Transportation Research Part E 72 (2014) 125–143

Contents lists available at ScienceDirect

Transportation Research Part E

journal homepage: www.elsevier.com/locate/tre

Integration of the cost allocation in the optimization of collaborative bundling

Christine Vanovermeire, Kenneth Sörensen *

University of Antwerp, Prinsstraat 13, 2000 Antwerpen, Belgium

ARTICLE INFO

Article history:
Received 2 September 2013
Received in revised form 26 August 2014
Accepted 21 September 2014

Keywords:
Collaboration
Logistics
Bundling
Cost allocation
Shapley value
Matheuristic

ABSTRACT

A model is proposed that integrates a cost allocation method – the Shapley value – into the optimization of the *synchronized* consolidation of transportation orders. By balancing each partner's delivery date changes (when synchronizing) against its *allocated* profit, it ensures that the operational plan is acceptable by all partners. In comparison to a model that first plans and then divides the costs, this model limits expensive delivery date changes and does not systematically favor a company with a slightly higher cost of change.

1. Introduction

The myopic optimization of a single functional entity in the supply chain can often have negative effects on the companies downstream or upstream (Ireland and Bruce, 2000). The bullwhip-effect for example Lee et al. (1997), demonstrates how self-optimization and non-communication can lead to an increased demand amplification along the chain. It is increasingly recognized that, in order to create more efficient supply chains, companies need to collaborate and become *partners* in a horizontal logistics *coalition*. As can be seen in Fig. 1, such collaboration can take on various forms. *Internal* collaboration entails cooperation between different functions in the same company (e.g., sales representatives communicate with the production site). *External* collaboration refers to cooperation between different companies and can take on two forms: vertical and horizontal. *Vertical* collaboration happens between a company and its suppliers or customers (e.g., Vendor Managed Inventory). The term *horizontal* collaboration is used to refer to cooperation between organizations on the same level of the supply chain, and even between competitors.

In an increasingly common form of horizontal collaboration, companies that have similar or complementary logistic needs combine their orders and transport them to their respective customers in a single logistic operation, rather than individually. The chosen means of transportation (truck, train, ...) can thus contain orders of several companies simultaneously. This is referred to as "order bundling" or simply "bundling".

A good business case for bundling occurs when companies are located in close proximity and share a significant number of clients. An example is the collaboration between two fast-moving consumer goods producers, Kimberly-Clark and Unilever-HPC. Kimberly-Clark was pressured by its retailers in the Netherlands to allow a very short-term replenishment

* Corresponding author.
E-mail addresses: Christine.Vanovermeire@uantwerpen.be (C. Vanovermeire), kenneth.sorensen@ua.ac.be (K. Sörensen).

http://dx.doi.org/10.1016/j.tre.2014.09.009
1366-5545/

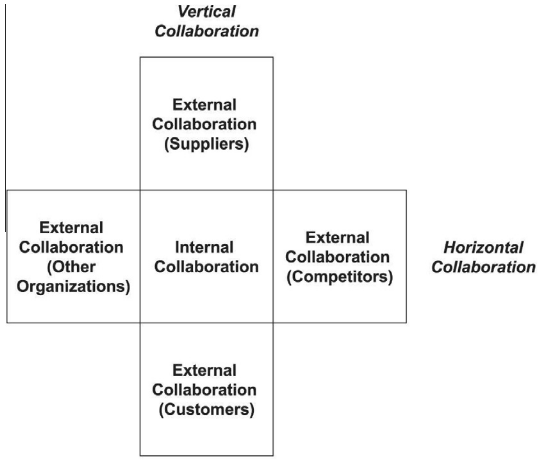


Fig. 1. The scope of collaboration (Barratt, 2004).

policy, which increased its delivery frequency and reduced its average shipment size. Rather than comply and face an increased transportation cost, Kimberly-Clark looked for another company that was shipping to those same retail stores and found one in Unilever-HPC, that shared 60 to 70% of its delivery addresses in the Netherlands. After a successful trial, both partners formed a coalition and started shipping their products from a shared warehouse to their mutual retail clients. To operate the shared distribution center and perform transportation on their behalf, a third-party logistic provider was added to the agreement. Order bundling in this coalition resulted in an increased service level (three delivery days per week instead of two), a 50% reduction in the number of trips, and a decrease in handling costs of 20% (Verweij, 2009; Cooke, 2011).

Another approach to profitable collaborative distribution can be found in the creation of so-called "shipping lanes", that arise when companies in the same region ship long-haul to another region. In a recent horizontal collaboration effort, plastics manufacturer JSP and metal forger Hammerwerk send out their trucks from the Czech Republic to Germany collaboratively, in collaboration with a third-party logistics provider they mutually agreed upon, and under the guidance of a neutral third party. Without increasing operational costs, they realize an increase in the number of days on which deliveries are made, while achieving a double-digit reduction in CO_2-emissions and reducing the inventory-in-transit cost (Guinouet et al., 2012).

In Europe, a framework supported by the European Union is currently being developed, entitled COllaboration COncepts for CO-modality (CO^3). The aim of this framework is to support horizontal collaboration efforts. It suggests the use of a *trustee*, an independent actor that helps to create and manage the collaboration and supports the partners in the coalition to maximize the synergy gains. Additionally, the use of a trustee limits the amount of information shared among competitors and remain compliant with European anti-trust law. CO^3 also provides a legal framework that describes the entry and exit clauses for a collaboration, and promotes a game-theoretical method (the Shapley value, see further) to divide the costs (Biermasz, 2012).

One of the criteria for exemption of European anti-trust law is that the agreement is indispensable to achieve the stated efficiency gains (Slaughter and May, 2012). This implies that the collaboration gains should exceed the economies of scale that can be achieved by traditional consolidation and groupage provided by many third-party logistics providers. In Vanovermeire et al. (2013a) and Biermasz (2012), it is argued that collaborative bundling differs from traditional consolidation or groupage in that it allows for the active *synchronization* of shipments. This implies that the coalition can decide to delay or expedite some orders of its partners if this results in a lower total transportation cost or is otherwise beneficial. In traditional consolidation and groupage, the third-party logistics provider has no impact on the timing of its customers' shipments, and is forced to ship them on the exact days specified by the customers.

The viability of collaborative bundling has been proven by TriVizor, a spin-off company of the University of Antwerp, Belgium.[1] This company currently manages several horizontal logistic coalitions as a neutral third party. It operates through a multi-party contract between itself, the shippers in the coalition, and a third-party logistics provider. TriVizor itself is payed for its services by a fixed percentage of the coalition gain (thereby increasing its own incentives to increase the efficiency of the coalition) and/or through a fixed amount per planned order. On a day-to-day basis, the partners in a coalition managed by TriVizor send their order data, i.e., the list of shipments they wish to send out, to a central database. Access to this database is restricted so that the partners do not have access to each other's order data. Using this data, the planners at TriVizor schedule the transports, which are then communicated to the third-party logistics provider for execution. TriVizor actively searches for bundling opportunities, but this is a manual and cumbersome process. For example, when the planners at TriVizor notice that a

[1] http://www.trivizor.be

shipment can be made more efficient if (part of) some orders are delayed or expedited, it contacts the planners at the coalition partners and asks for their approval (Van Breedam et al., 2013).

Notwithstanding the increased opportunities for logistic optimization, order synchronization does not come without a cost. More specifically, *flexible* partners, that allow their orders to be delivered on a different moment than the preferred one, will incur a (real or perceived) cost. This cost may be caused by higher inventories, decreased customer satisfaction, or any number of other factors. An important issue is that, although the benefits of order synchronization occur at the level of the coalition, it is the individual partners that incur the costs of their orders being delayed or expedited. For this reason, it is crucial that these benefits are distributed back to the partners in a way that is perceived as fair, i.e., that rewards the flexible company most.

When a coalition of companies engages in order bundling, they perform their daily distribution activities according to a joint operational plan. This operational plan is executed at a single coalition cost, that needs to be divided back to the individual partners. Ham et al. (2006) recommend that the issue of fair sharing of the coalition cost (or benefit) is addressed when negotiations start, and that a sharing rule is decided upon when preparing the pilot case. Biermasz (2012) also recommends to decide upon the cost allocation method before starting the operational distribution, and recommends the Shapley value to be used. All coalitions managed by TriVizor use a pre-determined cost allocation mechanism, which is usually based on some measure of the order volume (e.g., the number of pallets shipped). Although this way of working is simple and transparent, TriVizor have expressed their wish to implement a cost allocation method that induces partners in a coalition to behave in a way that benefits the coalition, i.e., in the most flexible way possible (Van Breedam et al., 2013).

As the cost allocation method is determined in advance, each operational plan is associated with one single cost allocation. However, it is possible that the delivery date changes of a partner are not adequately compensated by the allocated profit to that partner. This paper therefore suggests that, when automating and optimizing the operational planning process, the cost allocation mechanism has to be integrated into the optimization method. In this way, the decision to change an order's delivery date will be made dependent on the individually *allocated* benefit. Partners can then set thresholds below which they do not want an order to be delayed or expedited, and operational plans of which the cost allocation does not satisfy these individual thresholds can be excluded. The development of such a model, integrating the operational optimization *and* the cost allocation, is the topic of this paper.

The model proposed in this paper is the first to be truly *collaborative*, in the sense that it explicitly considers the optimization of the distribution effort *and* the cost or benefit allocation as a single decision. In this way it takes into account the fact that partners are still individual entities and provides solutions that satisfy all partners. In this paper, the first model that integrates a cost allocation method into an operational planning problem in the context of horizontal logistic collaboration is developed together with a simple heuristic to solve such an integrated collaborative model. The heuristic serves two purposes. First, it demonstrates that integrating a cost allocation method in an operational planning problem is — at least in principle and from an Operations Research point of view — possible. Secondly, it also shows that the resulting optimization problem is much more difficult to solve as it requires a large number of complicated intermediary computations. The results obtained in this paper are therefore emphatically not a demonstration of either the practical feasibility or the desirability of the idea, a challenge which is left for future research.

The remainder of this paper is organized as follows. In Section 2 we discuss the literature on cooperative supply chain planning. Section 3 discusses the Shapley value, a cost allocation method that will be used in our model. In Section 4, we develop a model for order bundling and cost allocation then discuss how this model can be turned into an integrated model through the addition of constraints that exclude operational plans in which some partners are not sufficiently compensated for their order delivery date changes. In Section 5, we develop a heuristic to solve the integrated collaborative model. The results of the model and heuristic are discussed in Section 6. Finally, Section 7 concludes and provides pointers for future research.

2. Cooperative supply chain planning

While the term Supply Chain Management is relatively new, the idea of coordinated planning is not. In an extensive review, Thomas and Griffin (1996) list many multi-echelon models, starting with the model of Clark and Scarf (1960), who optimize the total inventory along multiple installations. However, Thomas and Griffin (1996) argue that these models lack in the fact that no effort is done to find a solution that is *jointly* optimal. They suggest that the total *profit* rather than the total cost should be optimized. After optimization, the different sides have to negotiate how to divide the savings. Maximizing profit rather than minimizing cost is less frequently done, but still many examples can be found, such as Pal et al. (2012), Boyacı and Gallego (2002) and Zou et al. (2004). Özener et al. (2011) also maximize profit when optimizing the lane exchange problem for collaborating truckload carriers, and specify additional constraints that impose a minimal assignment of the trucks of each individual carrier to a specific lane.

Studying different collaboration approaches, Lehoux et al. (2010) indeed warn that none of them are necessarily equally profitable for all partners. They therefore emphasize the need for incentives (also called side payments) or a method to share the *total* profit to guarantee profit for both partners. Chen and Lee (2004), desiring to create a method that guarantees a fair profit distribution among all the participants of the supply chain — manufacturer, distributor and retailer —, propose a multi-objective approach. Rather than maximizing the total profit, the profit of each partner is maximized. The result of their algorithm is a Pareto-front, composed of a set of operational plans for which, given the profit level of one partner, the two other partners cannot obtain a higher profit. Of these operational plans, one can be chosen that will satisfy all partners.

In contrast to the previous models, Dudek and Stadtler (2005) state that centralized decision-making does not work in an environment with independent partners. They suggest a negotiation-based algorithm, that moves from a solution that is locally optimal for a buyer, to a solution upon which both partners can agree. They state that incentives are to be given to be able to reach the global optimum. Such incentives are discussed in Cachon (2003).

The use of decentralized decision-making and negotiation, such as suggested by Dudek and Stadtler (2005) is common in multi-agent systems. Davidsson et al. (2005) find an increasing number of logistic problems tackled with agent-based approaches. According to them, logistics is an excellent field to apply these approaches, as many of the problems consists of (1) well-defined entities with (2) complex behaviors and (3) information is not freely given. However, they warn that agent technology can also add unnecessary complexity. Despite its complexity, bidding by agents — a specific form of decentralized decision-making — allows to find a solution and a corresponding cost allocation that satisfies the constraints of all partners. This is shown by Conen (2002), who uses this mechanism to coordinate manufacturing and logistics and by Krajewska and Kopfer (2006) and Kwon et al. (2005), who use a combinatorial auction in carrier networks.

Finally, Berger and Bierwirth (2010) compare a centralized approach (full sharing of information) to a decentralized one (sharing information to a self-determined extent) in collaborative carrier networks. They find that the cost of decentralization is considerable, and suggest that the amount of centrally known data has to be widened.

When comparing decentralization to centralization in the multi-item replenishment problem, Chen and Chen (2005) find that a solution based on centralized decision-making is always superior to the solutions of the decentralized decision-making model. However, as in many centralized models, the gains are not equally spread. In this case, the retailer's cost even increases by approximately 10%, while the cost of the manufacturer drops almost 60%. To be able to satisfy all partners, quantity discounts are suggested.

Houghtalen et al. (2011) create models to support the collaboration between carriers exchanging capacity. First, inverse optimization is used to set capacity exchange prices to such a level that the optimal behavior is encouraged. Next, the routing decisions are centrally optimized. Having optimized the exchange prices in advance according to the behavior of the partners, the solutions of this centralized model are found to be acceptable for all partners.

The framework of Biermasz (2012) to support horizontal collaboration in logistics recommends a centralized approach with a neutral third party. Audy et al. (2007) describe several business models, managed by either a third party, a carrier, a customer, or several carriers and customers.

In line with these frameworks, our approach uses a centralized approach, finding a minimal *total* transportation cost. However, our model does allow individual partner preferences with respect to order delivery dates to influence the operational plan. In this way, the model avoids the additional complexity and overhead of agent-based decentralized models, while at the same time ensuring that the final operational plan satisfies all partners in the coalition. This approach can be readily applied to the many horizontal coalitions that currently exist, most of which use a neutral third party to perform the operational planning.

3. The Shapley value

One of the most important issues facing horizontal logistics coalitions is the division of the coalition cost between the partners (Cruijssen et al., 2007). A lot of papers have been written on this topic, many of them resulting in a new cost allocation method (Frisk et al., 2010; Liu et al., 2010; Soons, 2011; Audy et al., 2011).

One of the most studied methods is the Shapley value (Shapley, 1953). The Shapley value is a game-theoretical concept that allocates the total cost in a coalition based solely on the marginal costs of each partner in every possible subcoalition. The Shapley value can be calculated as follows. Assume a grand coalition N, which consists of partners $\{1, \ldots, i, \ldots, n\}$ and has a total transportation cost $c(N)$. Each possible subcoalition S has a transportation cost $c(S)$. The Shapley value for partner i is the weighted sum of all marginal costs of partner i in each subcoalition S:

$$\varphi_i = \sum_{S \subseteq N \setminus i} \frac{|S|!(n - |S| - 1)!}{n!} \times (c(S \cup i) - c(S)) \qquad \forall i \in N$$

The Shapley value has several desirable properties. Most importantly, the Shapley value always exists, and divides the total transportation cost among the different partners ($\sum_{i \in N} \varphi_i = c(N)$). Moreover, the cost allocated to each partner is never larger than its stand-alone cost ($\varphi_i \leqslant c(i)$).

According to Loehman and Whinston (1974), the Shapley value is completely "utilitarian" in that it only considers each partner's cooperative productivity (measured as the marginal profit) when determining its share of the gain. This stands in contrast to methods that also consider other factors (such as the number of orders, order sizes, ...). In Vanovermeire et al. (2013a) and Vanovermeire et al. (2013b), we show that, basing the cost allocation on the marginal costs, indeed gives the best incentives to increase cooperative productivity. Both papers discuss how incentives should be given to companies to behave in a flexible manner, i.e., allow time window constraints to be relaxed, as this results in an increased collaboration gain. In this paper, we use the Shapley value to give similar incentives to minimize the transportation cost by changing individual order delivery dates.

4. Sequential and integrated order synchronization with soft time windows and cost allocation

In this section, two models are developed to solve an order synchronization (or bundling) problem that differ in the way the cost allocation is handled. Both models assume that the coalition needs to bundle a set of orders in a set of trips, minimizing the total cost of the operation. The first model (which we will call the *sequential model*) assumes that the coalition first looks for the solution with the lowest cost and only then divides (using the Shapley value cost allocation) the cost of that solution. However, this solution and cost allocation might not satisfy some of the partners in the coalition. The second model (which we will call the *integrated model*) integrates the cost allocation into the model and finds a solution that has the lowest possible cost while ensuring that all partners are adequately compensated for their flexibility. A simple illustrative example of the difference between (the solutions found by) both models can be found in Section 6.1.

4.1. Order synchronization with soft time windows (sequential model)

Both models developed in this paper assume a horizontal logistics collaboration in which orders are shipped in *lanes* (i.e., long-haul point-to-point distribution, as seen in Breedam et al. (2011) or Vanovermeire et al. (2013b)). Starting from a common shared origin (warehouse, region, ...) goods are sent to a mutual destination. The goods are handled by a logistic service provider, that charges a price per pallet that decreases per additional pallet in a trip. A trip can contain 1 to *Cap* (maximum capacity) pallets.

The problem at hand is to plan the shipment of all orders in such a way that the total transportation cost is minimal. The coalition actively searches for bundling opportunities by changing order delivery dates to create a better fit between the deliveries of the different partners' orders. In other words, order delivery dates are actively *synchronized* within the limits of each order's time window.

The concept of time windows is common in the field of vehicle routing, where *hard time window* constraints are generally defined as continuous intervals (usually expressed in minutes or hours) during which a delivery can occur (Solomon et al., 1988). A solution (operational plan) in which an order is delivered outside of its hard time window is considered infeasible, and there is no cost for delivering an order anywhere within its time window. A more realistic approach is to set a preferred date or point in time for each order, but allow advancing or delaying orders to some degree. In this situation, *soft time windows* are introduced, in which a penalty cost is incurred when an order is delivered on any other moment than its preferred delivery moment (See, e.g., Taillard et al., 1997). The further away an order is delivered from its preferred delivery moment however, the less appealing this option is, and the larger the penalty cost incurred.

In our sequential model, a list of orders is assumed to be given, and each order is assumed to have a preferred delivery date f_i. An order which is delivered on this date incurs no cost. For each order, a cost function is defined that details the additional penalty costs incurred when the order is delivered on any other date. This cost function monotonically increases with the number of days the delivery of the order is removed from its preferred date. Without loss of generality, we assume in the rest of this paper that, for each order *i*, the marginal cost of delaying or expediting that order one day is the same value C_i. A hard time window is also introduced for each order, defined by the maximal number of days an order can be delayed (B_i) or expedited (A_i). Delivering an order outside of its hard time window $[f_i - A_i, f_i + B_i]$ is not allowed. In Fig. 2, this concept is visualized.

Next to its time window, each order also has an order size q_i expressed in number of pallets. The objective is to assign each order to a trip in such a way that the total number of pallets in each trip does not exceed the capacity *Cap*. Additionally, each trip should be scheduled on a specific day so as the minimize the total cost. A price list $c(p)$ is given that determines the cost of executing a trip containing p pallets.

The decision variables used in our model are the following.

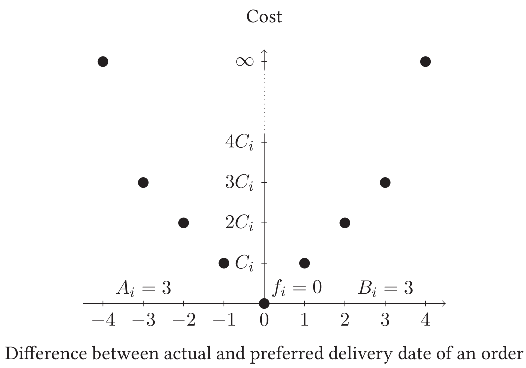


Fig. 2. Example of soft window constraints.

$$x_{ij} = \begin{cases} 0 & \text{if order } i \text{ is not in trip } j \\ 1 & \text{if order } i \text{ is in trip } j \end{cases}$$

$$z_{jp} = \begin{cases} 0 & \text{if trip } j \text{ does not have } p \text{ pallets} \\ 1 & \text{if trip } j \textit{ has } p \text{ pallets (and is therefore priced } c(p)) \end{cases}$$

$d_j \in \mathbb{N}$ the date of trip j

$e_i \in \mathbb{N}$ number of days order i is delivered earlier or later than its preferred delivery date

Using these decision variables, an MIP model of the bundling problem is the following.

$$\min \sum_j \sum_p c(p) z_{jp} + \sum_i C_i e_i \tag{1}$$

s.t.

$$\sum_j x_{ij} = 1 \quad \forall i \tag{2}$$

$$\sum_p z_{jp} = 1 \quad \forall j \tag{3}$$

$$\sum_i q_i x_{ij} \leqslant Cap \quad \forall j \tag{4}$$

$$Cap(1 - z_{jp}) \geqslant \sum_i x_{ij} q_i - p \quad \forall j, \forall p \tag{5}$$

$$d_j + M(1 - x_{ij}) \geqslant f_i - A_i \quad \forall j, \forall i \tag{6}$$

$$d_j - M(1 - x_{ij}) \leqslant f_i + B_i \quad \forall j, \forall i \tag{7}$$

$$e_i \geqslant f_i x_{ij} - d_j \quad \forall j, \forall i \tag{8}$$

$$e_i \geqslant d_j - f_i - (1 - x_{ij})M \quad \forall j, \forall i \tag{9}$$

The objective function (1) minimizes the total cost, which consists of the penalty cost of changing the delivery dates, and the cost of transporting the orders. Constraints (3) ensure that all orders are delivered exactly once. Moreover, all trips have to be priced exactly once, as per Constraints (3). When no pallets are transported, the cost of a trip is zero $c(0) = 0$. Constraints (4) enforce that the number of pallets in a trip does not exceed the capacity Cap. Constraints (5) ensure that a trip receives a price that is not too low according to the number of pallets it transports. With M representing a very large number, Constraints (6) and (7) guarantee that the delivery date of an order falls within its time window $[f_i - A_i, f_i + B_i]$. Constraints (8) and (9) determine the number of days the delivery date of an order has been changed away from its original delivery date.

As mentioned, the model developed in this section, which we have called the *sequential* model assumes that the cost of the solution found by solving the problem in Eqs. (1)–(9) is divided between the partners using the Shapley value cost allocation described in Section 3.

4.2. Order synchronization with soft time windows and an integrated cost allocation (integrated model)

Adding a penalty cost to the objective function does not guarantee an operational plan and a related cost allocation that will satisfy all partners (i.e., in which the benefit allocated to each partner exceeds the total cost of all its changed order delivery dates). The previous model can therefore not be considered a truly *collaborative* model as its solutions might be rejected by one or more of the partners.

To ensure that each partner is satisfied with the cost allocation, the coalition could divide the costs based on the sum of transportation and penalty costs. The Shapley value would indeed ensure that each partner now receives a share of the cost that is lower than its sum of transportation and penalty costs. However, it is not difficult to see that this way of working would remove all incentives to be flexible. Indeed, when a partner imposes very high penalty costs, the final solution will tend avoid changing the delivery dates of this partner. The partner will consequently incur low penalty costs and will be assigned a small fraction of the total cost. This way of working, in other words, strongly encourages partners to report high penalty costs for order delivery date changes, which is clearly an undesirable fact from the coalition's point of view.

To remedy this and create a model that will both satisfy all partners and provide an incentive to all partners to adopt a flexible attitude, we develop an *integrated* model in which changing the delivery date of an order not only decreases the *total* distribution cost, but also the *cost allocated to the owner of that order*. The model assumes that the solution will only be acceptable if *for each partner, its allocated share of the profit is at least as large as the total cost of its changed order delivery dates*. Solutions that do not satisfy this constraint, are considered infeasible. For an illustrative example of this principle, we refer to Section 6.1.

Besides resulting in solutions that will be acceptable to all partners, the integrated model has the additional advantage of not requiring the (penalty) cost of delivery date changes to be added to cost of transportation. As mentioned, it is often impossible to determine the exact costs of changing the delivery date of an order, mainly because this cost is related to such factors as increased inventory, customer satisfaction, and other parameters that may be hard to quantify. By adding these costs to the objective function, however, they are put on par with "real" costs (i.e., the money paid to the logistic service provider to transport the orders), which is undesirable.

In the integrated model developed in this section, the cost of changing delivery dates is considered a minimum threshold of profit that a partner expects to be awarded in exchange for its efforts. The integrated model minimizes only the transportation cost, and uses the penalty costs as a constraint on the number and size of the delivery date changes that a partner allows, enforcing an *allocated* profit that gives an adequate incentive to be flexible.

To be able to determine the profit allocated to a partner however, the cost allocation method has to be integrated into the model. A solution provided by this integrated model thus not only encompasses an operational plan that minimizes the transportation cost, but also the division of the total transportation cost. A feasible solution delivers all orders within the limits of the hard time windows and ensures that each partner is sufficiently compensated for its costs linked to its delivery date changes.

As the cost allocation is integrated into the model, the decision variables as defined in Section 4.1 do not suffice, and additional decision variables regarding the allocated cost need to be included. Moreover, to determine the *profit* a partner can make by being flexible (i.e., allowing its delivery dates to be changed), the allocated *cost when this partner would not have been flexible*, should be known as well. We therefore define two *states* for each partner. In the *rigid* state, its delivery dates are fixed to their preferred values and cannot be changed. In other words, A_i and B_i (the upper and lower limit of the time windows of orders i belonging to this partner) are set to zero. In the *flexible* state, the delivery dates of its orders can be changed (within the limits of the hard time windows). In the final solution of the integrated model, all partners are in the flexible state.

To denote the states of all partners in the coalition S, we define a *state set* $U = \{u_1, u_2, \ldots, u_k, \ldots, u_{|S|}\}$, with for each partner $k \in S, u_k = \text{rigid}$ or $u_k = \text{flex}$. We say that a state set $V = \{v_1, v_2, \ldots v_k, \ldots, v_{|S|}\}$, is a *subset* of U (and denote this as $V \subseteq U$) when for each partner k, its state in V is either rigid or equal to its state in U.

The sequential model in Section 4.1 is adapted in the following manner. First, additional decision variables to integrate the cost allocation in the model, are included.

$\varphi^U_{k\in S}$ allocated cost to partner k given the total cost of subcoalition S and assuming the state set U

Secondly, the objective function no longer contains a penalty cost.

$$\min \sum_j \sum_p c(p) y_{jp}$$

Indirectly, these penalty costs are still incorporated in the integrated model, by including additional constraints, stating that the profit allocated to a partner k should exceed its costs of changing delivery dates. As stated previously, to be able to determine the allocated *profit* for a partner, the allocated cost when this partner is in the rigid state has to be known. This allocated cost is called that partner's *baseline*, and should be at least as large as the allocated cost to that partner when it is in the flexible state.

Determining the baseline is not trivial, as there are several state sets U in which partner k can be rigid, and each of these situations will result in a different allocated cost to partner k. More specifically, each partner in the coalition S can freely choose whether they prefer to be flexible or rigid. Evidently, each partner can only choose its own state (rigid or not) and not the state of the other partners. We therefore define the baseline to be the highest allocated cost to partner k for all possible state sets of the remaining partners in that subcoalition S.

$$\text{baseline for partner k} \in \text{S for } U = \max_{V \subseteq U | v_k = \text{rigid}} \left(\varphi^V_{k\in S}\right)$$

The additional constraints, which we call *Shapley constraints*, can be formulated as follows:

$$\max_{V \subseteq U | v_k = \text{rigid}} \left(\varphi^V_{k\in S}\right) - \varphi^U_{k\in S} \geqslant \sum_{\text{orders } i \text{ of } k} C_i e_i \quad \forall k, \forall S, \forall U \tag{10}$$

The Shapley constraints enforce a solution to be feasible in the sense that all partners receive a share of the profit that is larger than the cost of order delivery date changes. By choosing as the baseline for a partner k the *highest* possible allocated when k is rigid, it will always receive a smaller cost when in the flexible state. The effect on its allocated profit of a partner becoming flexible will therefore always be positive. This will avoid the following undesirable effects.

- An alternative baseline would be the situation in which all partners are rigid. In this case, however, the cost allocated to partner k in the final solution (in which all partners assume a flexible state) can be higher than the one in the baseline allocation. This situation occurs if the cost effect of the other partners becoming flexible is extremely large and they therefore receive a share of the coalition cost so small that the remaining cost allocated to partner k is larger than in the baseline situation when all are rigid. Partner k will therefore prefer the solution in which all partners are rigid. Therefore, the only feasible solution — i.e., the only solution that would be accepted by partner k — would be the solution in which all partners are rigid. This leads to a suboptimal solution, enforced by one of the partners.

- If we would choose as the baseline for partner k and for another partner l the allocated costs in the situation in which all partners are flexible with exception of respectively partner k and partner l, another problem might arise. Using this choice of baseline, is possible that the allocated profits to partners k and l when they are flexible do not exceed their cost of changing delivery dates. In this case, partner k as well as partner l will prefer to remain in the rigid state. However, when both assume a rigid state, the total cost is higher than in the situation where one of the partners has a flexible state. Consequently, the allocated costs to both partners in the situation when both are rigid might be higher than the allocated cost when one partner is flexible as well. As partner k for example will not allow its delivery dates to be changed, the right hand side of Constraints (10) equals zero. In order for the left hand side to be greater or equal than zero, partner k needs partner l to be flexible. However, this is not possible, as this will lead to a solution in which the profit allocated to partner l does not exceed the cost of changing its delivery dates. In this case, the model is infeasible and no solution is possible.

Constraints (10) reveal the computational complexity of the integrated model. To be able to calculate the Shapley value for a given coalition S, the costs of all subcoalitions of S has to be calculated as well. Moreover, to determine the maximal cost allocation to a rigid partner k (i.e., the baseline situation), each situation in which partner k is rigid has to be evaluated. To determine all of these costs, for each subcoalition and each state sets U, an operational plan has to be constructed.

5. A heuristic for order synchronization with soft time windows and an integrated cost allocation

5.1. Outline of the heuristic

In this section we explain, in some detail, a heuristic to solve the integrated model developed in Section 5. The heuristic is able to synchronize orders of several partners having soft time windows, while integrating the Shapley value cost allocation method. The solution found by this heuristic does not only consist of an operational plan with a low transportation cost, it also encompasses the division of the total cost between the partners. The integration of the Shapley constraints makes the underlying problem (and therefore the heuristic) considerably more complex than their sequential versions, and requires a sizeable number of auxiliary calculations to be performed. The main aim of this paper is therefore not to develop an algorithm that has the required properties to be useful in practice (fast, robust, scalable, ...). Rather, we aim to show that the *solutions* found using the integrated model are more useful in practice. The algorithm developed in this section should therefore be considered a proof-of-concept implementation intended to clarify the necessary calculations. We leave the development of algorithms ready for real-life implementation for future research.

The complexity of solving the integrated model is essentially a result of the integration of the Shapley value calculations in the optimization. As the feasibility of the final solution (operational plan) depends on the amount of profit allocated to each partner, the cost allocation has to be (re) calculated *during* the optimization phase. However, to be able to calculate the Shapley value, information regarding the costs of all *subcoalitions* has to be available as well, and thus, the operational plans of those subcoalitions need to be constructed too. This statement can be extended as follows. To be able to construct the solution for a coalition N, the costs of that solution in all subcoalitions of N, with a number of partners smaller than $|N|$ should be known too. The procedure starts from an empty coalition is (with cost zero), and then proceeds to create the operational plan and calculate the resulting cost for all stand-alone coalitions. Then, it does the same for all coalitions containing two partners, and so on. Note that partners can not change their reported penalty costs on a per-subcoalition basis and that the cost of the subcoalitions is based only on information found in operational plan of the final (grand) coalition (i.e., deliver dates for each order of each partner in the final coalition and penalty costs C_i for each order).

Moreover, for each subcoalition, the *baseline* has to be set to be able to determine the *profit* (i.e., the difference between the stand-alone cost and the allocated coalition cost) allocated to each partner. This implies that for each subcoalition, the operational plans for all different possible state sets need to be created. The procedure starts from the state sets in which all partners are rigid (no partner allows its delivery dates to be changed), to the state sets in which only one partner is flexible, to the final plan in which all partners allow their delivery dates to be changed. This procedure is shown with an example for a coalition with three partners a, b and c:

Step 1 Create operational plan for a =rigid, b =rigid, c =rigid

Step 2 Create operational plan for a =flex, b =rigid, c =rigid

baseline $a = \varphi_a^{a=\text{rigid},b=\text{rigid},c=\text{rigid}}$

Create operational plan for a =rigid, b =flex, c =rigid

baseline $b = \varphi_b^{b=\text{rigid},a=\text{rigid},c=\text{rigid}}$

Create operational plan for a =rigid, b =rigid, c =flex

baseline $c = \varphi_c^{c=\text{rigid},a=\text{rigid},b=\text{rigid}}$

Step 3 Create operational plan for a =flex, b =flex, c =rigid

baseline $a = \max\left(\varphi_a^{a=\text{rigid},b=\text{rigid},c=\text{rigid}}, \varphi_a^{a=\text{rigid},b=\text{flex},c=\text{rigid}}\right)$

$$\text{baseline } b = \max\left(\varphi_b^{b=\text{rigid},a=\text{rigid},c=\text{rigid}}, \varphi_b^{b=\text{rigid},a=\text{flex},c=\text{rigid}}\right)$$

Create operational plan for a =flex, b =rigid, c =flex

$$\text{baseline } a = \max\left(\varphi_a^{a=\text{rigid},b=\text{rigid},c=\text{rigid}}, \varphi_a^{a=\text{rigid},b=\text{flex},c=\text{rigid}}\right)$$

$$\text{baseline } c = \max\left(\varphi_c^{c=\text{rigid},a=\text{rigid},b=\text{rigid}}, \varphi_c^{c=\text{rigid},a=\text{rigid},b=\text{flex}}\right)$$

Create operational plan for a =rigid, b =flex, c =flex

$$\text{baseline } b = \max\left(\varphi_b^{b=\text{rigid},a=\text{rigid},c=\text{rigid}}, \varphi_b^{b=\text{rigid},a=\text{flex},c=\text{rigid}}\right)$$

$$\text{baseline } c = \max\left(\varphi_c^{c=\text{rigid},a=\text{rigid},b=\text{rigid}}, \varphi_c^{c=\text{rigid},a=\text{rigid},b=\text{flex}}\right)$$

Step 4 Create operational plan for a =flex, b =flex, c =flex

$$\text{baseline } a = \max\left(\varphi_a^{a=\text{rigid},b=\text{rigid},c=\text{rigid}}, \varphi_a^{a=\text{rigid},b=\text{flex},c=\text{rigid}} \varphi_a^{a=\text{rigid},b=\text{flex},c=\text{flex}}\right)$$

$$\text{baseline } b = \max\left(\varphi_b^{b=\text{rigid},a=\text{rigid},c=\text{rigid}}, \varphi_b^{b=\text{rigid},a=\text{flex},c=\text{rigid}}, \varphi_b^{b=\text{rigid},a=\text{flex},c=\text{flex}}\right)$$

$$\text{baseline } c = \max\left(\varphi_c^{c=\text{rigid},a=\text{rigid},b=\text{rigid}}, \varphi_c^{c=\text{rigid},a=\text{rigid},b=\text{flex}}, \varphi_c^{c=\text{rigid},a=\text{flex},b=\text{flex}}\right)$$

The result is an iterative procedure that moves from an empty subcoalition to the full coalition, and, in each iteration, from a completely rigid solution to a solution in which all partners allow their delivery dates to be changed (under the condition that the cost of changing dates has to be compensated). An outline of the complete procedure is depicted in Algorithm 1.

Algorithm 1: A heuristic for order synchronization with soft time windows and Shapley constraints

```
t ← 0
while t ≤ n do
    forall the subcoalitions S with t partners do
        forall the possible state sets U in subcoalition S, with a flexible partners do
            Solve an exact model (Section 5.3)
            archive ← Check feasibility (Section 5.2)
            while there are violated constraints do
                Swap orders (Section 5.4)
                archive ← Check feasibility
                Solve an exact model
                Relocate orders (Section 5.5)
                archive ←Check feasibility
                Solve an exact model
                Random improvements (Section 5.6)
                archive ← Check feasibility
                Solve an exact model
            Select solution with the lowest transportation cost from the archive of feasible solutions
        t ← t + 1
```

For each subcoalition and for each given combination of rigid and flexible partners, a feasible solution is found with the lowest transportation cost. The heuristic design chosen is a Variable Neighborhood Search, as this concept is simple (easy to understand), as well as very effective (Hansen and Mladenovi, 2001). To create feasible solutions, four neighborhoods are developed, a detailed explanation of which can be found in Sections 5.3–5.6. Each neighborhood is iteratively explored until no feasibility (measured as the total violation of the Shapley constraints – see Section 5.2) improvement is found. The first three neighborhood perform a greedy search (finding a local optimum). This allows to find a feasible solution fast. However, as it does not guarantee to find the global optimum, a randomized search is included as well to further explore the solution space (Section 5.6). Finding the global optimum however is still not guaranteed.

The feasibility of a solution and the procedure to check it are discussed in Section 5.2. Whenever a feasible solution is found, it is stored in the archive. At the end of each iteration, the solution with the lowest transportation cost is selected from the archive.

5.2. Check feasibility

Changing delivery dates while disregarding the cost of such a change, might lead to a solution that violates the Shapley constraints (10). We check whether these constraints have been violated by the calculation of $violation_k$ for each partner k:

$$violation_k = \max_{V \subseteq U | v_k = \text{rigid}} (\varphi^V_{k \in S}) - \varphi^U_{k \in S} + C_i \sum_{\text{orders } i \text{ of partner } k} |d^{i*} - f_i|$$

Symbol d^{i*} is used to specify the date d_j of the trip that contains order i in the current solution.

The level of $violation_k$ is thus determined by the difference in the baseline of partner k and the cost allocated to partner k by the Shapley value, given the cost of the currently proposed operational plan, and the cost of the delivery date changes of partner k in the operational plan.

If one of the partners has violated constraints ($violation_k > 0$), the solution is infeasible and we try to repair the feasibility by either an exact procedure that does not consider the soft time windows (see next section), or by heuristically swapping or relocating orders. If the solution is feasible, it is stored in the archive, from which the best solution will be selected.

5.3. Solve an exact model

Each iteration starts with a solution that considers the hard time windows, but disregards the soft time windows. In other words, this solution takes into account the upper and lower limits of the time windows, but does not consider the information regarding the costs that each partner assigns to changing its delivery dates. The aim of this stage is to find a good solution from which to start the local search. This is achieved by solving a special case of the sequential model of Section 4.1. This modified model minimizes the number of changes that are made to delivery dates, and is constructed by setting $C_i = \varepsilon$, with ε being a very small number. Additionally, lower and upper limits A_i and B_i are set to zero for all orders i that belong to a rigid partner in the state set.

This model is not only solved at the beginning of each iteration, but every time a feasible solution is found (see Algorithm 1). The aim of this step is to find a similar solution with a lower transportation cost. The similarity of the solution found by the exact model is guaranteed by taking into account the moves made by the other neighborhoods. More specifically, when an order i is either swapped, relocated, or a random improvement has occurred involving this order, this change is stored and added to this model as an additional constraint. This constraint stipulates that the number of days the delivery date of the order i that has been swapped or relocated, cannot be changed further away from its original delivery date than in the current operational plan.

$$e_i \leqslant |d^{i*} - f_i| \tag{11}$$

The exact model is solved using the exact solver Gurobi (http://www.gurobi.com). The additional computational overhead of using this exact procedure is warranted, as it guarantees that the *submodel* is solved to optimality. Because of this fact, possible differences in solutions of the integrated and the sequential models can be linked with more certainty to the presence or absence of the Shapley constraints, rather than the ability of the algorithm to find the optimal solution.

5.4. Swap orders

The first local search operator attempts to find better solutions by swapping pairs of orders. Swapping orders implies that an order of one trip is interchanged with an order in another trip. It is important to note that the initial operational plan, before the swapping procedure starts, is the solution of the exact sequential model defined in Section 5.3. This solution is a lower bound on the solution that includes the Shapley constraints. The objective of the swapping procedure is to rearrange the division of changes in delivery dates among the partners with a minimal cost increase.

The swapping procedure works as follows. First, The orders of the partner that has the highest violation (partner *vltn*) are swapped. By calculating for each order of partner *vltn* its contribution to the violation ($orderViolation_i$), we determine the order in which we try to optimize each order (highest $orderViolation_i$ first). As relocating an order generally increases the transportation cost more than swapping orders, the orders are first swapped and in a later stage relocated.

To calculate $orderViolation_i$, a baseline has to be determined. Contrary to the total violation $violation_k$, the baseline used to determine $orderViolation_i$ investigates only one state set, i.e. the situation in which in the state of all partners is equal to their state in the current solution, with the exception of the partner that owns the order i, that is always rigid. This state set is symbolized by U^*.

In the baseline as well as in the currently proposed operational plan, the algorithm finds the trip that contains order i. To denote the specific trip in the baseline U^* containing order i, we use the notation $j|i^{U^*}$, and the trip in the current operational plan is denoted by $j|i^U$.

By summing the order sizes of all the orders in this trip, we determine the total number of pallets in the trips and consequently, the price per pallet that order i is subject to in both the baseline and the current operational plan. The $orderViolation_i$ of order i can then readily be determined by calculating the difference in prices per pallet multiplied by the number of pallets of that order, plus the additional costs of changing the delivery date of order i. Note that the price difference between the rigid situation and the new situation is forced to be equal or larger than zero. As we have shown that the solution provided by solving the sequential model in exactly can be considered a lower bound, this makes sense. Although the *individual* cost of that specific trip might be higher, the *total* transportation cost (on which the cost allocation is based) is as low as possible in this operational plan. This implies that there is no transportation cost decrease by allocating order i to a different trip. The main focus of the *orderViolation* is thus on orders of which the delivery date, has been changed, but for which the transportation cost decrease due to the change is not able to cover the additional cost of it.

$$orderViolation_i = \max\left[0, p\left(\sum_{\text{orders } a\in j|i^{U^*}} q_a\right) - p\left(\sum_{\text{orders } b\in j|i^{U}} q_b\right) q_i\right] - C_i|d^{i*} - f_i|$$

In order to swap an order, we look for viable candidate orders to swap it with. For all candidate orders o of all trips t that have delivery dates d_t within the time span defined by the upper and lower limit of the order to be swapped, being $[f_i - A_i, f_i + B_i]$, the feasibility (limited to the upper and lower limits and the capacity constraint) is checked, and the possible profit for partner *vltn* ($profit_{o,vltn}$) is calculated.

The possible profit (or loss) for partner k consists of two parts. The first part is the reduction (or increase) in the number of days an order is placed further away from its original delivery date. A second part consists of the price difference achieved by removing order i and adding order o. This price difference is dependent on the number of pallets $q^{old} = \sum_{a\in j|i^U} q_a$ of the trip that contains order i and the new quantity in trip t $q^{new} = \sum_{b\in t} q_b - q_o + q_i$ when order i is swapped with order o.

$$profit_{o,k} = C_o[|f_o - d_j| - |f_o - d_u|] + [p(q^{old}) - p(q^{new})]q_o$$

However, it is not only the profit for partner *vltn* that has to be calculated. The partner that order i is swapped with will also experience a difference in cost. The profit (or loss) for this partner is calculated by the same formula. However, in this case, order i is the candidate order.

To determine which order to swap with, a weighted total profit is used. As swaps that produce a high profit to partners with higher violated constraints are more desirable than swaps for partners that do not (and vice versa, swaps that produce a loss for partners with constraints (10) are to be avoided), the total profit is the sum of the profit for both partners, but weighted with the ratio of the positive $violation_k$ of the partner that owns order i on the sum of the positive $violation_k$ of all the partners of subcoalition S.

$$1 + \frac{\max(0, violation_k)}{\sum_{k\in S}\max(0, violation_k)}$$

The order with the highest weighted total profit is swapped with the order with the highest violation. The profit for partner *max* that this swap causes is deducted from $violation_k$. We continue to swap orders of partner *max* until there are no more orders i that have $orderViolation_i > 0$, or until $violation_k \leqslant 0$. Note that when formulating constraints in the optimization-phase, for each swap, only one constraint limiting the number of changes to the delivery date (in number of days) of the order of partner *max* is added.

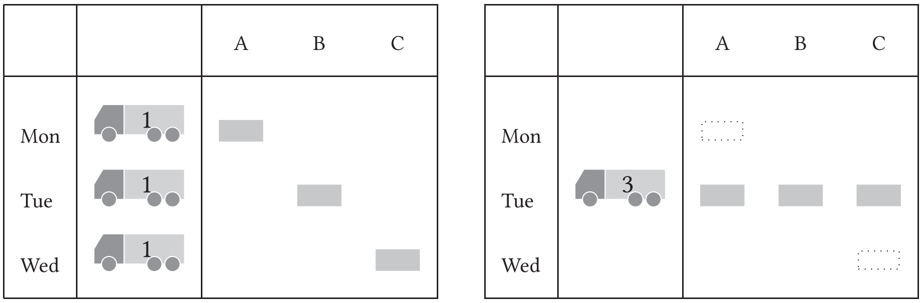
When a new partner l has the highest $violation_l$, the heuristic restarts the swapping procedure, swapping orders of this new partner *vltn*. If partner *vltn* did not change, the heuristic continues to the next phase, which is moving orders. The structured code of the procedure of swapping orders can be found in Algorithm 2.

Algorithm 2: Order swapping procedure

$vltn \leftarrow 0$
$violation_{vltn} \leftarrow 0$
forall the *partners k in S* **do**
  calculate violation$_k$
  if $violation_k > violation_{vltn}$ **then**
    $vltn \leftarrow k$
while *there is a* $violation_k > 0$ *and vltn has changed* **do**
  sort list of orders i of partner with highest violation $vltn$ in descending order of $orderViolation_i$
  $i \leftarrow 0$
  while $i <$ *number of orders of partner vltn and* $orderViolation_i > 0$ *and* $violation_{vltn} > 0$ **do**
    create list of feasible candidates o
    swap with highest $\frac{1+\max(0, violation_{vltn})}{\sum_k \max(0, violation_k)} profit_{o,i} + \frac{1+\max(0, violation_{\text{partner of order } o})}{\sum_k \max(0, violation_k)} profit_{i,o}$
    $violation_{vltn} \leftarrow violation_{vltn} - profit_{o,i}$
  calculate cost and divide it among partners
  forall the *partners k in S* **do**
    calculate violation$_k$ and determine partner $vltn$

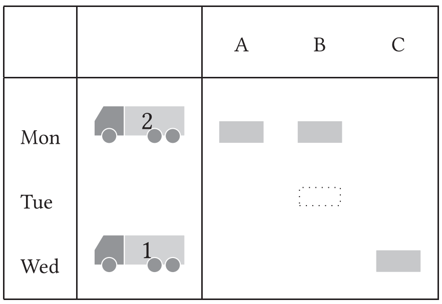
5.5. Relocate orders

If the solution proposed by the swapping procedure, followed by solving it with an exact model, does not provide a feasible solution, the Relocate procedure starts. The objective of this procedure is to change the delivery dates of the orders of the partners for which the Shapley constraints are violated, and locate these orders closer to their original delivery date without changing the delivery dates of the other orders.



(a) Operational plan when no delivery dates are changed

(b) Operational plan according to the sequential model



(c) Operational plan according to the integrated model

	Total cost
Situation (a)	89.85
Situation (b)	68.64
Situation (c)	80.93

(d) Resulting total costs according to the pace list in Table 5

Fig. 3. Example of the difference between minimizing the transportation cost with or without integrating the cost allocation method.

Table 1
Additional costs of changing delivery dates, cost allocation and profit for the integrated model.

	Minimum compensation for changed delivery dates	Stand-alone cost	Allocated cost	Profit
A	0	29.95	28.463	1.487
B	4	29.95	24.003	5.947
C	0	29.95	28.463	1.487

Table 2
Additional cost of changing the delivery dates of the orders for the sequential model, and possible cost allocations and profit.

	Cost of changes	Shapley allocation		Pay cost of changes	
		Allocated cost	Profit	Allocated cost	Profit
A	4	24.366	5.584	25.213	4.737
B	0	19.906	10.044	24.213	5.737
C	10	24.366	5.584	19.213	10.737

Table 3
Difference between synchronization with soft time windows and synchronization with soft time windows and an integrated cost allocation, first instance set.

	Diff total transp. cost	Diff $\sum e_{all}$	Diff φ_A	Diff $\sum e_A$
$A_A = B_A = A_B = B_B = A_C = B_C = 1, C_A = C_B = C_C = 10$				
Instance 1	0.00	0	0.00	0
Instance 2	0.00	0	2.19	−1
Instance 3	−5.45	1	−2.72	0
Instance 4	−2.35	1	−1.18	0
Instance 5	0.00	0	0.00	0
Instance 6	0.00	0	0.00	−1
Instance 7	−25.82	4	−13.21	1
Instance 8	−21.85	3	−7.51	2
Instance 9	0.00	0	0.00	0
Instance 10	0.00	0	0.00	0
$A_A = B_A = 5, A_B = B_B = A_C = B_B = 1, C_A = 50, C_B = C_C = 10$				
Instance 1–1	135.08	−2	45.03	−2
Instance 2–1	136.42	−3	70.61	−2
Instance 3–1	−5.45	1	−2.72	0
Instance 4–1	−2.35	1	−1.18	0
Instance 5–1	0.00	0	−3.53	0
Instance 6–1	0.00	0	0.00	0
Instance 7–1	0.00	0	−6.84	0
Instance 8–1	−1.34	1	−0.67	0
Instance 9–1	0.00	0	0.00	0
Instance 10–1	0.00	0	−24.03	1
$A_A = B_A = A_B = B_B = A_C = B_C = 1, C_A = 11, C_B = C_C = 10$				
Instance 1–2	0.00	0	0.00	0
Instance 2–2	0.00	0	2.19	0
Instance 3–2	−5.45	1	−2.72	0
Instance 4–2	−2.35	1	−1.18	0
Instance 5–2	0.00	0	0.00	0
Instance 6–2	0.00	0	0.00	0
Instance 7–2	−25.82	4	−13.21	1
Instance 8–2	−21.85	3	−7.51	2
Instance 9–2	0.00	0	0.00	0
Instance 10–2	0.00	0	0.00	1

For each order i of partner *vltn* and in order of decreasing $orderViolation_i$, we remove order i from its trip. Order i is then added to an existing trip or a new trip is created for order i, depending on which alternative produces the lowest cost (transportation cost plus cost of changing delivery dates). This is repeated until $violation_{vltn} = 0$ or until each order has been relocated.

If partner *vltn* remains unchanged, the algorithm starts the final procedure, in which the feasibility of the solution is improved by making random moves. If another partner has the highest violated constraints, the swapping procedure is restarted.

5.6. Random improvements

If the solution is still not feasible, a final step consists of randomly improving the solution. In this step, we find the partner with the highest violation, partner *vltn* and try to repair its feasibility by limiting the number of days the delivery date of a random order of that partner can be changed away from its original delivery date.

More specifically, the algorithm forces the maximal delivery date change of a random order of partner *vltn* to be limited to the number of days the delivery date is changed in the current solution, minus one. This limitation is enforced by adding a constraint to the model in Section 5.3.

$$e_i \leqslant |d^{i*} - f_i| - 1 \tag{12}$$

When no feasible solution is found by the random improvements procedure, the swapping procedure restarts from the solution found after the random move.

6. Results

6.1. Illustrative example

The difference between the sequential model (Section 4.1, minimizing the sum of transportation cost and penalty cost) and the integrated model (Section 4.2, minimizing the transportation cost under the Shapley constraints) is easily demonstrated with the simple example shown in Fig. 3.

In this example, three partners (A, B and C) each deliver one pallet on a different day, respectively Monday, Tuesday and Wednesday. The main difference between the partners is the upper and lower bounds they impose, as well as their soft time window constraints. Partners A and B do not allow their delivery dates to change for more than one day. Partner C on the other hand sets its upper and lower limit to two days. However, the cost linked to changing the delivery date is a lot higher:

Table 4
Difference between synchronization with soft time windows and synchronization with soft time windows and an integrated cost allocation, second instance set.

	Diff total transp. cost	Diff $\sum e_{all}$	Diff φ_A	Diff $\sum e_A$
$A_A = B_A = A_B = B_B = A_C = B_C = 3, C_A = C_B = C_C = 20$				
Instance 1	0.00	0	0.00	0
Instance 1–3	0.00	0.00	0.00	0.00
Instance 2–3	0.00	0.00	2.40	1.00
Instance 3–3	−6.33	3.00	−3.16	0.00
Instance 4–3	−30.24	5.00	−1.17	0.00
Instance 5–3	0.00	0.00	0.00	0.00
Instance 6–3	−0.51	4.00	12.58	−1.00
Instance 7–3	−27.57	4.00	−22.32	4.00
Instance 8–3	−1.34	1.00	−18.15	0.00
Instance 9–3	0.00	0.00	3.14	0.00
Instance 10–3	−34.27	4.00	−15.28	2.00
$A_A = B_A = 5, A_B = B_B = A_C = B_B = 3, C_A = 5, C_B = C_C = 20$				
Instance 1–4	0.00	0.00	0.00	−3.00
Instance 2–4	0.00	−1.00	2.19	−4.00
Instance 3–4	−6.33	2.00	−3.16	0.00
Instance 4–4	−30.24	2.00	−1.17	−3.00
Instance 5–4	0.00	0.00	0.00	0.00
Instance 6–4	−0.51	4.00	6.96	−1.00
Instance 7–4	−0.07	1.00	−3.24	1.00
Instance 8–4	0.00	0.00	−30.37	0.00
Instance 9–4	−6.30	4.00	−5.60	1.00
Instance 10–4	−24.34	5.00	−12.23	2.00
$A_A = B_A = A_B = B_B = A_C = B_C = 3, C_A = 19, C_B = C_C = 20$				
Instance 1–5	0	0	0	−2
Instance 2–5	0	0	2.397	0
Instance 3–5	−6.33	3	−3.165	0
Instance 4–5	−30.24	5	−5.775	0
Instance 5–5	0	0	0	0
Instance 6–5	146.22	−2	61.485	−2
Instance 7–5	−27.57	5	−13.346	3
Instance 8–5	−1.34	2	−27.473	2
Instance 9–5	0	0	14.422	0
Instance 10–5	−34.27	4	−15.278	2

Table 5
Price per pallet in euro per order of x pallets.

Pallets	Price	Pallets	Price	Pallets	Price
1	29.95	13	13.75	24	9.74
2	25.49	14	13.25	25	9.47
3	22.88	15	12.8	26	9.22
5	21.03	16	12.37	27	8.98
6	19.59	17	11.97	28	8.75
7	18.41	18	11.6	29	8.52
8	17.42	19	11.25	30	8.31
9	16.56	20	10.92	31	8.1
10	15.8	21	10.6	32	7.9
11	14.87	22	10.3	33	7.51
12	14.28	23	10.01		

Partners *A* and *B* require a compensation of 4 euro, while partner *C* wishes 10 euro to compensate the change of a delivery date for one day.

Based on the pace list in Table 5, when no orders are allowed to be synchronized (situation (a)), each partner is responsible for a cost of 29.95 euro (total cost equals 89.85 euro). The total cost in situation (b) on the other hand is equal to 68.64 euro, while in situation (c), the cost has only decreased to 80.93 euro. Our heuristic provided for situation (c) an immediate cost allocation. This is shown in Table 1.

When minimizing the transportation cost and the cost of changing delivery dates and disregarding the cost allocation (synchronization with soft time windows), the total transportation cost for the example is lower than when integrating the Shapley constraints (synchronization with soft time windows and an integrated cost allocation). Although situation (b) has a lower total cost ($68.64 + 10 + 4 = 82.64 < 80.93 + 4 = 84.93$), allocating this cost proves to be problematic. When allocating the costs according to the Shapley value, which requires solving the problem for each subcoalition, the profit of partner *C* is not enough to compensate for the changed order delivery date. This is shown in Table 2, columns *Shapley allocation*. Moreover, the actual cost of partner *C* is the allocated transportation cost (24.366 euro), plus the cost of changing the delivery date of an order (10 euro). Therefore, in this example, the total cost for company *C* exceeds its stand-alone cost.

Another possibility would be to compensate the costs of changing delivery dates directly. E.g., when changing the delivery date of the order of company *C*, this company receives a reduction in its payment of 10 euro. When all costs are paid, the remainder of the profit can be divided among the partners. This can also done by the Shapley formula, as shown in Table 2, columns *Pay cost of changes*. Although this satisfies all partners, there is a clear problem with this solution. Although company *A* and *C* have, in each subcoalition, changed their delivery dates to the same extent, they have been allocated a different cost. Company *C* has succeeded in absorbing a large part of the gain to compensate its costs. However, in this case, there is an evident incentive for a company to report a higher (false) cost of changing delivery dates. For this reason, directly compensating a company for its reported costs is not recommended.

6.2. Additional tests

Reporting high thresholds for delivery date changes is undesirable, as it will render the coalition more inflexible and decrease the opportunities for synchronization. The integrated cost allocation has the advantage of discouraging partners to report higher thresholds. The example in Section 6.1 shows that if company *C* reports a cost that is too high, it will have missed the opportunity to receive the gains of consolidating its pallet with the pallets of company *A* and *B*. If the reported cost however is accurate, company *C* will suffer when optimizing the operational plan without integrating the cost allocation.

We can imagine two situations in which integrating the cost allocation method is of high importance. First of all, when a partner suffers a substantially higher cost when changing the delivery date of an order, it is possible that the solution of the sequential model will not be accepted by that partner, as is the case in the example in Section 6.1. The partner with a high cost prefers a solution with less changes and a higher allocated transportation cost. We simulate a situation, where a partner allows its delivery dates to be changed over a large time span, but the related cost is high. In this case, the delivery dates of an order can only be changed when a very high profit is realized by that change.

Secondly, it is important for the coalition that a company should not be rewarded by reporting a delivery date change cost that is too high. Therefore, neither of both models directly compensate these costs. Instead, both models allocate the total transportation cost using the Shapley value without taking into account the compensation costs. However, the sequential model still allows a partner to benefit from reporting a higher cost. As the penalty cost is minimized together with the transportation cost in this model it will always opt — given that the transportation cost remains the same — to change the dates of the order of the company with the lowest delivery date change cost. A company with a slightly higher cost than its partners can thus avoid that its dates are changed, to the detriment of (one of) its partners.

To compare the solutions of the integrated model with those of the sequential model, we have created ten instances. The instances, which are available from the authors upon request, each consists of ten orders. The order size of each order is randomly chosen between 1 and 33 pallets and the owner of the orders (partner A, B or C) is determined randomly as well. For each of these instances, we assume three different situations. In the first situation, each partner allows its delivery dates to be advanced or delayed for one day ($A_A = B_A = A_B = B_B = A_C = B_C = 1$) and the cost of changing is 10 euro ($C_A = C_B = C_C = 10$). In the second situation, partner A increases its limits to five days ($A_A = B_A = 5$). The cost of changing the delivery date however increases to 50 euro per day. In the third situation, partner A still allows limits of only one day, similarly to its partners, but the cost of changing the delivery date is slightly higher than its partners (11 euro in comparison to 10 euro).

The first situation depicts a situation in which all companies are similar. The results of this situation are useful to investigate whether differences can be detected between the solutions of both models in a completely symmetric situation. The second situation is chosen because it presents a challenge for both models to determine the optimal delivery date changes for partner A, as its larger hard time window will be balanced against its more expensive soft time windows in two different ways. The final scenario tests how the different evaluations of soft time windows of the models impact the solutions when the soft time windows of the different partners differ only slightly.

The costs C_A, C_B and C_C have been chosen in such a way that, using the price list, synchronization profit is still possible. A second set of instances is used to test whether the conclusions still hold when different time window widths and different costs of changing delivery dates are applied. Again, the first set of ten instances have similar costs and time window widths, i.e. $A_A = B_A = A_B = B_B = A_C = B_C = 3$ and $C_A = C_B = C_C = 20$. Secondly, instead of partner A stating a significantly larger cost in comparison to its partners, he decreases its cost significantly, with $C_A = 5$. Finally, partner A decreases its cost to 19 euros, stating a slightly lower cost of change than its partners.

The differences in total cost, total days delayed or advanced, and the allocated cost and number of days moved away from the delivery date, belonging to partner A are shown in Tables 3 and 4. A positive number implies an increase (in cost or number of days), a negative number a decrease.

In many cases, the integrated approach is able to find a solution with a lower transportation cost than the solution minimizing the total cost (transportation cost and cost of changing delivery dates). It can be derived from Table 3 that in these cases, it is possible to make larger or more delivery date changes to further lower the transportation cost.

However, when the delivery date change cost of a partner is substantially higher than that of the others, it is possible that the Shapley constraints (10) are so restrictive that the total transportation cost of the solution of the integrated model is higher. In instance 1–1 and 2–1, it is clear that in order for company A to agree to the proposed solution, a final solution with a lower number of delivery dates changed (in days) for the orders of company A, is needed. However, the total transportation cost in such a solution is significantly higher. In analogy to the example in Section 6.1, the allocated transportation cost to company A therefore increases significantly as well. Company A is thus willing to trade a higher allocated transportation cost for a decrease in changes in delivery dates, which is only properly taken into account by the integrated model.

The positive effect of incorporating the cost of changing delivery dates into the constraints rather than into the objective of a model, is apparent in the examples in which partner A states to have a slightly higher cost of changing the date of a delivery than its partners. In all instances the number of days an order of company A is changed further away from its original delivery date is higher than would be the case when optimizing according to the sequential model. When there is an (almost) equal profit to be realized when either partner A or another partner will change the delivery date of an order, the synchronization with soft time windows using the sequential model will always prefer the delivery date of the order of the other partner to be changed. Including the cost of changing delivery dates into the Shapley constraints on the other hand does not favor the partner with slightly higher delivery date change cost. As the costs of changing delivery dates are often an estimate, and it should be avoided to give incentives to exaggerate this cost, the integrated model has a clear advantage over the sequential model. When having a slightly lower threshold than its partners, Instance 1–5 shows that, for an equal number of days moved, it is less evident that partner A is moving its orders.

Finally, as shown in Table 7, the calculation times required to solve the integrated model are higher than those of to solve the sequential model. This is expected, as the exact solver is used more often. Increasing the number of participants in a coalition, and thus increasing the number of subcoalitions that have to be solved, will widen this gap exponentially. This should be resolved by e.g., solving submodels with meta-heuristics, re-using information of good solutions in subcoalitions to find a new solution etc. This is however outside the scope of this paper.

7. Conclusions and future work

Although a sizeable number of optimization models and methods exist that focus on operational planning in the context of a horizontal logistic coalition, most neglect the fact that a solution to such an optimization problem should be acceptable to all partners. In this paper, we have discussed a cooperative transportation problem, in which partners are willing to bundle their orders and agree that there orders are synchronized, i.e., delivered on different dates than their preferred delivery date. Our model, however, does not neglect the fact that changing the delivery date of an order might result in additional costs for the owner of that order. Although it is easy to quote a price for such a change (e.g., a penalty fee to be paid by the coalition to that partner), it is generally difficult to determine its exact cost. Moreover, a partner in the coalition should not be rewarded for inflating these costs.

Table 6
Full output additional tests.

				Results for the sequential model									Results for the integrated model					
Inst.	φ_A	φ_B	φ_C	Total cost	$\sum e_a$	$\sum e_b$	$\sum e_l$	$\sum e_{all}$	Time (ms)	φ_A	φ_B	φ_C	Total cost	$\sum e_a$	$\sum e_b$	$\sum e_l$	$\sum e_{all}$	Time (ms)
$A_A = A_B = A_C = 1, C_A = C_B = C_C = 10$																		
1	1480.25	358.72	73.62	1912.59	2	1	1	4	5260	1480.25	358.72	73.62	1912.59	2	1	1	4	27,090
2	338.40	350.42	543.11	1231.93	3	3	1	7	4340	340.59	347.96	543.39	1231.93	2	4	1	7	139,280
3	880.05	227.28	242.88	1350.21	0	0	0	0	113,140	877.33	227.28	240.16	1344.76	0	0	1	1	204,060
4	236.25	293.40	851.28	1380.93	0	0	2	2	7950	235.08	293.40	850.11	1378.58	0	0	3	3	461,160
5	52.06	570.28	758.76	1381.10	1	0	3	4	18,310	52.06	570.28	758.76	1381.10	1	0	3	4	105,400
6	589.29	590.43	11.02	1190.73	1	0	1	2	13,130	589.29	590.43	11.02	1190.73	0	1	1	2	361,960
7	684.78	229.51	229.52	1143.81	1	0	0	1	26,180	671.57	216.30	230.12	1117.99	2	1	2	5	671,640
8	348.18	154.29	281.32	783.79	1	0	0	1	33,290	340.68	146.78	274.49	761.94	3	0	1	4	1,552,980
9	797.73	155.73	609.73	1563.18	0	2	0	2	10,440	797.73	155.73	609.73	1563.18	0	2	0	2	237,780
10	472.90	228.43	430.72	1132.05	2	0	2	4	8860	472.90	228.43	430.72	1132.05	2	0	2	4	149,530
$A_A = 5, A_B = A_C = 1, C_A = 50, C_B = C_C = 10$																		
1–1	1480.25	358.72	73.62	1912.59	2	1	1	4	5390	1525.28	403.74	118.65	2047.67	0	1	1	2	52,670
2–1	338.19	350.21	543.53	1231.93	2	4	1	7	4380	408.80	416.17	543.39	1368.35	0	3	1	4	238,560
3–1	880.05	227.28	242.88	1350.21	0	0	0	0	115,860	877.33	227.28	240.16	1344.76	0	0	1	1	232,760
4–1	236.25	293.40	851.28	1380.93	0	0	2	2	8460	235.08	293.40	850.11	1378.58	0	0	3	3	371,290
5–1	62.66	570.28	769.36	1402.30	0	0	3	3	18,340	59.13	577.35	765.83	1402.30	0	0	3	3	183,320
6–1	589.29	590.43	11.02	1190.73	0	1	1	2	13,090	589.29	590.43	11.02	1190.73	0	1	1	2	165,840
7–1	754.47	167.55	233.39	1155.41	0	0	2	2	27,410	747.63	170.97	236.81	1155.41	0	0	2	2	1,320,310
8–1	378.95	103.70	301.14	783.79	1	0	0	1	38,270	378.28	103.03	301.14	782.45	1	1	0	2	2,080,520
9–1	799.67	157.68	605.83	1563.18	0	2	0	2	12,200	799.67	157.68	605.83	1563.18	0	2	0	2	286,750
10–1	480.55	213.13	438.37	1132.05	1	0	3	4	9560	456.53	213.13	462.40	1132.05	2	0	2	4	1,246,060
$A_A = A_B = A_C = 1, C_A = 11, C_B = C_C = 10$																		
1–2	1480.25	358.72	73.62	1912.59	2	1	1	4	6710	1480.25	358.72	73.62	1912.59	2	1	1	4	26,610
2–2	338.40	350.42	543.11	1231.93	2	3	2	7	4460	340.59	347.96	543.39	1231.93	2	4	1	7	143,090
3–2	880.05	227.28	242.88	1350.21	0	0	0	0	127,050	877.33	227.28	240.16	1344.76	0	0	1	1	221,450
4–2	236.25	293.40	851.28	1380.93	0	0	2	2	9530	235.08	293.40	850.11	1378.58	0	0	3	3	498,980
5–2	52.06	570.28	758.76	1381.10	1	0	3	4	19,240	52.06	570.28	758.76	1381.10	1	0	3	4	103,770
6–2	589.29	590.43	11.02	1190.73	0	1	1	2	14,280	589.29	590.43	11.02	1190.73	0	1	1	2	383,970
7–2	684.78	229.51	229.52	1143.81	1	0	0	1	30,930	671.57	216.30	230.12	1117.99	2	1	2	5	714,110
8–2	348.18	154.29	281.32	783.79	1	0	0	1	39,510	340.68	146.78	274.49	761.94	3	0	1	4	528,610
9–2	797.73	155.73	609.73	1563.18	0	2	0	2	11,980	797.73	155.73	609.73	1563.18	0	2	0	2	228,130
10–2	472.90	228.43	430.72	1132.05	1	0	3	4	10,430	472.90	228.43	430.72	1132.05	2	0	2	4	152,760

Time, expressed in milliseconds, is the total time to solve the instance and the calculate the Shapley value. The instances are run on an Intel(R) Core(TM)2 Duo CPU P9700 @ 2.80GHZ.

In this paper, we have argued that instead of minimizing the total cost (i.e., the sum of the total transportation cost and the cost of changing delivery dates) a better approach is to minimize only the transportation cost, but ensure delivery dates are only changed if this will generate sufficient profit *for the partner undergoing the change*. This requires that the division of the total coalition cost between the partners is known during the optimization phase, and therefore that the cost allocation method (in this paper we have used the Shapley value) is *integrated* into the optimization method. In the heuristic developed in this paper, the so-called *Shapley constraints* ensure that the profit allocated to each partner is at least as large as this partner's reported cost of changed delivery dates.

The contributions of this paper to the literature are a new model and a heuristic algorithm that is capable of finding solutions (an operational plan and a cost allocation) that will satisfy all partners. The computational results show that this heuristic can often find a solution with a lower total transportation cost. Moreover, when a partner has a very high cost of changing delivery dates, the heuristic will alter the operational plan in such a way that this partners' order changes are small, at the same time allocating a smaller profit to this partner. Finally, when a partner states to have a slightly higher cost of changing its delivery dates than another partner, this model will avoid a systematic advantage in favor of this partner.

The main focus of this paper has been to demonstrate how a cost allocation method can be integrated in the operational planning problem of a horizontal logistics coalition and how this way of working potentially solves some of the pitfalls of collaborative distribution planning. The developed algorithm, on the other hand, should be considered a proof-of-concept implementation rather than one usable in practice. The addition of the Shapley constraints adds considerable complexity, and solving large instances of the optimization problem is currently not tractable. Future research should therefore focus on the development of faster algorithms, e.g., by integrating less computationally intensive cost allocation methods, by solving the subproblems heuristically rather than exactly, or by performing general computational optimization through more efficient neighborhood structures.

The additional complexity of the proposed integrated collaborative planning model also raises questions with respect to its practicality and desirability. It is clear that the additional cost of solving the integrated model, as well as the effort

Table 7
Full output additional tests 2.

	Results for the sequential model									Results for the integrated model								
	φ_A	φ_B	φ_C	Total cost	$\sum e_a$	$\sum e_b$	$\sum e_l$	$\sum e_{all}$	Time* (ms)	φ_A	φ_B	φ_C	Total cost	$\sum e_a$	$\sum e_b$	$\sum e_l$	$\sum e_{all}$	Time (ms)
$A_A = B_A = A_B = B_B = A_C = B_C = 3, C_A = C_B = C_C = 20$																		
1–3	1480.25	359	73.62	1912.59	1	1	2	4	0	1480.25	359	73.62	1913	1	1	2	4	53,230
2–3	338.188	350	543.533	1231.93	0	5	1	6	0	340.585	348	543.39	1232	1	5	0	6	883,390
3–3	880.05	227	242.88	1350.21	0	0	0	0	0	876.885	227	239.715	1344	0	0	3	3	280,030
4–3	236.082	243	915.812	1395.02	0	0	1	1	0	234.907	229	900.692	1365	0	2	4	6	619,400
5–3	52.0583	632	696.878	1381.1	2	0	2	4	0	52.0583	632	696.878	1381	2	0	2	4	237,770
6–3	598.697	586	6.31167	1190.73	2	0	0	2	0	611.272	563	15.7017	1190	1	5	0	6	856,790
7–3	686.395	229	228.295	1143.81	1	0	0	1	0	664.078	219	233.388	1116	5	1	0	6	500,640
8–3	399.245	108	192.16	698.95	1	0	3	4	0	381.1	102	214.89	698	1	1	3	5	1,039,000
9–3	776.658	155	631.923	1563.18	0	2	0	2	0	779.797	158	625.647	1563	0	2	0	2	492,050
10–3	448.875	228	454.745	1132.05	2	0	2	4	0	433.597	225	439.467	1098	4	0	4	8	1,240,460
$A_A = B_A = 5, A_B = B_B = A_C = B_B = 3, C_A = 5, C_B = C_C = 20$																		
1–4	1480.25	359	73.62	1912.59	4	0	0	4	0	1480.25	359	73.62	1913	1	1	2	4	54,200
2–4	338.188	350	543.533	1231.93	5	1	2	8	0	340.373	348	543.813	1232	1	5	0	6	153,120
3–4	880.05	227	242.88	1350.21	0	1	0	1	0	876.885	227	239.715	1344	0	0	3	3	317,470
4–4	236.082	243	915.812	1395.02	3	0	1	4	0	234.907	229	900.692	1365	0	2	4	6	619,600
5–4	52.0583	632	696.878	1381.1	2	0	2	4	0	52.0583	632	696.878	1381	2	0	2	4	208,360
6–4	598.697	586	6.31167	1190.73	2	0	0	2	0	605.652	574	10.0817	1190	1	5	0	6	878,300
7–4	667.322	217	232.162	1116.31	4	0	0	4	0	664.078	219	233.388	1116	5	1	0	6	462,680
8–4	391.215	100	184.13	674.86	5	0	3	8	0	360.842	119	194.632	675	5	0	3	8	802,410
9–4	781.603	160	627.453	1568.6	2	1	0	3	0	776.007	154	632.347	1562	3	3	1	7	345,680
10–4	443.24	228	449.11	1120.78	4	0	2	6	0	431.013	229	436.883	1096	6	0	5	11	643,400
$A_A = A_B = A_C = 3, C_A = 19, C_B = C_C = 20$																		
1–5	1480.25	359	73.62	1912.59	4	0	0	4	35,940	1480.25	359	73.62	1913	2	2	0	4	145,850
2–5	338.188	350	543.533	1231.93	1	5	0	6	432,450	340.585	348	543.39	1232	1	5	0	6	649,840
3–5	880.05	227	242.88	1350.21	0	0	0	0	90,080	876.885	227	239.715	1344	0	0	3	3	511,960
4–5	236.082	243	915.812	1395.02	0	0	1	1	27,410	230.307	231	902.992	1365	0	2	4	6	413,920
5–5	520.583	632	696.878	1381.1	2	0	2	4	89,510	520.583	632	696.878	1850	2	0	2	4	297,650
6–5	598.697	586	631.167	1190.73	2	0	0	2	38,330	660.182	612	646.117	1918	0	0	0	0	866,370
7–5	679.878	223	241.328	1143.81	1	0	0	1	77,890	666.532	218	232.162	1116	4	1	1	6	763,110
8–5	399.245	108	192.16	698.95	1	0	3	4	152,030	371.772	120	205.562	698	3	0	3	6	2,160,890
9–5	776.658	155	631.923	1563.18	0	2	0	2	71,700	791.08	169	603.08	1563	0	2	0	2	258,980
10–5	448.875	228	454.745	1132.05	2	0	2	4	334,960	433.597	225	439.467	1098	4	0	4	8	916,180

Time, expressed in milliseconds, is the total time to solve the instance and the calculate the Shapley value. The instances are run on an Intel(R) Core(TM)2 Duo CPU P9700 @ 2.80 GHZ.

required to gather all required data, may in some cases be substantial. We leave it to future research to demonstrate the limits of our model's usability in real-life horizontal logistics coalitions.

The concepts developed in this paper can readily be extended towards other collaborative optimization problems. An interesting avenue for future research is therefore the development of novel models and methods that will support truly collaborative optimization in a large variety of realistic logistic situations.

Acknowledgments

This research has been funded through a Ph.D. grant of the Flemish Agency for Innovation by Science and Technology (IWT).

Appendix A. Pace list

Table 5.

Appendix B. Full output additional tests 1

Tables 6 and 7.

References

Audy, J.-F., D'Amours, S., Rönnqvist, M., 2007. Business models for collaborative planning in transportation: an application to wood products. In: Establishing The Foundation Of Collaborative Networks. Springer, pp. 667–676.

Audy, J.-F., D'Amours, S., Rousseau, L.-M., 2011. Cost allocation in the establishment of a collaborative transportation agreement – an application in the furniture industry. J. Oper. Res. Soc. (62), 960–970.
Barratt, M., 2004. Understanding the meaning of collaboration in the supply chain. Supply Chain Manage.: An Int. J. 9 (1), 30–42.
Berger, S., Bierwirth, C., 2010. Solutions to the request reassignment problem in collaborative carrier networks. Transp. Res. Part E: Logist. Transp. Rev. 46 (5), 627–638.
Biermasz, J., 2012. Report on the Legal Framework for Horizontal Collaboration in the Supply Chain and Model Legal Agreements. Technical Report, CO^3 Position Paper.
Boyacı, T., Gallego, G., 2002. Coordinating pricing and inventory replenishment policies for one wholesaler and one or more geographically dispersed retailers. Int. J. Prod. Econ. 77 (2), 95–111.
Van Breedam, A., Vannieuwenhuyse, B., Verstrepen, S., 2011. The Making of the World's First Orchestrated Horizontal Collaboration. Presentation at the 2nd Horizontal Collaboration Summit, May 2011.
Cachon, G.P., 2003. Supply chain coordination with contracts. Handbooks Oper. Res. Manage. Sci. 11, 229–340.
Chen, J.-M., Chen, T.-H., 2005. The multi-item replenishment problem in a two-echelon supply chain, the effect of centralization versus decentralization. Comput. Oper. Res. 32 (12), 3191–3207.
Chen, C.L., Lee, W.C., 2004. Multi-objective optimization of multi-echelon supply chain networks with uncertain product demands and prices. Comput. Chem. Eng. 28 (6), 1131–1144.
Clark, A.J., Scarf, H., 1960. Optimal policies for a multi-echelon inventory problem. Manage. Sci. 6 (4), 475–490.
Conen, W., 2002. Economically coordinated job shop scheduling and decision point bidding-an example for economic coordination in manufacturing and logistics. In: Proceedings of the 16th Workshop on Planen, Scheduling und Konfigurieren, Entwerfen.
Cooke, A., 2011. Sharing supply chains for mutual gain. Supply Chain Quart. (2).
Cruijssen, F., Cools, M., Dullaert, W., 2007. Horizontal cooperation in logistics: opportunities and impediments. Transp. Res. Part E: Logist. Transp. Rev. 46 (3), 22–39.
Davidsson, P., Henesey, L., Ramstedt, L., Törnquist, J., Wernstedt, F., 2005. An analysis of agent-based approaches to transport logistics. Transp. Res. Part C: Emerging Technol. 13 (4), 255–271.
Dudek, G., Stadtler, H., 2005. Negotiation-based collaborative planning between supply chains partners. Eur. J. Oper. Res. 163 (3), 668–687.
Frisk, M., Göthe-Lundgren, M., Jörnsten, K., Rönnqvist, M., 2010. Cost allocation in collaborative forest transportation. Eur. J. Oper. Res. 205 (2), 448–458.
Guinouet, A., Jordans, M., Cruijssen, F., 2012. JSP and HF-Czechforge to Bundle Plastics and Steel Shipments from Czech Republic to Germany. Technical Report, CO^3 Collaboration Concepts for Co-modality, 2012.
Ham, A.V.D., Huijsman, M., Rustenberg, M., Verweij, K., 2006. Handboek generieke procesaanpak verladerssamenwerking. TNO Mobiliteit & Logistiek, Delft.
Hansen, P., Mladenovi, N., 2001. Variable neighborhood search: principles and applications. Eur. J. Oper. Res. 130 (3), 449–467.
Houghtalen, L., Ergun, O., Sokol, J., 2011. Designing mechanisms for the management of carrier alliances. Transp. Sci. 45 (4), 465–482.
Ireland, R., Bruce, R., 2000. Cpfr-only the beginning of collaboration. Supply Chain Manage. Rev. 4 (September/October), 80–88.
Krajewska, M.A., Kopfer, H., 2006. Collaborating freight forwarding enterprises. OR Spectrum 28 (3), 301–317.
Kwon, R.H., Lee, C.-G., Ma, Z., 2005. An Integrated Combinatorial Auction Mechanism for Truckload Transportation Procurement. Working Paper, University of Toronto. URL: http://www.mie.utoronto.ca/labs/ilr/MultiRound.pdf.
Lee, H.L., Padmanabhan, V., Whang, S., 1997. Information distortion in a supply chain, the bullwhip effect. Manage. Sci. 43 (4), 546–558.
Lehoux, N., D'Amours, S., Langevin, A., 2010. A win-win collaboration approach for a two-echelon supply chain: a case study in the pulp and paper industry. Eur. J. Ind. Eng. 4 (4), 493–514.
Liu, P., Wu, Yaohua, Xu, N., 2010. Allocating collaborative profit in less-than-truckload carrier alliance. J. Serv. Sci. Manage. 3, 143–149.
Loehman, E., Whinston, A., 1974. An axiomatic approach to cost allocation for public investment. Public Finan. Quart., 236–251.
Özener, O.-O., Ergun, O., Savelsbergh, M., 2011. Lane-exchange mechanisms for truckload carrier collaboration. Transp. Sci. 45 (1), 1–17.
Pal, B., Sana, S.S., Chaudhuri, K., 2012. A three layer multi-item production–inventory model for multiple suppliers and retailers. Econ. Model. 29 (6), 2704–2710.
Shapley, L.S., 1953. A value for n-person games. Ann. Math. Stud. 28, 307–317.
Slaughter, May. 2012. The EU Competition Rules on Horizontal Agreements. Technical Report, May 2012.
Solomon, M.M., Baker, E.K., Schaffer, J.R., 1988. Vehicle routing and scheduling problems with time window constraints: efficient implementations of solution improvement procedures. In: Golden, B., Assad, A. (Eds.), Vehicle Routing: Methods and Studies. Elsevier Science Publishers, pp. 85–106.
Soons, D., 2011. The Determination and Division of Benefits Among Partners of a Horizontal Cooperation in Transportation. Master's Thesis, TU/e School of Industrial Engineering, Eindhoven, May 2011.
Taillard, E., Badeau, P., Gendreau, Mi., Guertin, F., Potvin, J.-Y., 1997. A tabu search heuristic for the vehicle routing problem with soft time windows. Transp. Sci. 31 (2), 170–186.
Thomas, D.J., Griffin, P.M., 1996. Coordinated supply chain management. Eur. J. Oper. Res. 94 (1), 1–15.
Van Breedam, A., Vannieuwenhuyse, B., Verstrepen, S., 2013. TriVizor. Private Commun..
Vanovermeire, C., Sörensen, K., Van Breedam, A., Vannieuwenhuyse, B., Verstrepen, S., 2013a. Horizontal logistic collaboration: decreasing costs through flexibility and an adequate cost allocation strategy. Int. J. Logist. Res. Appl..
Vanovermeire, C., Vercruysse, D., Sörensen, K., 2013b. Analysis of different cost allocation methods in a collaborative transport setting. Int. J. Eng. Manage. Econ..
Verweij, K., 2009. Horizontal Supply Chain Collaboration: Experiences from Benelux Case Studies. November 2009. URL: <http://www.goederenstromen.nl/pers-downloads.html>.
Zou, X., Pokharel, S., Piplani, R., 2004. Channel coordination in an assembly system facing uncertain demand with synchronized processing time and delivery quantity. Int. J. Prod. Res. 42 (22), 4673–4689.