente, The Netherlands (2008)
6. Chen, R.L.Y., AhmadBeygi, S., Cohn, A., Beil, D.R. and Sinha, A.: Solving truckload procurement auctions over an exponential number of bundles. Transportation Science, **43**(4), 493–510 (2009)
7. Berger, S. and Bierwirth, C.: Solutions to the request reassignment problem in collaborative carrier networks. Transportation Research Part E: Logistics and Transportation Review, **46**(5), 627–638 (2010)
8. Xu, S.X. and Huang, G.Q.: Transportation service procurement in periodic sealed double auctions with stochastic demand and supply. Transportation Research Part B: Methodological, **56**, 136–160 (2013)
9. Wang, X. and Kopfer, H.: Collaborative transportation planning of less-than-truckload freight. OR Spectrum, **36**(2), 357–380 (2014)
10. Dai, B., Chen, H. and Yang, G.: Price-setting based combinatorial auction approach for carrier collaboration with pickup and delivery requests. Operational Research, **14**(3), 361–386 (2014)
11. Agussurja, L., Lau, H.C. and Cheng, S.F.: Achieving stable and fair profit allocation with minimum subsidy in collaborative logistics. In Proceedings of the 13th AAAI Conference, Phoenix, Arizona, USA., 3785–3792 (2016)

12. Zhang, B., Yao, T., Friesz, T.L. and Sun, Y.: A tractable two-stage robust winner determination model for truckload service procurement via combinatorial auctions. Transportation Research Part B: Methodological, **78**, 16–31 (2015)
13. Kuyzu, G., Akyol, Ç.G., Ergun, Ö. and Savelsbergh, M.: Bid price optimization for truckload carriers in simultaneous transportation procurement auctions. Transportation Research Part B: Methodological, **73**, 34–58 (2015)
14. Li, B., Krushinsky, D., Van Woensel, T. and Reijers, H.A.: An adaptive large neighborhood search heuristic for the share-a-ride problem. Computers & Operations Research, **66**, 170–180 (2016)
15. Handoko, S.D., Lau, H.C. and Cheng, S.F.: Achieving Economic and Environmental Sustainabilities in Urban Consolidation Center With Bicriteria Auction. IEEE Transactions on Automation Science and Engineering, **13**(4), 1471–1479 (2016)