2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

Designing Large-Scale Intelligent Collaborative Platform for Freight Forwarders

Pang Jin Tan[1], Shih-Fen Cheng[1], and Richard Chen[2]

Abstract— **In this paper, we propose to design a large-scale intelligent collaborative platform for freight forwarders. This platform is based on a mathematical programming formulation and an efficient solution approach. Forwarders are middlemen who procure container capacities from carriers and sell them to shippers to serve their transport requests. However, due to demand uncertainty, they often either over-procure or under-procure capacities. We address this with our proposed platform where forwarders can collaborate and share capacities, allowing one's transport requests to be potentially shipped on another forwarder's container. The result is lower total costs for all participating forwarders. The collaboration can be formulated as an integer linear program we call the Freight Forwarders' Collaboration Problem (FFCP). It is a variant of the bin-packing problem, hence it is NP-Hard. In order to solve large-scale FFCP instances efficiently, we propose a two-step approach involving an initial greedy assignment followed by a fine-tuning step. Computational experiments have shown that our approach can offer a significant reduction of run-time between 77% and 97%, without any loss of solution quality.**

I. INTRODUCTION

In today's global supply chain, freight forwarders play a pivotal role as crucial facilitators connecting shippers with carriers. Forwarders first secure transport capacities from carriers at discounted rates, thanks to the substantial volumes they handle. They then resell these capacities to multiple shippers, often bundled with additional services such as customs brokerage. For shippers, this not only streamlines the shipping process, but it also proves to be cost-effective compared to purchasing capacities directly from carriers.

The forwarding industry, however, has its own set of challenges, particularly in capacity management. Due to demand uncertainty from the shippers, accurately forecasting capacity requirements becomes a daunting task for forwarders. Procuring too much capacity has a negative effect on profitability, while insufficient capacity leads to lost sales.

To address these systemic inefficiencies, we propose a Digital Marketplace for Forwarders. This platform aims to facilitate collaboration among participating forwarders, helping each of them to reduce their operating costs. Here is how it works. Each forwarder first puts forward its capacities (that it has procured) and its transport requests (that it has promised to ship for its clients, i.e. the shippers) to the platform. With visibility across all capacities and transport demands, the platform optimally reallocates requests among available capacities, independent of specific forwarders. This approach results in a significantly lower total shipping cost for all requests, as compared to the case where each forwarder solely utilizes its own capacities. The process of assigning requests to boxes can be formulated as an integer linear program. To solve practical real-world large-scale scenarios, we propose a two-step approach combining a greedy approach with an exact fine-tuning step.

The next section provides a review of related work. In the subsequent section, we formulate the Freight Forwarders' Collaboration Problem (FFCP), an integer linear program that assigns requests to services down to the container level for a group of forwarders. To expedite the solving of the integer linear program, we propose a two-step approach. First, we perform a greedy assignment. Then, in the second step, we fine-tune the solutions to achieve optimality. Finally, we present our experimental results and discuss managerial insights.

II. RELATED WORK

Our work is most closely related to collaborative transportation. Generally, literature in collaborative transportation can be classified into the collaboration of the following categories: carriers, forwarders, and shippers. Carrier collaboration has been studied quite extensively, especially for both Full-Truck-Load (FTL) and Less-than-Truck-Load (LTL) trucking. Li et al. [1] proposed a single-lane request approach for FTL trucking, where buyers and sellers submit multiple requests but a central coordinator picks one lane to be exchanged which increases social welfare the most. Lai et al. [2] extended the approach by allowing multiple requests hence making the connection to bundle generation and pricing. Freight consolidation is also often seen as a strategy in collaborative transportation. Zhang et al. [3] studied fair allocation for shippers who ship via a consolidation center. Similarly, Lai et al. [4] studied a shipper consortium problem where shippers hand over LTL shipments to a logistics service provider who then decides optimal routes to consolidate and route shipments, and then allocates cost back to shippers.

On the other hand, air carriers and ocean liners collaboration are typically studied under alliance formation hence they typically utilize a cost allocation approach ([5], [6]). However, freight forwarder collaboration is much less studied. The most relevant for us is capacity sharing by Lai et al. [7]. In their model, forwarders collaborate to first procure capacity during the pre-freight season at a discount. During freight season, they would then each bid for lanes depending

[1]Pang Jin Tan and Shih-Fen Cheng are with the School of Computing and Information Systems, Singapore Management University, {pangjin.tan.2021,sfcheng}@smu.edu.sg
[2]Richard Chen can be reached at rchen25@gmail.com

on their actual demand. The authors proposed an auction mechanism that guarantees truthful bidding.

Another stream of work related to ours is on bin-packing. Delorme et al. [8] reviewed exact algorithms using branch-and-bound, branch-and-price, and constraints programming. The classic paper on approximation schemes by Johnson [9] introduced First Fit, Next Fit, Best Fit, Worst Fit, Any Fit, and their extensions. Different variants of the bin-packing problem have also been studied extensively. Lodi et al. [10] and Berkey and Wang [11] studied the two-dimensional case. Martello et al. [12] and Lodi et al. [13] studied the three-dimensional case. Seiden [14] studied online bin-packing. Finally, the bin-packing problem with variable cost and size, where our problem is a special case, is studied by Kang et al. [15], Pisinger et al. [16], and Crainic et al. [17].

III. PROBLEM FORMULATION

In this section, we formally describe the Freight Forwarders' Collaboration Problem (FFCP). As mentioned earlier, forwarders act as middlemen between shippers and carriers. Typically twice a year, once for the winter season and once for the summer season, a forwarder procures capacities from its partner carriers for different port pairs (a port pair refers to the transport service connecting a pair of origin and destination ports). During this process, a forwarder would analyze historical trends and negotiate with carriers for more discounts for services that are expected to have higher demand. As the freight season approaches, the forwarder receives transport requests from shippers and meets these requests using the procured capacities.

In this paper, we focus on Less-than-Container-Load (LCL) shipments for ocean freight. In other words, shippers need only partial space in a container rather than a full container. To better utilize the procured capacities, a forwarder would consolidate multiple LCL shipments of different sizes into a single container. The more efficient the consolidation strategy, the higher the profit the forwarder earns. The consolidation problem can be viewed as a one-dimensional bin-packing problem.

We now look at a concrete example to illustrate the points above. Table I lists the demands Forwarder A receives. There are 4 transport requests with different sizes. The first three originate from USLAX and destine for CNSHA, while the last one is from DEHAM to SGSIN. From the supply side, Forwarder A has procured 2 containers for the USLAX-CNSHA service at $900 per container, and 1 container for the DEHAM-SGSIN service at $1200 per container, as shown in Table II. A container is assumed to be 20ft, with a capacity of around 30 cubic meters (cbm). As such, requests 1 and 2 can be consolidated into one container since they both require the same service; requests 3 and 4 have to be shipped in separate containers. As such, the total incurred cost is $3000.

In practice, it is challenging for forwarders to forecast accurately what the freight demand would be like. Often a forwarder might end up with over-supply for some services and under-supply for others. However, if multiple forwarders could collaborate by sharing their capacities, this could help alleviate the supply issues. There is yet another advantage of collaboration. For the same service, different forwarders negotiate different rates with their carriers. As such, there would be additional cost savings if a forwarder uses a container belonging to a different forwarder for the same service at a lower cost. We will now illustrate this by introducing the supply and demand of Forwarder B and explaining how Forwarders A and B can collaborate.

TABLE I
FORWARDER A'S DEMAND.

Request	Volume (cbm)	Service Required
1	14	USLAX-CNSHA
2	12	USLAX-CNSHA
3	10	USLAX-CNSHA
4	15	DEHAM-SGSIN

TABLE II
FORWARDER A'S SUPPLY.

Service	Cost per container	Supply
USLAX-CNSHA	$900	2
DEHAM-SGSIN	$1200	1

TABLE III
FORWARDER B'S DEMAND.

Request	Volume (cbm)	Service Required
1	6	USLAX-CNSHA
2	6	USLAX-CNSHA
3	6	USLAX-CNSHA
4	6	USLAX-CNSHA
5	15	DEHAM-SGSIN

TABLE IV
FORWARDER B'S SUPPLY.

Service	Cost per container	Supply
USLAX-CNSHA	$1000	2
DEHAM-SGSIN	$1100	1

Table III shows the demand from Forwarder B and Table IV shows the supply of Forwarder B. Note that without collaboration, Forwarder B would incur a total cost of $2100, and hence the combined incurred costs of shipping requests for both Forwarders A and B would be $5100. Now suppose they collaborate by allowing requests to be shipped on a container procured by other forwarders. Note that the first three requests of Forwarder A and the first four requests of Forwarder B are all shipped from USLAX to CNSHA. We can minimize cost by assigning requests 1 and 3 from Forwarder A and request 1 from Forwarder B to the first container that belongs to Forwarder A. We then assign request 2 from Forwarder A and requests 2, 3, and 4 from Forwarder B to the second container of Forwarder A. The total shipping cost for USLAX-CNSHA is $1800. Likewise,

both DEHAM-SGSIN requests can be consolidated into a container procured by Forwarder B for a cost of $1100. The combined incurred cost is $2900, which is significantly lower than the case without collaboration.

In the case without collaboration, each individual forwarder solves a bin-packing problem to minimize the number of containers required. With collaboration, the problem is not just minimizing the number of containers but rather minimizing total cost where each bin has a cost factor. This is formulated as an integer linear program called FFCP as below. Note that we we do not keep the association of requests to forwarders. Instead, we simply formulate our problem as one assigning a set of requests to a set of containers. Associations to the forwarders are made implicit.

Index Sets

- R: the set of requests, indexed by r.
- S: the set of services, indexed by s.
- R_s: the set of requests that can be assigned to service s.
- S_r: the set of services that is feasible for request r.

Parameters

- c_s: the cost per container on service s.
- n_s: the number of containers available on service s.
- v_r: the volume of request r.
- v^{max}: the max volume of a box.

Decision variables

- $x^i_{r,s}$: 1 if request r is shipped on service s box i, 0 otherwise.
- y^i_s: 1 if service s box i is used, 0 otherwise.

Model FFCP

$$\min \sum_{s \in S} \sum_{j=1}^{n_s} c_s y^i_s \tag{1}$$

s.t. (2)

$$\sum_{s \in S_r} \sum_{i=1}^{n_s} x^i_{r,s} = 1, \forall r \in R, \tag{3}$$

$$\sum_{r \in R_s} v_r x^i_{r,s} \leq v^{max} y^i_s, \forall s \in S, i = 1, ...n_s, \tag{4}$$

$$x^i_{r,s}, y^i_s \in \{0, 1\}. \tag{5}$$

The objective is to minimize the total cost of shipping all the requests. Constraint (3) ensures that each request is assigned to exactly one of the containers. Constraint (4) ensures that the total volume of requests fitted in a container does not exceed maximum capacity. Constraint (5) ensures assignment variables are binary.

IV. SOLUTION APPROACH

In this section, we describe our solution approach to solve FFCP. Figure 1 provides a visualization of the assignment problem. Each vertex on the left represents a request. We see that it is indexed by a forwarder and a request number, for example, $r_{A,1}$ represents Forwarder A request 1, and so on. The vertices on the right represent the different services available. The supply is not shown for brevity, but the costs are shown to illustrate that different forwarders have procured different rates for the same port pairs. The edges of the graph show possible assignments between requests and services.

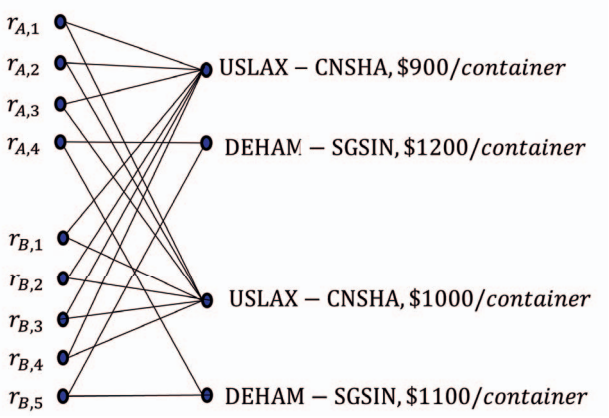

Fig. 1. Assigning requests to services for the reformulated approach.

Our next observation is that the FFCP can be decomposed into non-overlapping sub-problems where each sub-problem deals with a group of services that belong to the same port-pairs. This makes sense because if a request is supposed to be shipped on say service USLAX-CNSHA of a forwarder, then the request can be shipped on any other service offered by other forwarders as long as it is serving USLAX-CNSHA. Figure 2 illustrates our point. Instead of solving FFCP in its entirety, we break down the problem into groups of services, each group having services for the same port-pair, and solve each sub-problem separately. The total cost is simply the sum of the minimal cost for each sub-problem.

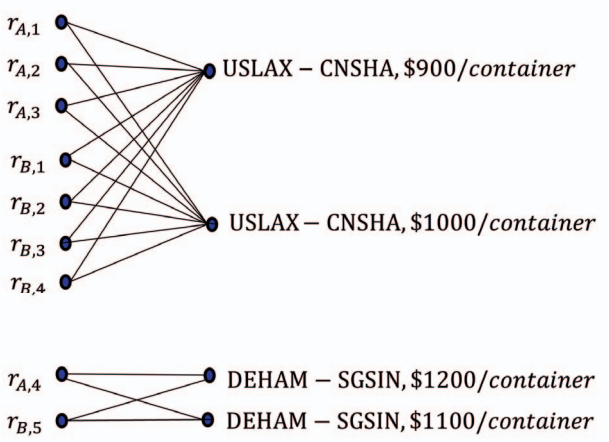

Fig. 2. Request-service assignments decomposed into non-overlapping sub-problems.

Finally, we turn our focus to solving the sub-problem, which is a special case of FFCP where each request can be assigned to any of the services. We take a two-step approach to solve this special case of FFCP. In the first step, we

obtain an initial solution by greedily assigning requests to a container with the first fit decreasing heuristic. This gives us a feasible solution and an upper bound on the number of containers needed for each service, n'_s. In the second step, we solve the special case of FFCP but the number of containers available for each service is now updated to n'_s, as obtained in the first step, instead of the original n_s.

Here is an illustration of the two-step approach to solving the sub-problem involving the USLAX-CNSHA port-pair assuming Forwarder A and Forwarder B collaborate. On the demand side, there are seven combined requests of size 14 cbm, 12 cbm, 10 cbm, 6 cbm, 6 cbm, 6 cbm, 6 cbm sorted in decreasing volume. On the supply side, there are four combined containers available with unit costs of \$900, \$900, \$1000, \$1000, sorted in increasing order.

In the first step, we repeatedly assign requests with the next largest volume to an available container with the lowest cost. Hence, the 14 cbm and 12 cbm requests are assigned to the first \$900 container, but the 10 cbm request is assigned to the second \$900 container. Next, the 6 cbm request cannot be fitted into the first container and hence needs to be assigned to the second container. Likewise, we can fit two more 6 cbm requests into the second container. Finally, for the last 6 cbm request, we need a third container which is the \$1000 container. This gives us a feasible assignment and hence we know that the optimal solution would not take more than two \$900 containers and one \$1000 container. Note that the actual supply is two \$900 containers and two \$1000 containers.

In the second step, we solve the following integer linear program exactly.

$$\min \sum_{s \in S} \sum_{i=1}^{n_s} c_s y_s^i \tag{6}$$

s.t. (7)

$$\sum_{s \in S} \sum_{i=1}^{n_s} x_{r,s}^i = 1, \forall r \in R, \tag{8}$$

$$\sum_{r \in R} v_r x_{r,s}^i \leq v^{max} y_s^i, \forall s \in S, i = 1, ..., n_s, \tag{9}$$

$$x_{r,s}^i, y_s^i \in \{0, 1\}, \tag{10}$$

where:

- $R = \{1, 2, 3, 4, 5, 6, 7\}$,
- $S = \{1, 2\}$,
- $n_1 = 2, n_2 = 1$,
- $c_1 = 900, c_2 = 1000$,
- $v_1 = 14, v_2 = 12, v_3 = 10, v_4 = v_5 = v_6 = v_7 = 6$,
- $v^{max} = 30$.

The optimal solution is obtained by assigning the 14 cbm, 10 cbm, and 6 cbm requests to one \$900 container and the remaining requests to another \$900 container. Note that we now use one less container as compared to the greedy approach in the first step.

V. EXPERIMENTAL SETUP

We discuss our computational experiments in this section and the next. We set up three different experiments to better understand our proposed two-step approach. In the first experiment, we vary the instance size from 10 services and 100 requests to 80 services and 800 requests. In the second experiment, we focus only on the case where each request can be assigned to any of the services. In the third experiment, we generate instances parameterized by forwarders explicitly, rather than instances based on requests and services. In all the experiments, we compare the solution quality and run-time of the proposed two-step approach with the exact approach. The experiments are conducted on a 144-core server with Intel Xeon Gold 6154 CPUs clocked at 3GHz, and a total RAM of 512GB running Rocky Linux 8.7. The models were implemented in Python and solved using ILOG CPLEX 22.1.

In the first experiment, a scenario is parameterized by the number of requests N_R and the number of services N_S. Each request is characterized by a port-pair and a volume. Each service is characterized by a port-pair, a cost-per-container, and a supply. Our first step is to generate a volume for each request, drawn from $U(1, 29)$. Here, $U(a, b)$ refers to a uniform distribution with lower bound a and upper bound b. Then, for each service, we generate a cost-per-container drawn from $U(800, 1200)$ and a supply drawn from $U(15, 50)$. Our next step is to associate a port pair for each request and each service. We first generate M, the number of port-pairs with $M \leq min(N_R, N_S)$. To ensure that every port pair gets assigned, we first assign the first M requests and first M services to each of the M port pairs. For the remaining requests, we randomly assign to each of them a random port pair. We do likewise for the remaining services.

In the second experiment, we focus only on single port pairs. In the case of single port pairs, each request can be assigned to any of the services. As explained earlier, an instance of FFCP can be decomposed into non-overlapping sub-problems involving different port pairs. The size of the biggest sub-problem will impact the overall run-time for an instance of FFCP and that is the reason for studying the sub-problem separately. To generate scenarios for single port pairs, we generate a random volume drawn from $U(1, 29)$ for each request. As for each service, we generate a cost drawn from $U(800, 1200)$ and a supply drawn from $U(15, 50)$. Instead of associating random port pairs to requests and services, we simply allow each request to be assignable to all of the services.

In the third experiment, we generate instances parameterized by forwarders' profiles. First, given the number of port pairs M, we generate m_s services for each port pair, where m_s is drawn from $U(1, 5)$. For each service, we generate a cost drawn from $U(800, 1200)$ and a supply drawn from $U(15, 50)$. Now we have all services where each service is associated with a port-pair, we can assign services to forwarders. For each forwarder, we iterate through each port-pair, and with a probability p we assign the port-pair to the forwarder. The specific service assigned is randomly chosen from the remaining unassigned services of the same port pair. The process is repeated until all services are assigned.

VI. RESULTS

The first experiment studies the run-time for different instance sizes ranging from 10 services and 100 requests to 80 services and 800 requests. Fig 3 shows the results. The run-time of the two-step approach is significantly lower than the run-time of the exact approach. In these instances, the reduction ranges from 81% to 97%. In general, run-time increases with instance size. However, we also see that for the larger instance size involving 60, 70, and 80 services, the run-time does not seem to increase. The reason is that while instance size plays an important part, it is the size of the biggest sub-problem that is the main driving factor for the overall run time. We also highlight that the solutions obtained via the two-step approach are the same as those generated using the exact approach.

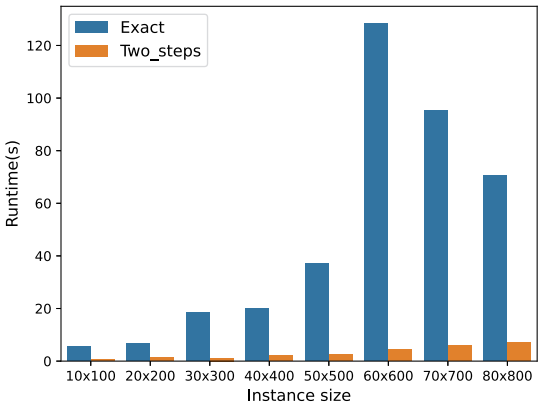

Fig. 3. Run-time for different instance sizes.

In the second experiment, we focus on instances where each request can be assigned to any of the services. In other words, we focus on solving the sub-problems. These sub-problem instances are also parameterized by the number of services and the number of requests. However, the number of services is typically not that large in practice. We investigate the run time of both the exact and two-step approaches for cases where the number of services ranges from 2 to 5 and the number of requests ranges from 10 to 90. The results are shown in Figure 4. Our two-step approach consistently outperforms the exact approach in terms of solution run time. For a given number of services, the run time is quadratic in the number of requests for the exact approach, whereas the run time is linear in the number of requests. Furthermore, the run-time savings is greater as the number of services increases. In these instances, the maximum run-time savings range from 77% to 96%. There is also no loss of solution quality for the instances in the second experiment.

In the third experiment, we generate scenarios based on forwarders' profiles. First, we observe that there is significant savings in terms of combined costs across all forwarders when they collaborate. The savings range from 15% to 25% as shown in Table V. Furthermore, the three scenarios depict different scales of collaboration. As the number of collaborating forwarders increases, there is a higher chance for a request to find a matching service from a different forwarder. This in turn makes the problem harder to solve. The results in table VI show that our proposed two-step approach performs significantly better, especially for scenarios with a larger number of forwarders. Moreover, the solutions generated by our two-step approach are the same as those generated by the exact approach.

TABLE V
RUN-TIME FOR DIFFERENT FORWARDERS' CONFIGURATIONS.

Scenario	Cost (without collaboration)	Cost (with collaboration)
10 forwarders, 47 requests, 20 services, 10 port-pairs	32660	27895
15 forwarders, 150 requests, 74 services, 20 port-pairs	117731	89112
20 forwarders, 338 requests, 106 services, 20 port-pairs	237281	176902

TABLE VI
RUN-TIME FOR DIFFERENT FORWARDERS' CONFIGURATIONS.

Scenario	Exact(s)	Two-steps(s)
10 forwarders, 47 requests, 20 services, 10 port-pairs	1.98	0.43
15 forwarders, 150 requests, 74 services, 20 port-pairs	21.50	0.92
20 forwarders, 338 requests, 106 services, 20 port-pairs	209.98	1.38

VII. CONCLUSION

In this paper, we introduce the Freight Forwarders' Collaboration Problem. The motivation comes from the observation that freight forwarders often find it challenging to procure the right capacities to serve their transport requests. Over-supply or under-supply of capacities can happen often. We propose that freight forwarders collaborate by sharing their capacities. In other words, a forwarder's transport requests can be assigned to a container procured by another forwarder. The combined shipping costs of all the forwarders to satisfy the demands can therefore be potentially lower. The assignment of transport requests to containers can be formulated as an integer linear program we call the Freight Forwarders' Collaboration Problem (FFCP).

Noting that FFCP is a variant of the bin-packing problem and hence NP-Hard, we need to seek an efficient solution for large-scale instances of FFCP. We first note that FFCP can be decomposed into non-overlapping sub-problems. Each sub-problem is then solved with a two-step approach. In the first step, we repeatedly assign the next biggest request to the cheapest container that has available space. This gives us a feasible assignment and also upper bounds on the number of containers required for each service. In the second step, we use these upper bounds to solve FFCP sub-problems exactly. The strategy of first using greedy to bound our problem

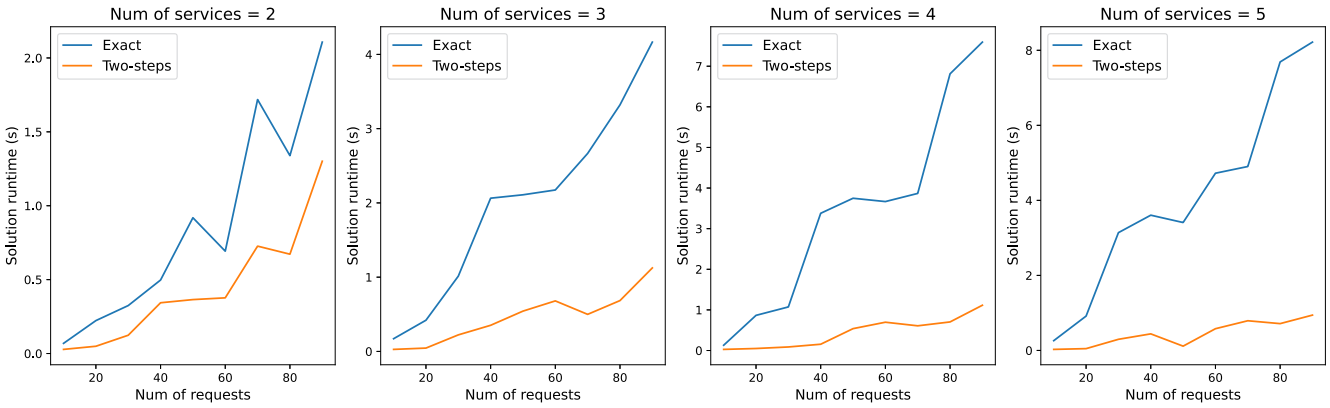

Fig. 4. Run-time comparison for number of services = 2,3,4,5.

which is then solved exactly in a second step proves to be efficient in many large-scale instances.

To study the performance of the proposed two-step approach, we conduct three experiments. In the first experiment, we vary the instance size of FFCP. The two-step approach has a lower run-time as compared to the exact approach with reduction ranging from 81% to 97%. In the second experiment, we vary the instance of the sub-problem of FFCP. This is the case where each request can be assigned to any of the services. The two-step approach also has a lower run-time as compared to the exact approach with reductions ranging from 77% to 96%. In the third experiment, we generate instances based on forwarders' profiles. The run-time savings are more significant when there is more collaboration between the forwarders. We also note that in all the experiments, the resulting objective values are the same as those generated based on the exact approach.

In conclusion, our proposed two-step approach provides a practical way to solve the FFCP for real-life instances. This is an important first step towards achieving forwarders' collaboration as we have demonstrated an efficient method to optimally assign requests to capacities for a group of collaborating forwarders. For future work, we shall study incentive mechanisms for a successful implementation of such a collaboration.

Acknowledgment

This research is supported in part by the Ministry of Education, Singapore, under its Social Science Research Thematic Grant (MOE Reference Number: MOE2020-SSRTG-018)).

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of the Ministry of Education, Singapore.

References

[1] J. Li, G. Rong, and Y. Feng, "Request selection and exchange approach for carrier collaboration based on auction of a single request," *Transportation Research Part E: Logistics and Transportation Review*, vol. 84, pp. 23–39, 2015.

[2] M. Lai, X. Cai, and Q. Hu, "An iterative auction for carrier collaboration in truckload pickup and delivery," *Transportation Research Part E: Logistics and Transportation Review*, vol. 107, pp. 60–80, 2017.

[3] W. Zhang, N. A. Uhan, M. Dessouky, and A. Toriello, "Moulin mechanism design for freight consolidation," *Transportation Research Part B: Methodological*, vol. 116, pp. 141–162, 2018.

[4] M. Lai, X. Cai, and N. G. Hall, "Cost allocation for less-than-truckload collaboration via shipper consortium," *Transportation Science*, vol. 56, no. 3, pp. 585–611, 2022.

[5] R. Agarwal and Ö. Ergun, "Network design and allocation mechanisms for carrier alliances in liner shipping," *Operations Research*, vol. 58, no. 6, pp. 1726–1742, 2010.

[6] L. Houghtalen, Ö. Ergun, and J. Sokol, "Designing mechanisms for the management of carrier alliances," *Transportation Science*, vol. 45, no. 4, pp. 465–482, 2011.

[7] M. Lai, W. Xue, and Q. Hu, "An ascending auction for freight forwarder collaboration in capacity sharing," *Transportation Science*, vol. 53, no. 4, pp. 1175–1195, 2019.

[8] M. Delorme, M. Iori, and S. Martello, "Bin packing and cutting stock problems: Mathematical models and exact algorithms," *European Journal of Operational Research*, vol. 255, no. 1, pp. 1–20, 2016.

[9] D. S. Johnson, "Fast algorithms for bin packing," *Journal of Computer and System Sciences*, vol. 8, no. 3, pp. 272–314, 1974.

[10] A. Lodi, S. Martello, and D. Vigo, "Recent advances on two-dimensional bin packing problems," *Discrete Applied Mathematics*, vol. 123, no. 1-3, pp. 379–396, 2002.

[11] J. O. Berkey and P. Y. Wang, "Two-dimensional finite bin-packing algorithms," *Journal of the Operational Research Society*, vol. 38, no. 5, pp. 423–429, 1987.

[12] S. Martello, D. Pisinger, and D. Vigo, "The three-dimensional bin packing problem," *Operations Research*, vol. 48, no. 2, pp. 256–267, 2000.

[13] A. Lodi, S. Martello, and D. Vigo, "Heuristic algorithms for the three-dimensional bin packing problem," *European Journal of Operational Research*, vol. 141, no. 2, pp. 410–420, 2002.

[14] S. S. Seiden, "On the online bin packing problem," *Journal of the ACM (JACM)*, vol. 49, no. 5, pp. 640–671, 2002.

[15] J. Kang and S. Park, "Algorithms for the variable sized bin packing problem," *European Journal of Operational Research*, vol. 147, no. 2, pp. 365–372, 2003.

[16] D. Pisinger and M. Sigurd, "The two-dimensional bin packing problem with variable bin sizes and costs," *Discrete Optimization*, vol. 2, no. 2, pp. 154–167, 2005.

[17] T. G. Crainic, G. Perboli, W. Rei, and R. Tadei, "Efficient lower bounds and heuristics for the variable cost and size bin packing problem," *Computers & Operations Research*, vol. 38, no. 11, pp. 1474–1482, 2011.