Journal of Supply Chain Management Science

https://journals.open.tudelft.nl/jscms

http://dx.doi.org/10.18757/jscms.2021.5781

Auction-based coopetition in the landside air cargo supply chain

Sanne van Alebeek [a*], Alessandro Bombelli [b]

[a] *Faculty of Civil Engineering and Geosciences, Delft University of Technology, 2628 CN Delft, The Netherlands*

[b] *Faculty of Aerospace Engineering, 2629 HS Delft, Delft University of Technology, The Netherlands*

Article history: received 16-04-2021, accepted 18-06-2021, published 30-06-2021

Abstract – Increased competitiveness with other transport systems and declining operation margins have motivated freight forwarders in the air cargo transport industry to look into horizontal collaboration. This paper focuses on developing a fully integrated five-phase auction-based coopetition model, a form of horizontal collaboration where competition is preserved. A combinatorial auction is used to exchange transportation requests without having to reveal critical company information. Freight forwarders submit requests into an auction pool, where they are grouped into attractive bundles by a central planner and offered for auction. The request selection and bundling procedures are based on the time windows of the request deliveries. A freight forwarder's bid on each bundle is equal to the marginal profit of that bundle, which is obtained by solving two NP-hard routing problems with a simulated annealing and large neighborhood search meta-heuristic. A unique aspect of the auction is that the dock capacity of the ground handlers is taken into account, which helps to alleviate truck congestion at the ground handlers. The potential of the auction-based coopetition model is shown for an air cargo supply chain scenario. There is a clear increase in profitability for the collaborating freight forwarders because the auction model decreases the transportation costs for the entire coalition. This cost reduction is achieved by an increase in transport efficiency, while the collaboration disadvantages, as seen in the literature, are limited.

Keywords: Combinatorial auction; less-than-truckload planning; coopetition; pick-up and delivery problem with time windows

1. Introduction

The increase in demanding consumer lifestyles and the current societal search for sustainability call for an increase in efficiency of logistic service providers. Until now, freight forwarding companies in the air cargo supply chain were able to manage the complexity and competitiveness of the market with high margins, optimization of their own resources, and vertical collaboration. The emergence of integrative services increased competitiveness with other transport systems, and declining operations margins have motivated the air cargo transport industry to look into horizontal collaboration (Ankersmit et al. 2014, Ferrell et al. 2020). Horizontal collaboration occurs when companies that work at the same level of the supply chain decide to work together rather than operate separately, with the goal to increase their efficiency.

Freight forwarders (FFs) organize the transportation requests from shippers by finding a routing strategy for their trucks from the freight forwarding depot to the ground handlers (GHs) on the airport. Bombelli and Tavasszy (2021) developed a landside air cargo supply chain problem with pick-up and delivery time windows to model horizontal collaboration between FFs for the transport of air cargo. The collaboration is modeled by assuming that a homogeneous fleet of trucks is shared by all FFs. A central planner, with full information of the collaborating forwarders and all transport requests, computes the optimal routing strategy for the fleet. Wu (2019) developed a meta-heuristic for this formulation to decrease the computational time, which poses challenges for medium- and large-sized instances. A new addition to the model is that it accounts for the maximum dock capacity that a GH is

[*] Corresponding author. Email address: Sannevanalebeek@gmail.com

willing to share with a consortium of FFs in a vertical collaboration fashion. With an increase in online demand and worldwide rapid transport of goods, it occurs more frequently that the number of arriving trucks at GHs for export operations exceeds the dock capacity, leading to queuing and delays, which are highly undesirable.

In centralized collaborative settings, such as in (Wu 2019), it is assumed that a central planner has full information on the scenario to be optimized. Therefore, in these situations, the most optimal collaboration solution can be computed: this implies that all stakeholders are willing to share all the information that is needed. In non-collaborative (individual) settings, there is no exchange of information, and every party optimizes its own request fulfillment. There is limited research on quantifying the added value of exchanging more information.

Recent studies show good results in improving reliability, use of resources, sustainability, congestion, costs, travel distance, and other system performances (Wu 2019, Gansterer et al. 2020b, Verdonck et al. 2013, Berger and Bierwirth 2010). However, the unwillingness of companies to share information is delaying the shift towards more horizontal collaboration (Gansterer et al. 2020a, Raweewan and Ferrell 2018). In this paper, the potential of an auction-based horizontal collaboration system is analyzed, where the system preserves competition between FFs and requires limited information to be revealed. Such a type of collaboration is referred to as auction-based *coopetition* (a portmanteau of collaboration and competition).

Berger and Bierwirth (2010) defined an auction-based collaboration system consisting of five auction phases. All cooperating FFs submit transportation requests into a common auction pool. The requests are then bundled and offered for auction. The FFs bid on the offered bundles of transportation requests, and the bundles are assigned to FFs based on their bids. In the final step, the collected profits are distributed among the FFs.

Gansterer et al. (2020b) propose an auction-based collaboration system that uses only aggregated information of all requests of a carrier to determine which requests should be entered into the auction pool. To create request bundles, they only use a valuation of the individual requests in the auction pool. In this way, it is ensured that no critical or sensitive information is inadvertently exchanged. In the paper, it is shown an increase in the profit and a reduction in the number of generated bundles needed for a profit increase. A new profit-sharing method is introduced that guarantees individual rationality and does not require sensitive or critical information from the collaborators. More specifically, every stakeholder receives a profit that is no lower than the profit that would be received via individual planning: this is a crucial factor to enhance horizontal collaboration.

The contribution of this paper is twofold: (1) a fully integrated five-phase auction mechanism is modeled, including dock capacity constraints and a tailored request selection and bundling approach (methodological contribution). (2) three different types of collaboration, auction-based coopetition, no collaboration, and full collaboration, are compared on transport efficiency while keeping track of the possible disadvantages (assessment of system potential).

The paper is structured as follows. In Section 2, we introduce the methodology, with a special focus on the auction-based coopetition and on the meta-heuristic that is used to solve one decision block of such coopetition framework. Section 3 describes the details of the mathematical formulation of the auction framework. Section 4 shows the results of the auction when compared to other collaboration schemes for two synthetic instances based on real-world data. Finally, Section 5 provides conclusions and directions for future works.

2. Methodology

The main contribution of this paper is to show how horizontal coopetition, in the form of an auction-based system, can affect cargo transport from FFs to GHs in key areas and improve current performances. Transport efficiency is seen as one of the major opportunities for horizontal collaboration (Cruijssen et al. 2007). The goal is to evaluate the possible increase in transport efficiency while keeping track of the underlying disadvantages that the introduction of an auction-based coopetition system would entail.

To analyze the key effects of introducing the auction-based coopetition, a comparison between three types of collaboration is made: Individual collaboration (I), Auction-based coopetition (A), and Full collaboration (F). The comparison of the three collaboration types is based on KPIs and other performance measures, as discussed in Section 2.3 and Section 2.4.

The auction coopetition is the main focus of this paper. The five auction phases from Berger and Bierwirth (2010) form the basis for the solution method of the auction coopetition. In this paper, all five auction phases are modeled and solved, as shown in Section 3. By comparing the auction coopetition to the individual planning and the full collaboration, the potential of the auction-based coopetition is determined. An overview of all necessary steps, including their input information, is shown in Figure 1 (Rounded blue boxes indicate the three collaboration models, rectangular boxes indicate inputs and outputs to the model, and rounded boxes indicate processes).

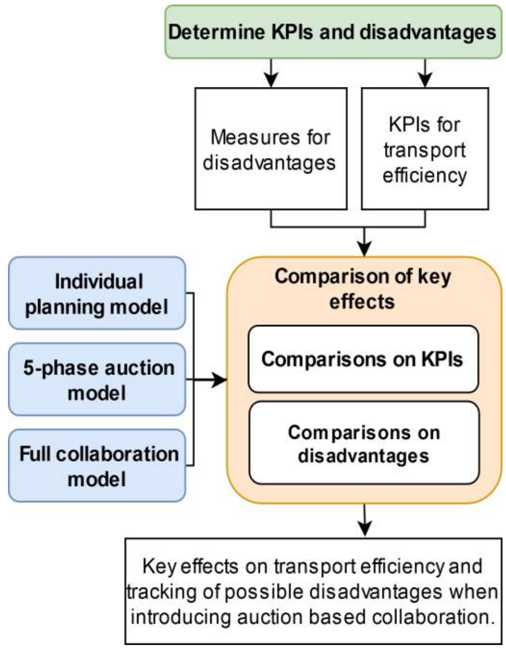

Figure 1. Methodology framework.

2.1. Problem Formulation

In both the auction coopetition and the full collaboration, a consortium of FFs agrees to collaborate to perform export deliveries to GHs. Each FF has a set of requests that have to be transported from the location of the FF to certain GHs within a specified planning horizon. At the same time, the GHs involved in the initiative agree to reserve a subset of their export docks for trucks belonging to the consortium. In our paper, the consortium of FFs is relatively small. While being this beneficial from a computational perspective, we also believe this to be practically more easily implementable (e.g., many small consortia rather than a large one). Hence, we assume that each GH reserves a single dock for the small consortium so that the others can serve all the remaining FFs. The goal of the coalition is to find a more attractive planning solution for all involved parties compared to a non-collaborative scenario. A graphical representation of all three collaboration types can be found in Figures 2, 3, and 4.

In the auction coopetition and the full collaboration, a central planner supervises the coalition and defines: (1) a new request allocation, i.e., which FF will handle which request, (2) a routing strategy for each FF, i.e., a sequence of FFs and GHs to visit, (3) a loading strategy, i.e., a sequence of requests to be picked up and then delivered, (4) a dock assignment strategy, i.e., a truck-dock pair for every warehouse visited. Given the size and complexity of the problem, the goal is not necessarily to solve each problem to optimality. Rather, the goal is to demonstrate that by adhering to the rules of the consortium, every FF can increase its profit with respect to the situation where it continued to operate as a single entity.

The main difference between the auction coopetition and the full collaboration is the method for assigning requests to the FFs. In the full collaboration, the central planner has full information on the shipments of every FF in the coalition. With the meta-heuristic based on (Wu 2019), a request assignment is found such that routing costs are minimized. In the auction coopetition, no critical company information is shared with the other FFs or the central planner. The central planner also acts as an auctioneer for the combinatorial auction. Additionally, the FFs are seen as individual companies that are all responsible for the transportation of their assigned requests. The assignment of the requests to the collaborating FFs is done by means of five integrated auction phases, initially proposed by Berger and Bierwirth (2010):

1. Request selection by the FFs: every FF selects requests deemed as non-profitable or not very appealing, based on predetermined characteristics to enter the auction pool.

2. Request bundling by the central planner: the requests in the auction pool are put together in sets to form more attractive sets called bundles.
3. Bidding by the FFs: each FF bids on the bundles generated in the previous step. A bid is based on the marginal profit for a FF that is associated with handling that specific bundle.
4. Winner determination by the central planner: the bundles get assigned to the FFs in such a way that the overall profit for all FFs combined is maximized.
5. Profit sharing for the FFs: the profit achieved by the request exchanges is divided among all participating FFs in a way that accounts for their contribution while preserving individualism.

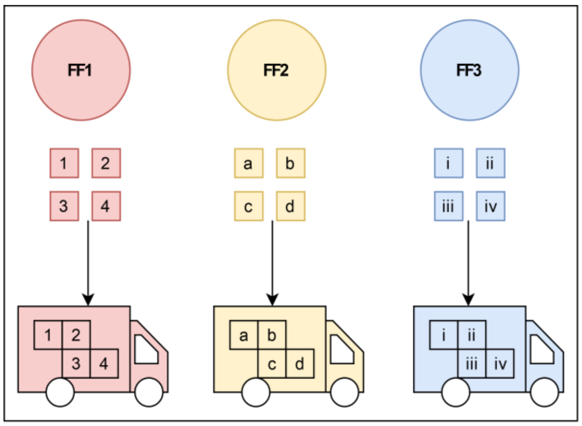

Figure 2. Diagram of individual planning.

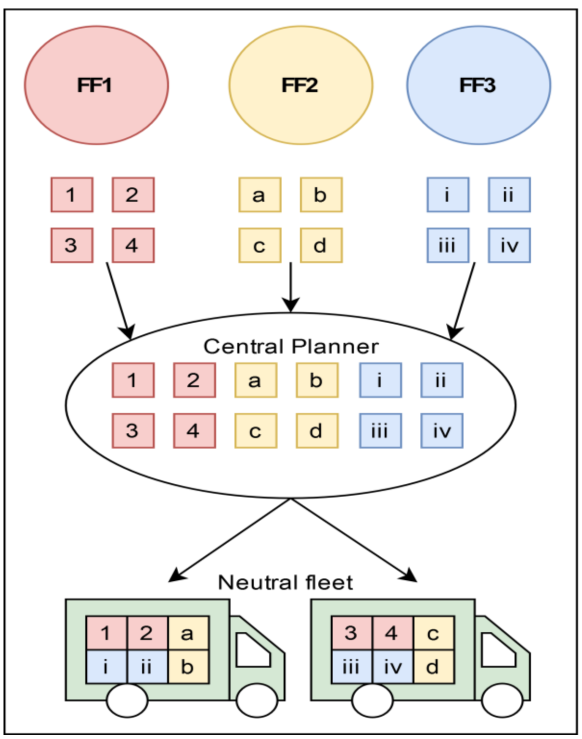

Figure 3. Diagram of full collaboration.

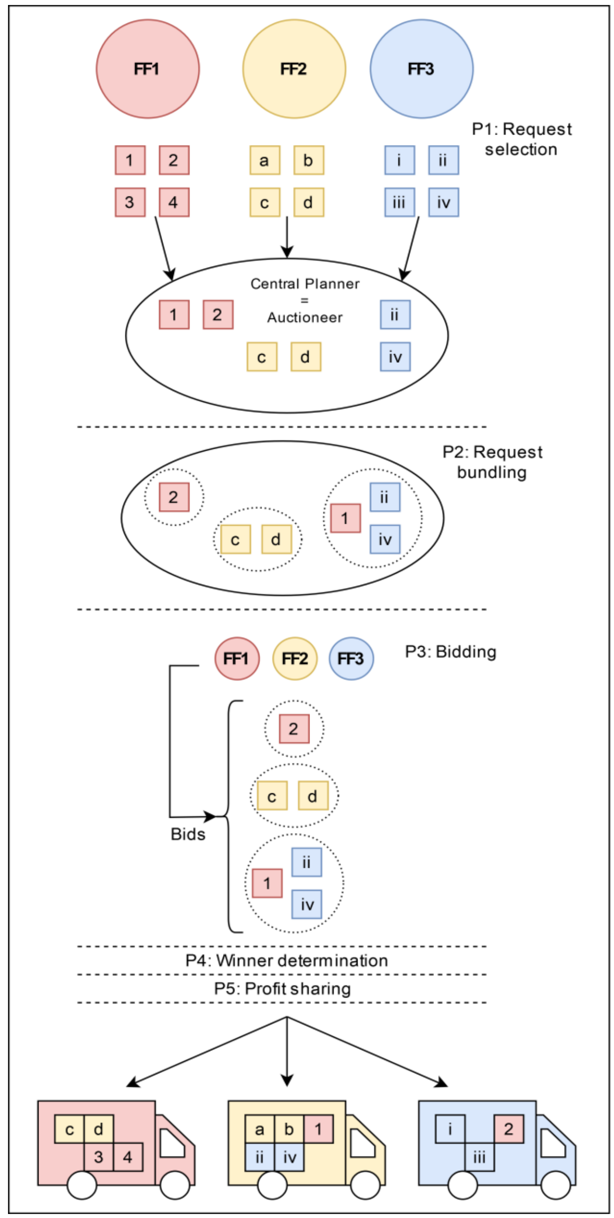

Figure 4. Diagram of auction-based coopetition.

In the individual planning, each FF defines their routing strategy, loading strategy, and dock assignment strategy, not taking into account the requests of the other FFs. In each of the three collaboration types, each request is mapped by two nodes, a pick-up node on the FF side and a delivery node on the GH side. All pick-up and delivery nodes are characterized by a time window $[e_i, l_i]$, where e_i is the early service time of a request, while l_i is the late service time of a request: trucks can only visit a node within its time window. An early arrival ($< e_i$)

implies that the truck will have to wait for the request to be available. A late arrival ($> l_i$) implies infeasibility since the time window was missed. It is assumed that all requests are consolidated at the FFs or upstream in the supply chain. Therefore, trucks are only moving Unit Load Devices (ULDs). The dock capacity is only considered on the delivery side of the problem, i.e., there is one dock available at each GH. This assumption is justified because GHs are generally the bottleneck in the landside air cargo supply chain. In addition, two restrictions that are specific to the air cargo supply chain are added. (1) To ensure an efficient loading strategy for the trucks, all pick-ups precede all deliveries in a truck tour. (2) To model rear-loading of the trucks and to avoid unnecessary unloading at intermediate GHs, deliveries are carried out in reverse order with respect to pick-ups in a Last-In-First-Out (LIFO) approach. This requirement is especially crucial because, in this paper, the ULDs occupy most of the lateral space in the trailers.

2.2. Meta-Heuristic

The meta-heuristic is based on a simulated annealing (SA) framework. In each iteration of the SA framework, a new neighbor solution is found with a Large Neighborhood Search (LNS). The LNS generates a new solution S^* from the current solution S_c by removing requests from a route and then inserting requests into a route (Ropke and Pisinger 2006, Wu 2019). The cost (J) of a solution (S) is determined by the total amount of time it takes to deliver all requests to the $GH_s(TTT_S)$ and a predetermined transportation cost per time parameter (c_τ): $J(S) = c_\tau TTT_S$. The objective of the meta-heuristic is to find a routing solution that minimizes the total transportation time. Additional costs are added to the objective function if the solution is infeasible, either if the solution generates dock capacity violations or if some shipments are not delivered.

Part of the input of the meta-heuristic is the set of requests that need to be shipped. By adjusting the set of shipments, the meta-heuristic can be used to calculate the full collaboration costs and the individual planning costs. To calculate the full collaboration costs, the set of requests is equal to all requests of all FFs. To calculate the individual planning costs for one FF, the set of requests is the subset of requests originating from that specific FF.

In this paper, the total amount of allowed run time is set for every SA performed. In general, the probability of accepting a worse neighbor solution as the new current solution depends on a temperature T that decreases with every iteration. A common approach to update T from iteration i to iteration $i+1$ is to decrease it with a constant such that $T_{i+1} = cT_i$, where generally c is a number very close to 1, e.g., 0.999. In addition, the initial temperature T_0 is generally chosen in such a way that a solution worse than the initial solution by a predefined percentile amount is chosen with a 50% probability (Ropke and Pisinger 2006). This combination of c and T_0 enables the LNS to accept more easily worse solutions in early iterations to better explore the solution space. In later iterations, due to the decrease in temperature, it is less likely for the LNS to move from a better to a neighboring worse solution instead. In this paper, we experimentally tested different combinations of T_0 and c, and eventually chose a rather unconventional yet effective strategy for the problem at hand. The temperature T at each iteration is equal to the amount of time (in seconds) that is left of the total allowed run time (3600 s) for the SA. We chose this value because, when compared with an average cost of an initial solution, it makes it reasonably likely for a worse solution to be accepted as the new current solution in early iterations. On the other hand, the transition from a temperature to the next one is more abrupt, hence accepting new solutions only if they are better quite quickly. Since the time we allow for our meta-heuristic is quite limited, this approach accepts worse solutions only in the first iterations, while it keeps exploring the most promising solution otherwise. It might be argued that our choice of T makes use of the SA framework only for the initial iterations. In the end, the probability of accepting a neighbor solution S^*, if $J(S^*) > J(S_c)$, is determined by the probability function $P(S^*) = \exp\left[\frac{J(S_c) - J(S^*)}{T}\right]$.

In each iteration of the SA framework, a new neighbor solution is found with an LNS that generates a new solution S^* from the current solution S_c by randomly choosing a removal operator that removes requests from a route and then randomly choosing an insertion operator that inserts requests into a route. The removal operators are: 1. Shaw removal, (Shaw 1997), (Ropke and Pisinger 2006). 2. Random removal (Ropke and Pisinger 2006). 3. Worst removal (Ropke and Pisinger 2006). 4. Shortest route removal (Li et al. 2016). 5. FF-GH removal (Bombelli and Tavasszy 2021).

The insertion operators are: 1. Basic greedy insertion (Ropke and Pisinger 2006). 2. Tabu greedy insertion (Ropke and Pisinger 2006). 3. 2-Regret insertion (Ropke and Pisinger 2006). 4. Route addition (Bombelli and Tavasszy 2021).

After a removal and insertion pair has produced a new neighbor solution S^*, a routine is carried out to identify and reduce dock capacity violations without increasing other violations. In fact, while in the computation of J the overall dock capacity violation of the new solution is considered, no preemptive action is explicitly taken to limit such violation. The routine consists of two stages: Time slack strategy and departure time adjustment strategy. If, for example, two trucks arrive at the same dock at the same time, the time slack strategy tries to resolve the dock capacity violation by making one of the trucks waiting on the other truck. The truck with the most time slack in its route is the truck that has to wait. The time slack of a route is explained in more detail in Bombelli and Tavasszy (2021). If it is possible to resolve the dock capacity violation with the time slack strategy without causing other violations, the departure time adjustment strategy is applied. If this is not possible, the solution is discarded. With the departure time adjustment strategy, it is possible to prevent waiting times (caused by early arrival) by adjusting the departure times of the trucks. If, for example, a delayed truck has to wait for 5 minutes for another truck to unload, it would have been better if that delayed truck left the origin depot (O_d) 5 minutes later. In that case, the truck would not have to wait the extra 5 minutes. Whether delaying the departure of a truck is possible is also based on the time slack of a route.

In individual planning, a FF is not aware of the planning of other FFs. Therefore, the time slack strategy and departure time adjustment strategy cannot be used for individual planning. However, in individual planning, trucks do have to queue, which is modeled using only the time slack strategy.

2.3. Quantifying Transport Efficiency

Comparing the KPIs of one collaboration type with another shows the difference in performance on many different aspects of efficiency. It is also interesting to compare not only the overall system performance but also the performance per FF. In this paper, the following KPIs are used: (1) Profit: The profit made by a FF is equal to the revenue minus the costs for transporting the requests to the GHs. The revenue is assumed fixed per FF, and the costs only depend on the total transportation time. Consequently, the difference in the total profit between the types of collaboration gives insight into the cost efficiency of the overall transportation planning. (2) Distance traveled: The distance traveled of a collaboration type is defined as the sum of the distances driven by the trucks. Obviously, it takes time to travel a certain distance, so the distance is implicitly incorporated in the costs. To give a complete overview, the distance is presented as well, where the distance is an indicator of the environmental impact. (3) Load factor: There are two load factors, namely weight and space. More specifically, trucks can be either weight- or volume-bounded, and whichever condition is met first, according to the specific shipments loaded, is the limiting one. A higher load factor means that the trucks are fuller, which indicates a more efficient use of resources like trucks and drivers. Additionally, it is an indicator of the environmental impact. (4) Waiting time for a dock to become available: The total waiting time of all trucks in a collaboration type gives insight into the amount of congestion at the GHs. (5) Amount of trucks: A reduction in the number of required trucks could eventually reduce the investment costs of the coalition or of an individual FF. (6) Amount of truck arrivals at GHs: A truck arrival is defined as a truck that docks at a GH. The total amount of truck arrivals at the GHs provides insight into the truck congestion at the GHs.

The main reason for choosing these six KPIs is to gain insight into the transport efficiency of the system, especially congestion. This is achieved primarily with the KPIs load factor, waiting time for a dock to become available, and the amount of truck arrivals at the GHs. Waiting time for a dock to become available and the number of truck arrivals at the GHs, in particular, provide two different perspectives. While it might be argued that limiting the number of truck arrivals at a warehouse should automatically alleviate congestion, careful dock planning might still be able to handle cases with many arrivals as long as they do comply with the planned schedule. From a business perspective, it is especially interesting to look at the profit, distance traveled, waiting time for a dock to become available, and amount of trucks. These last three KPIs directly influence the total costs that a FF incurs operating. If more profit can be achieved by the FFs, then they may consider working together without revealing confidential pieces of information and eventually transport their requests more efficiently.

2.4. Collaboration Disadvantages

Although collaboration has been identified as a key factor in reducing transportation costs and increasing efficiency in most supply chains, its implementation is still not fully developed. This is mainly due to several factors that are perceived by stakeholders as disadvantages. In this paper, the following aspects that could become disadvantages of collaboration are tracked: (1) Profit allocation: When FFs work together and obtain an increased joint profit, this profit needs to be divided over all participating FFs. A profit reallocation is considered fair if (a) All FFs make at least the same amount of profit after the profit reallocation compared to the individual situation (individual rationality) (b) FFs get higher profit shares if they contribute more to the total collaboration gain (proportional to contribution). (2) Autonomy: When different FFs collaborate, they agree on a set of rules for their collaboration, which restricts them in making individual decisions. How many decisions, how the decisions are made, and in whose best interest they are made is tracked for all five auction phases. (3) Ease of use: To track the ease of use, the number of extra actions needed from the FFs to ensure a well-functioning auction-based coopetition system is identified. (4) Information sharing: For each step in the auction, the amount, the type, and with who the information is shared is tracked. (5) Market position: The market position of a FF is determined by its share of the transportation market relative to the shares of its competitors in the same market. A possible downside of collaboration is that competitors (FFs) could obtain information about each other's pricing. With this information, they could try to undercut their competitors, causing the others to lose market share.

3. Auction Model

The overall auction model consists of five phases. Each of the five phases requires input from the previous phase or additional data. In Figure 5, a graphical overview of the five phases can be found (Horizontal arrows represent information exchanges between the FFs and the central planner and vice versa.).

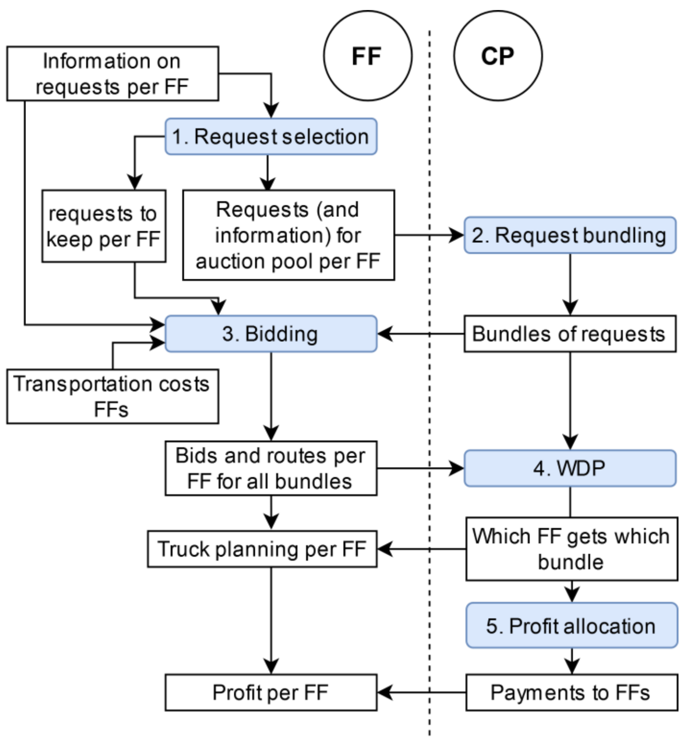

Figure 5. Overview of the auction model. Left: actions by the FFs. Right: actions by the central planner.

3.1. Request Selection

In this phase, all FFs individually decide which requests they want to keep and which requests they want to submit to the auction pool. This decision is based on predetermined characteristics of the requests. No exchange of information is needed between FFs or the central planner. The input for the request selection phase is information on all requests per FF. For each request r_i, consisting of pick-up node i and delivery node $i + \sigma$ (where σ is the total number of requests), the location and the time window of the delivery are known to the FF. Studies have shown that selecting requests based on a characteristic that can make a request unattractive for one

FF yet attractive for another is more effective, for the entire coalition, than only selecting on profit or revenue (Gansterer and Hartl 2016, Schopka and Kopfer 2017). In the landside air cargo supply chain, such a characteristic is the location of the delivery combined with the time window in which the request needs to be delivered at that GH. For example: a FF may have to deliver three requests at GH1 within similar time windows and two requests at GH2 at completely different time windows. The three requests with a similar delivery time window are more attractive for this FF to keep. Furthermore, there is a chance that the other two requests are attractive for other FFs of the consortium.

The similarity of two time windows is determined by the amount of overlap they have. The amount of overlap between node i and node j (o_{ij}) is calculated with $o_{ij} = \max(0, \min(l_i, l_j) - \max(e_i, e_j))$. The amount of overlap of a group of nodes $(o_{i,j,\cdots,m,n})$ is calculated with $o_{i,j,\cdots,m,n} = \max(0, \min(l_i, l_j, \cdots, l_m, l_n) - \max(e_i, e_j, \cdots, e_m, e_n))$.

For each FF, the requests are sorted on delivery location. All delivery nodes with pick-up at FF f and delivery at GH g are denoted by B_{fg}^G. The total overlap of an individual delivery node (o_i) has with all other requests in B_{fg}^G, is calculated with $o_i = \sum_{j \in B_{fg}^G} o_{ij} \ \forall i \neq j$. The total overlap of a set of delivery nodes (o_S) is calculated with $o_S = \sum_{i \in S} \sum_{j \in S} o_{ij} \ \forall i < j$.

Example 3.1. *FF1 has to deliver the requests summarized in Table 1, with nodes* $13, 14, 15 \in B_{11}^G$ *and* $16, 17, 18 \in B_{12}^G$. *To clarify the notation: request 1 consists of pick-up node 1 and delivery node 13, which also shows that* $\sigma = 12$.

Table 1. Delivery data for FF1.

Request	Node	FF	GH	e_i	l_i	o_{ij}	o_i	B_{1j}^G
1	13	1	1	300	480	$o_{13,14} = 40$	220	
2	14	1	1	160	340	$o_{14,15} = 40$	80	260
3	15	1	1	300	480	$o_{15,13} = 180$	220	
4	16	1	2	300	480	$o_{16,17} = 0$	0	
5	17	1	2	100	280	$o_{17,18} = 180$	180	180
6	18	1	2	100	280	$o_{18,16} = 0$	180	

Based on the overlap calculations, the following selection criteria were considered: (1) For each FF, keep the requests that individually have the highest amount of total overlap (o_i). The requests with delivery nodes 13, 15, 17, and 18 in example 3.1. (2) For each FF, keep the requests that have the highest amount of overlap as a group $(o_{ij,\cdots,m,n})$. The requests with delivery nodes 13, 14, and 15 in example 3.1. (3) For each FF, keep the requests that have the highest amount of overlap as a set (o_S). The requests with delivery nodes 13, 14, and 15 in example 3.1.

In this paper, a combination of methods (3) and (1) is chosen that ensures the requests are chosen as a set and outliers of the set are excluded. First, method 3 is applied to select the initial set of requests to keep. Then the total overlap of each individual request (o_i) in this set is evaluated against a preset minimum. If the request does not meet the preset minimum, it will be put into the auction pool. Selection of the best set continues until the threshold, a percentage that depicts the minimum number of requests the FF wants to keep, is met. All requests that are not selected are put into the auction pool.

3.2. Request Bundling

Once each FF has decided which requests to keep, each FF communicates to the central planner which requests are put into the auction pool. The set of requests in the auction pool is denoted by A. FFs do not share information on their routing or their capacity constraints with the central planner in this phase. The central planner only needs to know the location and the time window of delivery of all requests in the auction pool. In phase four of the auction system, each FF is assigned one or no bundle. Therefore, the number of bundles that are reassigned is always less or equal to the number of FFs (N_f). Because each request can only be assigned to one FF, the objective of the bundling phase could also be seen as finding promising partitions of the auction pool. The partitions consist of at most N_f bundles. The collection of bundles is denoted by D, which is initially an empty set. Suggestions for bundle criteria for selecting promising bundles: (1) Requests in the same bundle have the same delivery location. (2) Requests in the same bundle have a high amount of overlap at a GH location. (3) A bundle in itself is only promising if the rest of *A* is partitioned into at most $N_f - 1$ promising bundles.

The following steps are executed to add promising bundles (not yet included in D) to D: (i) All requests in *A* are sorted by their delivery location: $A_j^G =$ requests in the auction pool with delivery at GH j. The first sets of requests that are added to D are $A_j^G\ \forall j \in G$. These bundles form a partition of the auction nodes that comply with criteria (1) and (3). (Assuming there are more (or equal) FFs than GHs). (ii) All requests in *A* are sorted by their pick-up location: $A_i^F =$ requests in the auction pool with pick-up at FF *i* = the requests that FF *i* put into the auction pool. To ensure the feasibility of phase 4, the sets of requests $A_i^F \forall i \in F$are added to D. (iii) All requests in *A* are sorted by their pick-up and delivery location: $A_{ij} =$ requests in the auction pool with pick-up at FF *i* and delivery at GH *j*. The A_{ij} comply with criteria (1) and are added to D. (iv) The requests per GH in the auction pool (A_j^G) are partitioned into sets of requests that have a high amount of overlap with each other. This partitioning is done with hierarchical clustering, where the resulting clusters can be seen as bundles. The amount of overlap between any two requests in A_j^G denotes the closeness of those requests. The distance between any two requests is the maximum overlap in A_j^Gminus the overlap between the two requests. The linkage criterion used is complete-linkage clustering. More about hierarchical clustering can be found in (Johnson 1967). These types of bundles comply with criteria (1) and (2).

For the bundles produced with steps (*i*) to (*iv*), the main problem with complying with criterion (3) is that *A* is partitioned into too many bundles. Therefore, some bundles are combined to form new bundles, those are A_j^G. A combination of two bundles is simply a bundle that contains all requests from both bundles. This combination method limits the number of extra bundles added to D, namely $\binom{N_g}{2}$ bundles, where N_g is the number of GHs.

3.3. Bidding

It is imaginable that a FF does not want to handle certain types of requests. It is possible to incorporate this into the model by allowing the FF not to bid on that bundle. However, to ensure maximal flexibility of the solution, it is assumed that every FF bids on every offered bundle. Only if a FF is not capable of handling a certain bundle, they do not bid on that bundle. The bids in the combinatorial auction are based on the marginal profits of a FF for handling the requests in the bundles. The marginal profit of handling bundle b for FF *f* is defined as the profit with bundle b minus the profit without bundle b, the latter being the same as the profit of only handling the kept requests, see request selection phase in Section 3.1. The profit of handling a set of requests is determined by the revenue a FF obtains from the shipper minus the costs of handling those requests. As mentioned before, FFs keep their initial revenue. However, the cost of serving the requests is not known yet. To determine the costs of handling a set of requests, a routing problem must be solved. In this work, the routing problems are solved with the meta-heuristic discussed in Section 2.2. For each bundle, two routing problems must be solved:

1. FF *f* calculates the cost of handling bundle b $\left(cost_{with}^{fb}\right)$ by solving the routing problem with the kept requests and the bundle requests.
2. FF f calculates the costs of only handling the requests that they kept $\left(cost_{kept}^{f}\right)$ by solving the routing problem with only the kept requests.

The extra cost of handling bundle b for FF *f* is called the marginal cost: the cost of handling the kept requests with the requests in the bundle minus the cost of only handling the kept requests $\left(cost_{with}^{fb} - cost_{kept}^{f}\right)$. The FFs communicate the marginal costs of all bundles to the central planner. Notice that the FF does not communicate the cost or profit of their initial route planning nor their initial revenue.

The central planner can then calculate the marginal profit of handling bundle b for FF f (mp_{fb}). Again, it is important to note that revenue is not reallocated between the FFs, nor is extra revenue created. Thus, the marginal revenue is zero. That means that the only way the auction coopetition can create extra profit for all FFs is to reduce the total transportation costs. The bid on bundle b from FF f is set as the marginal profit. The central planner constructs a bid matrix (MP) based on the marginal profits of the FFs, where $N_B =$ number of bundles. The amount of routing problems that need to be solved is $N_F \times (N_B + 1)$. This shows why it is important to keep the number of bundles limited.

$$MP = \begin{bmatrix} mp_{1,1} & \cdots & mp_{1,N_B} \\ \vdots & \ddots & \vdots \\ mp_{N_F,1} & \cdots & mp_{N_F,N_B} \end{bmatrix}$$

3.4. Winner Determination Problem

The central planner calculates the optimal reallocation of the bundles based on the bids of the FFs. The required input for the winner determination problem (WDP) is shown in Table 2. The central planner is also informed about the number of docks the GHs have reserved for the participants of the collaboration: $N_D^g =$ number of docks available at GH g, which is set to 1 in this paper. The outcome of the WDP is a bundle to FF assignment that maximizes the total system's marginal profit. For example, if bundle b is assigned to FF *f*, this is denoted as $x_{fb} = 1$. To also take the dock capacity into account, the central planner calculates all dock capacity violations between every bundle to FF assignment. In Table 2, all required information for this calculation is shown.

Table 2. Information provided from the FF to the central planner for the WDP.

Parameter	Description
mp_{fb}	Bid from FF i on bundle b.
rk_f	Routing of only kept requests by FF *f*.
rw_{fb}	Routing of kept requests by FF *f* with bundle *b*.
tk_f	Corresponding timestamps[a] of only kept requests by FF *f*.
tw_{fb}	Corresponding timestamps[a] of kept requests by FF *f* with bundle *b*.

[a] Timestamps are the departure and arrival times for each node in the route planning

Each bid is based on a truck routing with corresponding timestamps. Therefore, the central planner now knows at what time every truck arrives at a GH. If in assignment x_{fb} and x_{uv} two trucks arrive at the same GH at the same time, this is a dock capacity violation. The amount of dock capacity violations between assignment x_{fb} and x_{uv} is denoted with: $DC_{fb,uv}$. The mathematical model of the WDP is formulated based on the paper by Gansterer and Hartl (2016), see Table 3 and the mixed-integer linear program (equations 1 to 8). The objective function (1) maximizes the total marginal profit of the entire auction system. Each FF can win at most one bundle (2), and each bundle can only be assigned at most once (3). Constraint (4) ensures that each request is assigned exactly once. An FF can only win a bundle if he submitted a bid for the bundle (5). Constraint (6) regulates that if two bundle to FF assignments are chosen that have a dock capacity violation, the decision variable k is set to the number of dock capacity violations. If only one of the two assignments is chosen, decision variable k remains zero. (7) defines that all x_{fb} are binary, and (8) defines that $k_{fb,uv}$ is always a natural number. This mathematical formulation can be seen as an extension of the well-known set partitioning problem. To guarantee feasibility, the central planner created feasible bundles as mentioned in Section 3.2. The output of the WDP is a bundle to FF assignment, which means that each FF receives an overview of which requests they have to transport. The routing plan for these requests was calculated by themselves with the predetermined meta-heuristic from Section 2.2.

There are two possible outcomes of the WDP:

1. A bundle to FF assignment that has no dock capacity violations. In this situation, the route planning of each FF is feasible in combination with the route planning of all other FFs. Therefore, the planning can be executed accordingly.
2. A bundle to FF assignment with dock capacity violations. Here, the violations need to be solved. If there are dock capacity violations in the outcome of the WDP, decision variable k shows which bundle to FF assignments cause the dock capacity violations. If for example x_{11} and x_{22} need the same dock at the same time, the $k_{11,22}$ will denote the number of dock capacity violations between this pair. An algorithm finds the exact time and location of the dock capacity violation and tries to shift the timestamps of the arriving trucks in such a way that

the routes become feasible. More about this procedure, called the time slack strategy and the departure time adjustment strategy, can be found in 2.2. If it is not possible with this procedure to solve the dock capacity violations in the routes, the WDP is called again to find an alternative bundle to the FF assignment. This is done by adding an extra constraint to the WDP based on $k_{fb,uv}$. The extra constraint ensures that in the new solution x_{11} and x_{22} cannot both be chosen: If $k_{fb,uv} \geq 1$ the following constraint is added to the WDP: $X_{fb} + x_{uv} \leq 1$. This procedure is an iterative approach, as the new bundle to FF assignment may again contain an unsolvable dock capacity violation. If it is not possible to find a feasible bundle to FF assignment within an acceptable number of iterations (15), the iterative algorithm is stopped, and all FFs handle their original requests themselves (going back to complete individual planning).

Table 3. Explanation of the notation of the WDP.

Notation	**Description**
$\pi(MP, D)$	Total marginal profit after solving the WDP.
D	Set of offered bundles, $b \in D$.
MP	Matrix containing the bids.
mp_{fb}	Bid from FF *f* on bundle *b*.
C	Cost associated with a dock capacity violation in the WDP.
F	Set of FF, $f \in F$.
R	Set of requests, $r \in R$.
W_{br}	0/1 Parameter indicating whether request *r* is included in bundle *b* or not.
Q_{fb}	0/1 Parameter indicating whether FF *f* submitted a bid for bundle *b* or not.
$DC_{fb,uv}$	Parameter indicating the number of dock capacity violations between x_{fb} and x_{uv}.
x_{fb}	Decision variable indicating whether bundle *b* is assigned to FF *f*.
$k_{fb,uv}$	Decision variable indicating the number of dock capacity violations if there is a dock capacity violation in the bundle to FF assignment. Initially, these are all set to 0.

$$\pi(MP, D) = max \sum_f \sum_b mp_{fbx_{fb}} - C \sum_f \sum_b \sum_u \sum_v k_{fb,uv} \tag{1}$$

$$\sum_b x_{fb} \leq 1 \quad \forall f \in F \tag{2}$$

$$\sum_f x_{fb} \leq 1 \quad \forall b \in D \tag{3}$$

$$\sum_f \sum_b x_{fbW_{br}} = 1 \quad \forall r \in R \tag{4}$$

$$X_{fb} \leq Q_{fb} \quad \forall f \in F, \forall b \in D \tag{5}$$

$$\left(x_{fb} + x_{uv}\right) * DC_{fb,uv} - k_{fb,uv} \leq DC_{fb,uv} \qquad \forall f, u \in F, \forall b, v \in D \tag{6}$$

$$X_{fb} \in \{0,1\} \qquad \forall f \in F, \forall b \in D \tag{7}$$

$$K_{fb,uv} \in \{0,1,2,\cdots\} \forall f, u \in F, \forall b, v \in D \tag{8}$$

In this study, the dock capacity violations are solved in this phase, the WDP. There are other moments where the dock capacity can be solved. For instance, even before the request selection, the delivery time windows of the requests could be adjusted in such a way that all requests get a mutually exclusive time window. So, each request can be delivered in their own time window and no other request. Unfortunately, this restricts the entire solution space of the problem unreasonably. Another option would be to take the dock capacity into account in the request selection and bundling phase. The meta-heuristic used for the bidding in this paper does take into account the dock capacity, as is explained in Section 2.2. If all requests that go to the same GH are transported by the same FF, the meta-heuristic ensures that there is no dock capacity violation at that GH. However, this would simply mean a redistribution of the requests where each FF is assigned all requests that go to one GH. This is no longer auction coopetition as defined in this paper.

3.5. Profit Sharing

In this phase, the extra profit obtained by the coopetition is distributed among the FFs. This phase is executed by the central planner, and it does not require additional information. In this paper, a new profit-sharing mechanism, developed by Gansterer et al. (2020b), is chosen because of the following characteristics: (1) Profit reallocation is fair, as described in 2.4. (2) Profit reallocation can be executed without critical information. (3) Profit reallocation ensures group rationality. (4) Profit reallocation is computationally manageable. All necessary information for the profit reallocation is derived from: (A) The bids on the bundles that the FFs are assigned in the WDP. Φ_f is defined as the marginal profit for FF f of handling the assigned bundle, i.e., the marginal profit (negative marginal cost) of handling the acquired requests. (B) The bids of the FFs on bundles consisting of their own offered requests. ξ_f is defined as the marginal profit for FF f of the bundle that consists of the requests that were put into the auction pool by FF *f*, i.e., the marginal profit (negative marginal cost) of handling the FFs initially offered requests.

Subsequently, the central planner can now establish how much profit each FF will gain if adhering to the request assignment found in the WDP. For each FF, the amount of extra profit is equal to $\vartheta_f = \phi_f - \xi_f$. This is equal to the gained profit by buying requests minus the missed profit for selling requests for FF *f*. The total extra profit gained by the auction-based coopetition is $\Theta = \sum_{f \in F} \vartheta_f$. This extra profit is distributed among the FFs in the coopetition with the profit-sharing equation: $\lambda_f = \frac{\Theta}{2}\left(\frac{|\phi_f|}{\Phi} + \frac{|\xi_f|}{\Xi}\right)$, which assigns the weighted average of contributed sales and purchases to each FF. To ensure individual and group rationality, the sum of the absolute values of the marginal profits is used. The marginal profit for buying requests for the total coopetition is $\Phi = \sum_{f \in F}|\phi_f|$, the marginal profit for selling requests for the total coopetition is $\Xi = \sum_{f \in F}|\xi_f|$. The amount FF *f* pays to the central planner is equal to $\max(0, \vartheta_f)$. So only if serving the requests in the bundle costs less than serving their initially offered requests, the FF pays the central planner. The total share of the extra profit FF *f* obtains from the central planner consists of two parts: 1) compensation for extra costs and 2) a share of the total collaboration gain (λ_f). The compensation of a FF is equal to the extra costs made by following the WDP assignment: $compensation_f = -\min(0, \vartheta_f)$. An example is shown in Table 4.

Table 4: Example of the profit-sharing mechanism.

FF	buys bundle	ϕ	ξ	ϑ	pay to CP	Compensation	λ = Extra Profit	Pay to FF
A	a	-4	-12	8	8	0	$\frac{10}{2} * \left(\frac{4}{16} = \frac{12}{26}\right) = 3.56$	3.56
B	b	-3	-8	5	5	0	$\frac{10}{2} * \left(\frac{3}{16} = \frac{8}{26}\right) = 2.48$	2.48
C	c	-9	-6	-3	0	3	$\frac{10}{2} * \left(\frac{9}{16} = \frac{6}{26}\right) = 3.97$	6.97
Total		16	26	10	13	3	10	13

4. Results

First, the three collaboration types are compared on the KPIs, as defined in Section 2.3. Second, the auction coopetition is analyzed on possible collaboration disadvantages, which results in a trade-off, which is discussed in Section 4.2. To ensure a valid comparison between the different types of collaboration, the allowed run time for each collaboration type is set to an equal amount of time in a real-world scenario (ART). If the ART is set to 60 minutes, this means that: The full collaboration can run for 60 minutes. All FFs can run their individual planning for 60 minutes each ($N_F \times 60$ in total). The auction coopetition can run for almost $N_F \times 60$ minutes. In a real-world auction coopetition scenario, the request selection and bidding phases would be executed in parallel by the participating FFs. The bundling, WDP, and profit-sharing phases are executed by the central planner. However, in this work, all calculations are performed by one party. To make the comparison fair, the computational time for the auction coopetition is set to $N_F \times \ ART$, because the bidding phase (phase 3) is by far the most time-consuming.

4.1. Comparison on Collaboration Efficiency

As mentioned in Section 2.3, the three different collaboration types are compared on the following KPIs: 1. Profit (Pr). 2. Distance traveled (Di). 3. Load factor (LF_{we} and LF_{wi}). 4. Waiting time for a dock to become available (WT). 5. Number of trucks (Tr). 6. Amount of truck arrivals at the GHs (TGH). The name of the instance indicates in order: the number of FFs, the number of GHs, and the number of requests in the data instance. The partition of the total amount of requests per FF is shown below the instance name. In Table 5 and Figure 6, the results for the 3_2_27 instance can be found.

Table 5. Results of instance 3_2_27.

Instance	ART	Type	Pr	Di	LF_{we}	LF_{wi}	WT	Tr	TGH	Pr per FF
3_2_27	10	I	127	99	68	61	47	7	9	[78, 13, 35]
[9, 11, 7]		A	202	86	67	61	0	7	7	[87, 58, 58]
		F	187	95	93	85	0	5	8	-
	60	I	139	98	78	71	38	6	9	[79, 25, 35]
		A	203	87	67	61	0	7	7	[85, 60, 58]
		F	195	92	93	85	0	5	8	-

As one can see in Table 5, the auction coopetition performs best on profit, distance, waiting time, and the number of trucks arriving at the GHs. The load factors of the auction coopetition and individual planning are almost the same, while the full collaboration has much higher load factors. The load factors directly correspond to the number of trucks. For the auction coopetition, the profit increase compared to the individual planning for FF2 stands out. All three collaboration types do improve, yet not significantly when the ART is increased from 10 to 60 minutes. For both the auction coopetition and the full collaboration, the waiting time is zero, while this is not the case for individual planning. Here the effect of a central planner can clearly be seen.

The profit of the auction coopetition is 46% higher than the individual planning while using more trucks. Another main difference can be found in the routing of the trucks. In the individual planning, each FF has to visit both GH1 and GH2, which causes three truck movements between the two GHs. In the auction coopetition, requests get reassigned to FFs in such a way that there are no movements between the two GHs. In the auction coopetition the central planner has information on which truck needs to be at a GH, at which time this information is shared in the bidding phase. Therefore, the central planner can decide to delay a truck at the origin depot, such that there are no two trucks arriving at one GH at the same time.

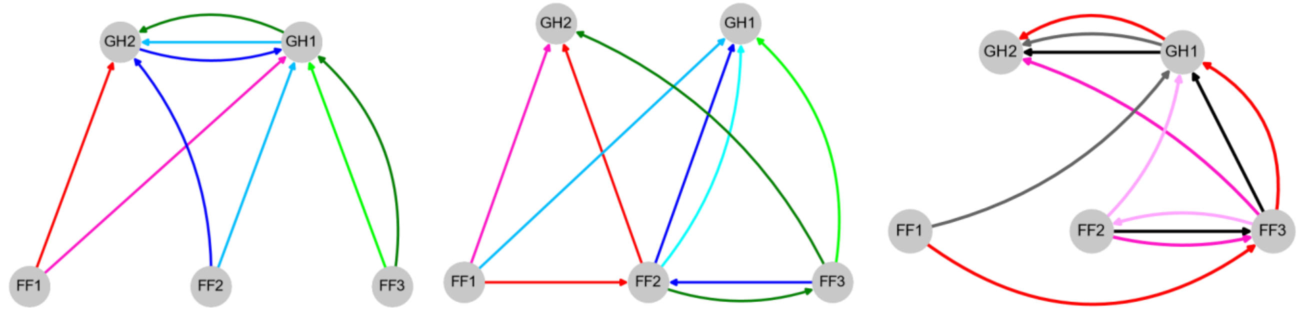

Figure 6. 3_2_27, ART = 60. Routing of the trucks: **I** (L), **A** (M), **F** (R). In **I** and **A**: tints of red = FF1, tints of blue = FF2, tints of green = FF3.

In the full collaboration, the number of trucks is the lowest. It is interesting to see that there are three truck movements between the GHs and four between the FFs. Apparently, the low amount of required trucks is compensated by more truck movements on both sides of the truck planning. Additionally, the full collaboration of the 3_2_27 instance is run with an exact solution method. The incumbent solution found after a six-hour run was a total profit of 219. This solution is roughly 8% better than the solution found with the auction coopetition model.

In Table 6, the results for the 4_3_50 instance are shown for the ART of 10 and 60 minutes. The auction coopetition performs best on profit, distance, waiting time, and truck arrivals at the GHs. Remarkably, in the full collaboration, the amount of used trucks goes up when increasing the ART from 10 to 60 minutes, even though the profit increases.

Similarly, in the auction coopetition the amount of truck arrivals at the GHs goes up, while the profit increases. These changes could be caused by the fact that the cost of the routing only depends on the total travel time and not on the truck use.

Table 6. Results for instance 4_3_50.

Instance	ART	Type	Pr	Di	LF_{we}	LF_{wi}	WT	Tr	TGH	Pr per FF
4_3_50 [12,10,14,14]	10	I	205	142	81	74	172	11	23	[-14, 57, 89, 74]
		A	406	127	74	67	0	12	12	[83, 91, 115, 116]
		F	235	183	87	80	33	10	21	-
	60	I	214	142	81	74	164	11	23	[-15, 43, 81, 106]
		A	424	117	72	66	0	12	14	[99, 90, 117, 117]
		F	246	184	58	53	16	15	24	-

4.2. Trade-off with Collaboration Disadvantages

In the auction coopetition the redistribution of the extra profit is done in such a way that every FF is compensated for their possible losses. Hence, individual rationality is guaranteed. The share of the extra profit that is obtained by every

FF is determined by their contribution (by buying or selling requests) to the coopetition. Consequently, the profit-sharing mechanism defined in this paper is considered fair, as defined in Section 2.4. The FFs do lose most of their autonomy over the requests they put into the auction pool. Most decisions of the CP are made in the best interest of the entire coalition. Overall, most extra actions required for fully functioning auction coopetition, concern data exchanges or relatively simple parameter specifications. Therefore, the auction coopetition is relatively easy to use for the FFs. It is important to notice that in the auction-based coopetition a FF is never required to share: 1. Their initial price to the shipper for handling a request (revenue per request). 2. Their initial profit. 3. Their initial number of requests. 4. Their profit of the bundles. 5. Their cost structure. Because the auction coopetition prevents the exchange of critical company information, there is no chance of losing your market position due to the exchange of information. However, it could be possible that implicit information can be retrieved from the redistribution of the requests. Further investigation of this phenomenon is required.

The auction coopetition seems to perform the best on several of the transport efficiency KPIs. Especially on the KPIs that are used as indicators for the number of truck congestion at the GHs, waiting time, and the number of truck arrivals at the GHs. The full collaboration also performs very well on the waiting time KPI. In individual planning, the waiting times are always the highest, which makes sense because there is no central planner to avoid queuing. The waiting time increases the costs for the FFs considerably. Because, in the individual planning, the FFs do not know the planning of the other FFs, it is possible that for one FF the waiting time is so high its profit becomes negative. On the forehand, there is no way for the FFs to know if this will occur, which causes uncertainty in their planning and their profit.

Both the decrease in congestion and the increase in certainty are clear indicators of the added value of a central planner.

Both in the auction coopetition and the full collaboration, the amount of total travel time, which waiting time is a part of, for all FFs is minimized. If there is still waiting time remaining in the planning, the associated cost is fairly divided over all collaborating FFs. This increases the certainty of the FFs and their profitability. Therefore, the profit reallocation mechanism is considered an advantage rather than a disadvantage.

The auction coopetition consistently performs the best in terms of profit. The total profit of the auction coopetition is the highest in every data instance run in Section 4.1. Additionally, the profit per FF seems to be divided more evenly across the FFs compared to the individual planning. Furthermore, the auction coopetition performs quite well in a short ART (10 minutes), where the full collaboration probably needs a lot more computational time to find an equally good solution. There are some downsides to the auction coopetition. In many of the instances from Section 2.3, the auction coopetition requires the most number of trucks. This causes the load factors to somewhat decrease, which is a KPI of the environmental impact. On the other hand, the auction coopetition does frequently perform the best on the distance KPI, which is also an environmental impact indicator.

5. Conclusion

The main contribution of this paper is twofold. First, the fully integrated five-phase auction-based coopetition model is a methodological contribution. A unique aspect is the fact that the dock capacity of the ground handlers is taken into account in an auction-based model. Also, the developed request selection and bundling procedures are based on the time windows of the request deliveries. Second, the potential of such auction-based coopetition is shown for an air cargo supply chain scenario. There is a clear increase in profitability for the freight forwarders and a decrease in congestion at the ground handlers. Furthermore, the total traveled distance decreases, which indicates a beneficial environmental effect. The disadvantages of the auction-based coopetition are limited due to the auction phases used in this paper. The data used in this paper is artificial. Therefore, it would require more research to establish the real-world potential of the auction-based coopetition system presented in this paper.

The solution space of the auction coopetition is reduced in the earlier phases, which probably causes the KPIs to only slightly improve by increasing the ART. Because the solution space is smaller, the optimal solution is worse than (or equal to) the optimal solution of the entire (full collaboration) problem but is found faster. Therefore, it seems likely that the auction coopetition model is able to find a decent solution within restricted computational time. It is important to note that if all three collaboration types are run for a sufficiently large time (until they reach optimality), the Full collaboration will find the best results. The auction coopetition will then find a solution just as good as or worse than the full collaboration. The individual planning will perform just as good as or worse than the auction coopetition. Notwithstanding this inherent hierarchy, an output of our work is that the fully collaborative scheme, not surprisingly, is the most challenging scheme from a computational perspective. While we approached it with a classic exact formulation solved via branch-and-bound, more sophisticated approaches might severely decrease the computational time and highlight the theoretical lower bound (optimal solution of the fully collaborative scheme). This theoretical lower bound is important to better assess the effective gap between the fully collaborative and the auction-based scheme.

Network flexibility is seen as the number of options a FF has to fulfill all transportation requests. With an increase in, for example, the number of initial requests (size of a FF), the amount of transportation options increases. Network flexibility becomes an issue when a FF can retrieve implicit information about the pricing of another FF by always winning their requests. The possibility of retrieving implicit information should be investigated further.

The difference in the results of the individual planning and the auction coopetition is larger than in the literature. This could be caused by the following factors: (1) The dock capacity of the GHs is taken into account, which is a unique feature of this paper. This causes a part of the differences (in profit) between the individual planning and auction coopetition. In larger instances, this difference further increases because more dock capacity violations occur in individual planning. (2) The simulated annealing approach chosen in this paper seems to perform the best for the auction coopetition. (3) The solution space of the auction coopetition is much smaller than for the individual planning and the full collaboration. Therefore, the individual planning and full collaboration would require more time to find an equally good solution as the auction coopetition.

The auction-based coopetition framework performed much better and with a lower computational burden than the full collaboration model. A future research direction would be to investigate which aspects of the auction model could also be used in a meta-heuristic to speed up the search in the full collaboration model and the individual planning. For instance, the bundling approach to find attractive bundles to load into the same truck can be used to improve the performance of the meta-heuristic. When there is limited improvement in the solution quality using this approach, a switch to the more flexible version where bundles can be broken up can be activated. In this paper, the hypothesis is that one of the collaborators of the auction coopetition could be able to acquire implicit information about the other participants. Another future research recommendation would be to investigate whether it is possible to acquire implicit information about the pricing of another collaborator/competitor in auction-based coopetition. In particular, profit-sharing implies that revenue is known to some extent by the auctioneer in an auction-based system. When this is not possible, a method could be devised that accounts for different scenarios in terms of revenue per stakeholder involved and computes a profit-sharing scheme that, on average, works well with all scenarios. Another research direction should address the inherent non-deterministic nature of some of the parameters used in this research, such as travel times, processing times, discrepancies between provided and actual information, in order to have a more robust framework that is better applicable for real-world problems.

References

Ankersmit, S., Rezaei, J., and Tavasszy, L. (2014) 'The potential of horizontal collaboration in airport ground freight services', *Journal of Air Transport Management*, 40, 169-181.

Berger, S. and Bierwirth, C. (2010) 'Solutions to the request reassignment problem in collaborative carrier networks', *Transportation Research Part E: Logistics and Transportation Review*, 46, 627-638.

Bombelli, A. and Tavasszy, L. (2021) 'The ground handler dock capacitated pickup and delivery problem with time windows (GHDC-PDPTW): mathematical formulations and solution method', (*Working Paper*).

Cruijssen, F., Cools, M., and Dullaert, W. (2007) 'Horizontal cooperation in logistics: opportunities and impediments', *Transportation Research Part E: Logistics and Transportation Review*, 43,129-142.

Ferrell, W., Ellis, K., Kaminsky, P., and Rainwater, C. (2020) 'Horizontal collaboration: opportunities for improved logistics planning', *International Journal of Production Research*, 58(14), 4267-4284.

Gansterer, M. and Hartl, R. F. (2016) 'Request evaluation strategies for carriers in auction-based collaborations', *OR Spectrum*, 38, 3-23.

Gansterer, M., Hartl, R. F., and Savelsbergh, M. (2020a) 'The Value of Information in Auction-Based Carrier Collaborations', *International Journal of Production Economics*, 221, 107485.

Gansterer, M., Hartl, R. F., and Sörensen, K. (2020b) 'Pushing Frontiers in Auction-Based Transport Collaborations', *Omega*, 94, 102042.

Johnson, S. C. (1967) 'Hierarchical Clustering Schemes', *Psychometrika*, 32, 241-254.

Li, B., Krushinsky, D., Van Woensel, T., and Reijers, H. A. (2016) 'An Adaptive Large Neighborhood Search Heuristic for the Share-a-Ride Problem', *Computers & Operations Research*, 66, 170-180.

Raweewan, M. and Ferrell, W. (2018) 'Information Sharing in Supply Chain Collaboration', *Computers & Industrial Engineering*, 126, 269-281.

Ropke, S. and Pisinger, D. (2006) 'An Adaptive Large Neighborhood Search Heuristic for the Pick-up and Delivery Problem with Time Windows', *Transportation Science*, 40(4), 455-472.

Schopka, K. and Kopfer, H. (2017) 'Pre-selection Strategies for the Collaborative Vehicle Routing Problem with Time Windows', In *Dynamics in Logistics*, pages 231-242. Springer, Cham, Switzerland.

Shaw, P. (1997) 'A New Local Search Algorithm Providing High-Quality Solutions to Vehicle Routing Problems', *APES Group, Dept of Computer Science, University of Strathclyde, Glasgow, Scotland, UK*, pages 1-12.

Verdonck, L., Caris, A., Ramaekers, K., and Janssens, G. (2013) 'Collaborative Logistics from the Perspective of Road Transportation Companies', *Transport Reviews*, 33, 700-719.

Wu, T. (2019) 'An Improved Optimization Method to Increase the Truck Planning Efficiency of Landside Air Cargo Operations', Master thesis, Delft University of Technology, The Netherlands.