DEPARTMENT OF ENGINEERING MANAGEMENT

Integration of the cost allocation in the optimization of collaborative bundling

Christine Vanovermeire & Kenneth Sörensen

UNIVERSITY OF ANTWERP
Faculty of Applied Economics
Stadscampus
Prinsstraat 13, B.226
BE-2000 Antwerpen
Tel. +32 (0)3 265 40 32
Fax +32 (0)3 265 47 99
www.ua.ac.be/tew

FACULTY OF APPLIED ECONOMICS

DEPARTMENT OF ENGINEERING MANAGEMENT

Integration of the cost allocation in the optimization of collaborative bundling

Christine Vanovermeire & Kenneth Sörensen

RESEARCH PAPER 2013-019
SEPTEMBER 2013

University of Antwerp, City Campus, Prinsstraat 13, B-2000 Antwerp, Belgium
Research Administration – room B.226
phone: (32) 3 265 40 32
fax: (32) 3 265 47 99
e-mail: joeri.nys@uantwerpen.be

The papers can be also found at our website:
www.ua.ac.be/tew (research > working papers) &
www.repec.org/ (Research papers in economics - REPEC)

D/2013/1169/019

Integration of the cost allocation in the optimization of collaborative bundling

Christine Vanovermeire[1] and Kenneth Sörensen[1]

[1]University of Antwerp, Prinsstraat 13, 2000 Antwerpen,
firstname.lastname@uantwerpen.be

September 4, 2013

Abstract

By bundling orders of different partners within the time windows suggested to their 3PLs, a large reduction in cost can be achieved. However, these benefits can be further enlarged by loosening the time window constraints. Using the real costs structure of delivering an order too early or too soon, a partner can give more delivery date options, given that he receives an adequate compensation for changing the delivery date of this order. To be able to create an operational plan and a compensatory cost allocation, we create a model that integrates the Shapley value when bundling and solve it with a heuristic. We show that such an integrated approach –in comparison to first plan, then divide– increases the satisfaction of the partners.

1. Introduction

Optimizing a single functional entity in the supply chain can have often negative effects on the companies downstream or upstream the supply chain (Ireland and Bruce 2000). The bullwhip-effect for example (Lee et al. 1997), describes how self-optimization and non-communication can lead to an increased demand amplification on the chain. In order to create more efficient supply chains, partners have to collaborate. As can be seen in Figure 1, these collaborations can be internally, between different functions (e.g., sales representatives have to communicate with the production site), externally, between suppliers or customers (*vertical* collaboration, e.g. Vendor Managed Inventory), or even between competitors or other organizations on the same level of the supply chain (*horizontal* collaboration).

When horizontally collaborating with companies that have similar or complementary logistic needs, the term “bundling” is used to imply that rather than transporting orders independently, the transport of multiple companies is organized simultaneously and the means of transportation can thus contain orders of several companies.

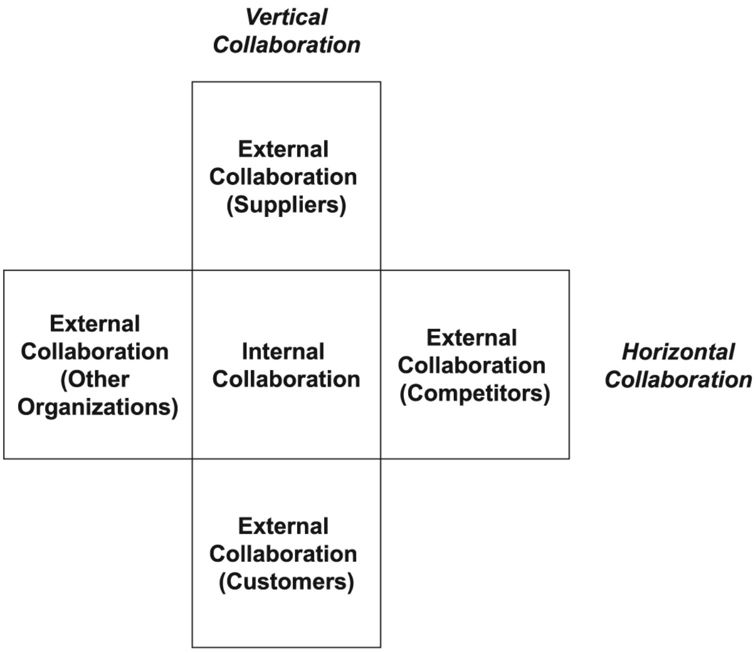

Vertical Collaboration
External Collaboration (Suppliers)
External Collaboration (Other Organizations)
Internal Collaboration
External Collaboration (Competitors)
Horizontal Collaboration
External Collaboration (Customers)

Figure 1: The scope of collaboration(Barratt 2004)

A good business case for horizontal collaboration occurs when companies are located close to each other and have a significant number of mutual clients. An example is the collaboration between two fast-moving consumer goods producers, Kimberly-Clark and Unilever-HPC. Kimberly-Clark and Unilever-HPC have 60-70% of their delivery addresses in common in the Netherlands. They initiated a project, shipping their products from a shared warehouse to their mutual retail clients. The bundling of their orders resulted in an increased service level (three delivery days instead of two per week), 50% less trips and a decrease in handling costs of 20% (Verweij 2009).

Another case of profitable collaborative bundling is the creation of so-called “lanes”, when companies in the same region ship long-haul to another region. Two companies located in Germany for example –JSP in Eschenau and Hammerwerk in Fridingen– send out their trucks to the Czech Republic collaboratively. 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 and Jordans).

In Vanovermeire et al. (2012), we argue that collaborative bundling differs from the traditional bundling of a third party logistic provider, as the collaboration is more intensified, and the bundling can be actively *synchronized*. This implies, for example, that partners can decide that an order can be transported earlier or later when the benefits exceed the costs of changing the delivery date of that order. However, as collaboration does not imply that partners are willing to give up their own interests for the sake of the coalition, the *total* benefits do not only need to exceed the costs of changing the delivery date of an order, also the *allocated* benefit for the partner who’s delivery dates are changed has to be larger than that partner’s cost related to these changes.

When planning collaboratively however, partners are faced with one collective cost. One of the most important difficulties when collaborating is the division of that cost (Cruijssen et al. 2007). A lot of research has been done on this topic, and it often seems that each case study results in a new cost allocation method (Frisk et al. 2010, Liu et al. 2010, Soons 2011, Audy et al. 2011). One of the most crucial properties of a good cost allocation method is that it has to ensure that each partner will benefit from entering the coalition (Vanovermeire et al. 2012). In other words, the profit of a partner, i.e. the stand-alone cost minus the allocated cost, should be positive.

Applied to the example used in this paper, any changes to the delivery dates that are allowed by a partner, should add additional profits to that partner. In other words, when the decision is taken to change a delivery date, the cost allocated to the owner of the order has to be known already.

To be able to know the allocation, the cost allocation *method* should be chosen beforehand as well. As the Shapley value (Shapley 1953) is recommended by several authors to be used when collaborating horizontally (Krajewska et al. 2008, Verstrepen and 't Hooft 2011) and especially in the case of synchronization (Vanovermeire et al. 2013), we opt to use this method in this paper (see Section 3).

This paper suggests that, when automating and optimizing the operational planning process, the cost allocation mechanism has to be integrated into the optimization method, as the decision to change an order's delivery date is dependent of the individually *allocated* benefit. By doing so, the model proposed in this paper is truly *collaborative*, in the sense that it takes into account the fact that partners are still individual entities.

2. Cooperative supply chain planning

According to Thomas and Griffin (1996), while the term Supply Chain Management is relatively new, the idea of coordinated planning is not. In their review, they have listed many multi-echelon models, starting with the first model of Clark and Scarf (1960), who optimize the total inventory along multiple installations. However, according to them, 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.

Lehoux et al. (2010) indeed warn, when studying different collaboration approaches, that none of them are necessarily equally profitable to all partners. They investigate four collaboration approaches and analyse them on the basis of the profit of each actor. A particular collaboration mode is however not necessarily advantageous for both partners. They therefore emphasize the need for incentives (also called side payments) or a method for sharing 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 agent-based approaches, as many of the problems consists of (1) well-defined entities with (2) complex behaviours 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.

When comparing decentralization versus 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 or 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.

Contrarily to the suggestions of Dudek and Stadtler (2005) and Davidsson et al. (2005), our approach uses a centralized approach, finding a minimal *total* transportation cost. However, we do allow the individual preferences to influence the operational plan.

3. The Shapley value

To calculate the Shapley value (Shapley 1953), the following formula is used. Assume a grand coalition N, which exist of partners $\{1, ..., i, ...n\}$ and a total transportation cost $c(N)$. Each subset of partners S has a transportation cost $c(S)$. The Shapley value for partner i, when dividing the cost of the grand coalition N, is the weighted sum of all marginal costs of partner i in each subset S:

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

The Shapley value is a game-theoretical concept, that is solely based on the marginal costs of a partner. According to Loehman and Whinston (1974), the Shapley value is thus completely "utilitarian", i.e. it only considers a player's cooperative productivity (the marginal profit) and only that productivity when determining its share of the gain. In Vanovermeire et al. (2012) and Vanovermeire et al. (2013), we show that, basing the cost allocation on the marginal costs, indeed gives the best incentives to increase that cooperative productivity. They both discuss how incentives should be given to loosen the time window constraints allowing to increase the profits of the collaboration. In this paper, we use the Shapley value to give similar incentives to minimize the transportation cost by changing delivery dates.

4. Model

4.1. Synchronization with soft time windows

In this paper, a model is developed, in which collaboration occurs in *lanes* (as seen in Van Breedam et al. (2011) or in Vanovermeire et al. (2013)). Starting from a common shared warehouse, region,..., goods are sent out to a mutual destination. The goods are handled by a logistic service provider, offering a price per pallet that decreases per additional pallet in a trip. A trip can contain 1 to C (maximum capacity) pallets.

The problem at hand is to bundle all orders in such a way that the total transportation cost is minimal. The companies actively search for more bundling opportunities by changing delivery dates to create a better fit with their partners. This is defined as *synchronization* in our paper. To be able to synchronize, each order needs a *time window*, i.e., an upper and a lower limit on that order's delivery date.

In the vehicle routing problem, the concept of time windows is also used. In traditional routing problems, hard time window constraints are defined, as can be seen in Solomon et al. (1988). Time windows in this case are continuous intervals (generally expressed in minutes or hours on a given day) in which a delivery can occur. Although our problem is periodical (tours occur on several days) and the time windows in our case are discrete intervals (stating the different days in which an order can be delivered), the concept of time windows is similar. When an order cannot be delivered within the time window, the solution is infeasible. However, when an order is delivered within the time window, no additional costs occur.

A more realistic approach however is that companies have a preferred date or point in time, but advancing or delaying orders to a small degree is possible. The further away an order is delivered from its original delivery date however, the less appealing this option is to a partner. Delivering orders on the preferred date generally adds no cost, while delivering earlier or later can increase stock levels, cause customer dissatisfaction, increases the risk of late deliveries and so on.

Therefore, *soft* time windows are introduced. Rather than defining a small time span in which an order can be delivered, a cost function is defined. In Taillard et al. (1997) for example, lateness at customer locations is allowed although a penalty is incurred and added to the objective value.

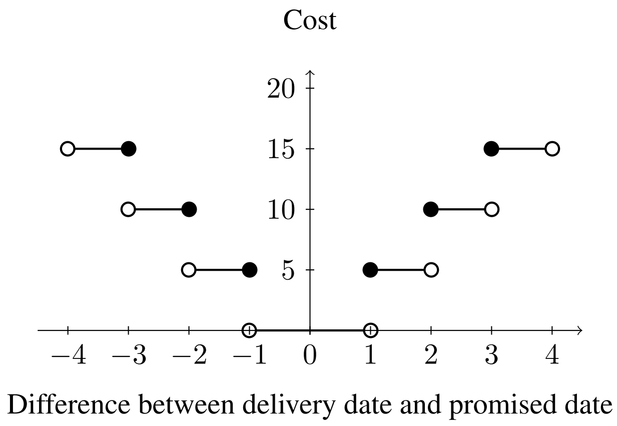


Figure 2: Example of soft window constraints

In our case, a cumulative cost or a penalty is suffered for each day an order is delivered further from its original delivery date. In this section, a model is introduced for such soft time windows. Similar to Taillard et al. (1997), the additional costs of changing the delivery date of orders is added to the objective value. However, although we assume that each partner suffers a different cost when changing the delivery date of an order, as there is no guarantee that each partner will be satisfied with the solution of this model, this model is not inherently a *collaborative* model. In Section 4.2, this problem is resolved. The collaborative aspect is such that the cost related to changing the delivery date of an order of a partner should be exceeded by the profit *allocated to that partner*.

In Figure 2, such soft time windows are visualized. Rather than only delivering in the small time span that does not add additional costs to the partner, a lot more possibilities exist. The further away the order is delivered from its preferred delivery date, the higher the additional costs are. Although we assume a fixed cost per day throughout this paper, cost functions can also e.g. state that delivering orders too early is less costly than delivering too late.

Given is a list of orders. For each order i, we know the preferred delivery date f_i, the order size q_i, expressed in number of pallets and the cost of changing the delivery date of an order for one day C_i. Moreover, we define an upper and lower limit indicating how many days an order can be advanced or postponed, defined as respectively the maximal advancement A_i and the maximal delay B_i. For a given list of trips and a price list $c(p)$ dependent on the number of pallets p in the trip j, we have to decide on the following decision variables:

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

$$z_{jp} = \begin{cases} 0 & \text{trip } j \text{ is not priced } c(p) \\ 1 & \text{trip } j \text{ is priced } c(p) \end{cases}$$

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

$e_i \in \mathbb{N}$ changes in delivery date of order i, expressed in number of days

Our optimization problem minimizes the total cost, which consists of the cost of changing the delivery dates, and the cost of transporting the orders (Equation 1). All the orders have to be delivered, and cannot be bundled more than once in a trip (Equations 2). All the trips have to be priced as well and cannot be priced twice (Equations 3). When no pallets are transported, the cost will be zero or $c(0) = 0$. The number of pallets in a trip cannot exceed the capacity of that trip (Equations 4). Equations (5) ensure that a trip receives a price that is not too low according to the number of pallets it transports. The delivery date cannot exceed the time span that is defined by the maximal advancement/delay $[f_i - A_i, f_i + B_i]$ (Equations 6 - 7). Equations (8-9) determine the number of days the delivery date of an order has been changed further away from its original delivery date.

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

s.t.

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

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

$$\sum_i q_i x_{ij} \leq Cap \qquad \forall j \tag{4}$$

$$Cap - z_{jp} Cap \geq \sum_i x_{ij} q_i - p \qquad \forall j, \forall p \tag{5}$$

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

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

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

$$e_i \geq d_j - f_i - (1 - x_{ij}) M \tag{9}$$

4.2. Synchronization with soft time windows and an integrated cost allocation

Adding a penalty cost to the objective function however does not guarantee an operational plan and a related cost allocation that will satisfy all partners. We therefore state that the previous model is not truly a *collaborative* model. In the following model, changing the delivery date of an order not only decreases the total cost, it also decreases the cost *of the owner of that order* to such an extent that it covers the reported cost of changing the delivery date of that order.

Furthermore, one of the drawbacks of the previous model is that the total transportation cost and the total cost of changing the delivery dates is summed. However, in contrast to the transportation cost, it is often impossible to state the *real* or exact costs of changing the delivery date of an order. In many cases, the cost structure of changing delivery dates is merely an estimate. Therefore, a separate evaluation of the transportation cost and the cost of changing delivery dates, rather than using the sum of the transportation cost and the cost of changing delivery dates, is a more realistic approach. Our model therefore minimizes solely the transportation cost. To take the cost of changing delivery dates into account, our model allows companies to determine a minimum threshold of profit allocated they need for a delivery date change to be accepted.

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 model thus not only encompasses an operational plan that minimizes the transportation cost, but also the division of costs. A feasible solution delivers all orders within the limits of the time windows and ensures that each partner earns enough to compensate the costs linked to the soft time window constraints.

As the cost allocation is integrated into the model, the decision variables as defined in Section 4.1 do not suffice. Additional decision variables regarding the allocated cost need to be included. Moreover, to determine the *profit* a partner can make by allowing that its delivery dates can be changed, the allocated cost when he would not have been flexible, should be known as well. We therefore define that a partner can have two "states". First, a *rigid* state, in which an operational plan is created in which no delivery dates are allowed to be changed, or, A_i and B_i (being the upper and lower limit of the time windows of all orders i) are set to zero. Secondly, a partner can assume a *flexible* state, the delivery dates of the orders of that partner can be changed within the limits of the time windows, given that there are also soft time windows defined.

A partner however is not alone in a coalition. To denote the states of all the partners in a (sub)coalition S, we define the combination of states $U = \{u_1, u_2, ..., u_S\}$. For each partner k, $u_k =$ rigid or $u_k =$ flex. A combination of states $V\{v_1, v_2, ..., v_S\}$ is a subset of U when for each partner k in V, its state is either rigid or equals the state of partner k in U.

The model in Section 4.1 is adapted in the following manner:

First of all, additional decision variables, integrating the cost allocation in the model, are included.

$\varphi^U_{k\in S} \in \mathbb{Z}$ allocated cost to partner k given the total cost of S when the partners assume the states U $\quad \forall U, \forall S, \forall k$

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 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* of changing delivery dates for a partner, the allocated cost when he would not allow dates to be changed has to be known.

This allocated cost, is called the *baseline*. For a given baseline, this baseline should be higher than (or equal to) the allocated cost to that partner when he allows its delivery dates to change.

Determining the baseline is ambiguous, as there are several combinations of states 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 combinations of states of the remaining partners in that subcoalition S.

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

The additional constraints, which we call *Shapley-constraints*, will be:

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

By choosing the *highest* possible allocated cost to partner k when k is rigid, the difference in cost allocation by allowing delivery dates to be changed will always be positive. This will avoid the following situations:

- Using as a baseline a combination of states in which all partners have a rigid state, in the final solution (when all partners assume the states ”flex”), the allocated cost to partner k can be higher than the baseline allocation. It is possible that the cost effect of the other partners becoming flexible is extremely large and that they therefore profit to such an extent (i.e., receive a share in the cost so small) that the remaining cost allocated to partner k in this situation is larger than in the 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.

- When the baselines for partner k and for another partner l are the allocated costs in the situations in which all partners are flexible with exception of respectively partner k and partner l, another problem arises. It is possible that the allocated profits when flexible to partner k and to another partner l do not exceed their cost of changing delivery dates. In this case, partner k as well as partner l will prefer not to change the dates of any of their orders. 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 (the chosen baseline). 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 the 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 problem at hand. To be able to calculate the Shapley value, the costs of all the subcoalitions of S has to be calculated as well. Moreover, to determine the maximal cost allocation to a rigid partner k, each situation in which partner k is rigid has to be evaluated. To determine all these costs, for each subcoalition and each combination of states U, an operational plan has to be constructed. The problem is clearly nested. We have therefore created a matheuristic to be able to solve this problem. The solutions of the approach integrating the cost allocation method are then compared to the solutions of the exact model as proposed in Subsection 4.1.

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

5.1. Outline of the heuristic

In this section, we explain in detail a heuristic that is able to synchronize with soft time windows and integrate the cost allocation. This implies that a solution provided by this heuristic does not only consist of an operational plan with a low transportation cost, it also encompasses the distribution of the cost of that operational plan.

As a delivery date cannot be changed without knowledge of the amount of profit allocated to the partner subject to the change, the cost allocation has to be (re)calculated *during* the optimization phase to be able to determine the final operational plan. 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 beforehand. This statement can be extended as follows. To be able to construct the solution for a coalition S, the values of all coalitions with a number of partners smaller than $|S|$ should be known. The cost of an empty coalition is given, i.e. a zero cost. We have therefore enough information to create the operational plan and calculate the resulting cost allocation for all partners working alone ($|S| = 1$). Then, we can solve the coalitions containing two partners ($|S| = 2$), and so on.

Moreover, for each subcoalition, the baseline has to be set. Before creating the final operational plan in which each partner allows its orders to be synchronized, the maximal allocation for each rigid partner has to be determined. This implies that for each subcoalition, the operational plans for all different possible combinations of states (a state is either rigid or flexible) are created. We start from the solution in which no partner allows its delivery dates to be changed, to the solutions in which only one partner allows its delivery dates to be changed, to the final plan in which all partners allow their delivery dates to be changed. This sequence is chosen to ensure that in each following situation, a good baseline can be determined. This 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 $a = \max(\varphi_a^{a=\text{rigid},b=\text{rigid},c=\text{rigid}}, \varphi_a^{a=\text{rigid},b=\text{flex},c=\text{rigid}})$

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

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

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

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

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

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

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

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

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

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

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

This results in an iterative procedure that moves from an empty coalition 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). This procedure is depicted in Algorithm 1.

For each subcoalition and for each given combination of rigid and flexible partners, we find a feasible solution with the lowest transportation cost. To create feasible solutions, there are four neighbourhoods defined, of which a detailed explanation can be found in Sections 5.3–5.6. How to determine whether a solution is feasible, is discussed in Section 5.2. When a feasible solution is found, this is stored in the archive. At the end of each iteration, the solution with the lowest transportation cost is selected.

5.2. Check feasibility

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

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

```
$t \leftarrow 0$
**while** $t \leq n$ **do**
    **forall the** *subcoalitions $S$ with t partners* **do**
        $a \leftarrow 0$
        **forall the** *possible combinations of states $U$ 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
                Remove and add orders (Section 5.5)
                archive ←Check feasibility
                Solve an exact model
                Random improvements (Section 5.6)
                archive ← Check feasibility
                Solve an exact model
            **end**
            select solution with the lowest transportation cost out of the archive with all feasible
            solutions
            $a \leftarrow a + 1$
        **end**
        $t \leftarrow t + 1$
    **end**
**end**
```

Algorithm 1: Outline of the heuristic integrating the cost allocation

An additional symbol d^{i*} is needed, which specifies the date d_j of the trip in the current solution that contains order i.

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 optimizing, swapping or removing and adding orders. If the solution is feasible, it is stored in the archive, of which the best solution will be selected.

5.3. Solve an exact model

Each iteration starts with a solution that considers the upper and lower limits of the time windows, but does not consider or use the information regarding the costs that each partner links 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 exact model of Section 4.1. As no information regarding C_i can be used, we set $C_i = \varepsilon$, with ε being a very small number (to minimize the total number of changes in delivery dates in a solution).

As some of the partners might be rigid, the lower and upper limits A_i and B_i are adjusted in these cases. For all orders i that belong to rigid partners, $A_i = B_i = 0$.

Finally, this model is not only solved at the beginning of each iteration. After each step when a feasible solution is possible, this model is solved again to try to find a similar solution with a lower transportation cost. However, to create a new solution, the steps made by the other neighbourhoods are taken into account. More specifically, when an order i is either swapped,removed and added or a random improvement has occurred during another phase, this 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 removed and added, cannot be changed further away from its original delivery date than in the current operational plan.

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

To solve these models optimally, the exact solver Gurobi is used (*http://www.gurobi.com*). Finding the optimal solution however is time-consuming. We can therefore solve only smaller instances of the problem in a reasonable amount of time. However, the objective of this paper is not to create the best performing algorithm. We use exact methods to ensure that our results are comparable to the results when solving the model in Section 4.1 optimally.

5.4. Swap orders

The first neighbourhood that is explored is swapping 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 solving the exact model in Section 5.3. It can therefore be regarded as a lower bound, as no operational plan with a lower cost can be found. The objective of the swapping procedure is to rearrange the division of changes in delivery dates among the partners with a minimal cost increase.

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 adding and removing an order generally increases the transportation cost more than swapping orders, the orders are first swapped and in a later stage removed and added.

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 combination of states, i.e. the situation in which in the combination of states of all the partners is equal to the combination of states in the current iteration, with the exception of the partner that owns the order i, who is always rigid. This combination of states is symbolized by U^*.

In the baseline as well as in the currently proposed operational plan, we seek out 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 those trip, we determine the total number of pallets in the trips and consequently, the price per pallet that the 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, times 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 stated that the solution provided by solving the model in Section 5.3 exactly can be considered as 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 orders that have received a new delivery date, but the transportation cost decrease due to the change is not able to cover the additional cost of changing the delivery date.

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

In order to swap an order, we look for viable candidates to swap an order 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 $violation_k \leq 0$. 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.

5.5. Remove and add orders

If the solution proposed by the swapping procedure, followed by solving it with an exact model, does not provide a feasible solution, the remove and add-procedure starts. The objective of this procedure is to change the delivery dates of the orders of the partners that are not satisfied with

```
$vltn \leftarrow 0$
$violation_{vltn} \leftarrow 0$
forall the partners $k$ in $S$ do
    calculate violation$_k$
    if violation$_k$ $>$ violation$_{vltn}$ then
        $vltn \leftarrow k$
    end
end
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 $\frac{1+\max(0,violation_{vltn})}{\sum_k \max(0,violation_k)} profit_{o,i} + \frac{1+\max(0,violation_{\text{partner of order } o})}{\sum_k \max(0,violation_k)} profit_{i,o}$
        $violation_{vltn} \leftarrow violation_{vltn} - profit_{o,i}$
    end
    calculate cost and divide it among players
    forall the partners $k$ in $S$ do
        calculate violation$_k$ and determine partner $vltn$
    end
end
```

Algorithm 2: Integration of the cost allocation method

the solution, closer to the original delivery date without changing the delivery dates of the other orders.

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 — cost being transportation cost plus the cost of changing delivery dates —. This is repeated until $violation_{vltn} = 0$ or until each order has been removed and added again.

If partner $vltn$ remains unchanged, we start the final search, in which we randomly improve the feasibility of the solution. If another partner has the highest violated constraints, the swapping procedure restarts again.

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, we randomly chose an order belonging to partner $vltn$. The extent to which the delivery date of that order can be changed, is limited to the number of days the delivery date

Table 1: Additional costs of changing delivery dates, cost allocation and profit for synchronization with soft time windows and an integrated cost allocation

	cost of changing delivery dates	allocated cost	profit
A	0	28.463	1.487
B	4	24.003	5.947
C	0	28.463	1.487

is changed in the solution at hand, minus one. This limitation is enforced by adding a constraint to the model in Section 5.3.

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

When no feasible solution is found, the swapping procedure restarts, starting from the solution provided from this search.

6. Results

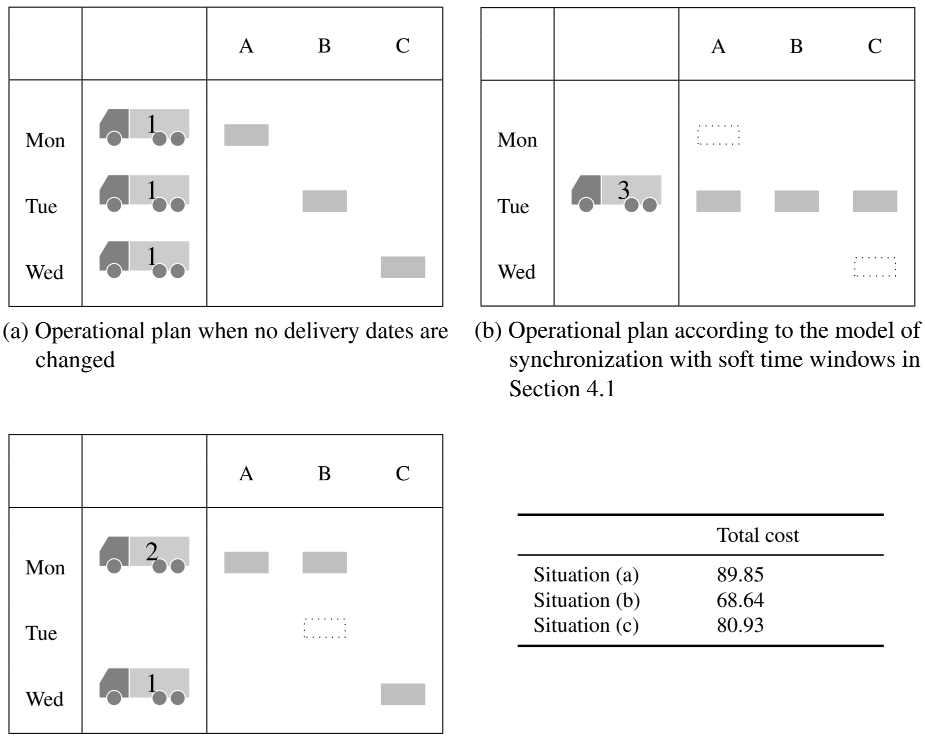
6.1. Example of the benefits of integrating the cost allocation mechanism

The difference between the synchronization with soft time windows in Section 4.1 — minimizing the transportation cost and the penalty cost — and the synchronization with soft time windows and an integrated cost allocation in Section 4.2 — minimizing the transportation cost given constraints that integrate the cost allocation — is easily demonstrated with the simple example shown in Figure 3.

In this example, there are 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 lies within their upper and lower bounds and their soft time window constraints. Partner 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: Partners A and B suffer a cost of 4 euro, while partner C needs 10 euro to compensate the change of a delivery date for one day.

Based on the pace list in Table 4, 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 in



(a) Operational plan when no delivery dates are changed

(b) Operational plan according to the model of synchronization with soft time windows in Section 4.1

(c) Operational plan according to the model of synchronization with soft time windows and integrated cost allocation in Section 4.2

	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 4

Figure 3: Example of the difference between minimizing the transportation cost with or without integrating the cost allocation method

Table 2: Additional cost of changing the delivery dates of the orders for the model synchronization with soft time windows in Section 4.1, and possible cost allocations and profit

		Shapley allocation		**Pay cost of changes**	
	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

this example is lower than when integrating the cost allocation (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 these costs proves to be very difficult. When allocating the costs by the Shapley formula, we create an operational plan for each subcoalition. The weighted sum of the resulting marginal costs is then the allocation. As can be seen from Table 2, the profit allocated to partner C does not exceed the cost of changing the delivery date of that order. 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.

It is possible 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. 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. We therefore refrain from directly compensating the cost of changing delivery dates.

6.2. Additional tests

The integrated cost allocation has the advantage of discouraging partners to report false costs. The example in Section 6.1 shows that if company C reported 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 where 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 synchronization with soft time windows in Section 4.1 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, a company should not be rewarded by reporting a cost of changing the delivery date that is too high. Therefore, the cost of changing the dates is not directly compensated. Instead, for both models, we allocate the transportation cost by the Shapley value, disregarding the costs of changing dates. However, it is still possible for a company to benefit from reporting a higher cost. As the penalty cost is minimized in the synchronization with soft time windows in Section 4.1, the model will always opt — given that the transportation cost remains the same — to change the dates of the order of the company with the lowest cost of changing a delivery date. 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 model with soft time windows and an integrated cost allocation to the model with with soft time windows in Section 4.1, we have created ten instances. The instances consist out 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 and 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 final 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 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 Table 3. 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 change delivery dates for more days to further lower the transportation cost.

However, when the cost of a partner is substantially higher than the others, it is possible that the Constraints 10 are so restrictive that the total transportation cost of the solution of the model integrating the cost allocation is higher. In instance 1-1 and 1-2, 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 a model integrating the cost allocation.

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 (with

Table 3: The difference between synchronization with soft time windows and synchronization with soft time windows and an integrated cost allocation

	Diff total transp. cost	Diff $\sum e_{all}$	Diff φ_A	Diff $\sum e_A$
$A_A = B_A = A_B = B_B = A_C = B_C =, C_A = C_B = C_C = 10$				
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 1-3	0.00	0	0.00	0
Instance 2-3	0.00	0	2.19	0
Instance 3-3	-5.45	1	-2.72	0
Instance 4-3	-2.35	1	-1.18	0
Instance 5-3	0.00	0	0.00	0
Instance 6-3	0.00	0	0.00	0
Instance 7-3	-25.82	4	-13.21	1
Instance 8-3	-21.85	3	-7.51	2
Instance 9-3	0.00	0	0.00	0
Instance 10-3	0.00	0	0.00	1

exception of instance 8-2), 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 model in Section 4.1. 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 modelled in Section 4.1 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 constraints therefore does not favour the partner with slightly higher cost of changing delivery dates, which is the case when optimizing the sum of the transportation cost and the cost of changing delivery dates. As the costs of changing delivery dates are often an estimate, and it should be avoided to give incentives that will overprice this cost, synchronization with soft time windows and integrated cost allocation proposed in Section 4.2 has a clear advantage over the model without an integrated cost allocation.

7. Conclusion

Although much effort has been done on developing tools to optimize cooperative efforts, most of them neglect that a solution should be *jointly* optimal. In this paper, we have discussed a cooperative transportation problem, in which partners are willing to bundle their orders, and agree that their orders are synchronized. However, delivering orders on a different day than the actual requested delivery date, might result into additional costs for the partner which the order is delivered on a different date. Such an effect might be easily measurable (e.g. a penalty fee to be paid), but is in many cases difficult to determine.

We state that, instead of minimizing the total cost — i.e., the total transportation cost and the cost of changing delivery dates —, a more realistic approach is to minimize transportation cost, but only allow the delivery dates to be changed if this will generate sufficient profits *for the partner undergoing the change*. Therefore, the cost allocation should be known during the optimization phase, rather than optimizing and dividing the cost afterwards. We have chosen to develop a heuristic that integrates the Shapley value.

We find that the model we solve is capable to provide solutions (an operational plan and a cost allocation) that will satisfy all partners. It can often find a solution with a lower total transportation cost. Moreover, when a partner has a very high cost of changing delivery dates, it will alter the operational plan in such a way that the number of days an order of that partner is altered further away from its original date is limited, trading the changes to the delivery dates for a higher allocated cost. 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 favour of the partner with a higher cost of changing delivery dates.

Finally, the algorithm proposed in this paper can easily be extended to other collaborative problems. The main focus of the neighbourhoods searched is to find feasible low-cost solutions with regard to the Shapley constraints. To find a low cost, an exact solver is used. However, to be able to solve large instances of this problem, or even extending the problem to e.g. vehicle routing problems, this exact solver can easily be converted to a well-performing metaheuristic.

Acknowledgements

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

References

J.-F. Audy, S. D'Amours, and L.-M. Rousseau. Cost allocation in the establishment of a collaborative transportation agreement—an application in the furniture industry. *Journal of the Operational Research Society*, (62):960 –970, 2011.

M. Barratt. Understanding the meaning of collaboration in the supply chain. *Supply Chain Management: An International Journal*, 9(1):30–42, 2004.

T. Boyacı and G. Gallego. Coordinating pricing and inventory replenishment policies for one wholesaler and one or more geographically dispersed retailers. *International Journal of Production Economics*, 77(2):95–111, 2002.

G.P. Cachon. Supply chain coordination with contracts. *Handbooks in operations research and management science*, 11:229–340, 2003.

C.L. Chen and W.C. Lee. Multi-objective optimization of multi-echelon supply chain networks with uncertain product demands and prices. *Computers & Chemical Engineering*, 28(6): 1131–1144, 2004.

J.-M. Chen and T.-H. Chen. The multi-item replenishment problem in a two-echelon supply chain: the effect of centralization versus decentralization. *Computers & Operations Research*, 32(12):3191–3207, 2005.

A.J. Clark and H. Scarf. Optimal policies for a multi-echelon inventory problem. *Management science*, 6(4):475–490, 1960.

W. Conen. 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*, 2002.

F. Cruijssen, M. Cools, and W. Dullaert. Horizontal cooperation in logistics: Opportunities and impediments. *Transportation Research Part E: Logistics and Transportation Review*, 46(3): 22–39, 2007.

P. Davidsson, L. Henesey, L. Ramstedt, J. Törnquist, and F. Wernstedt. An analysis of agent-based approaches to transport logistics. *Transportation Research part C: emerging technologies*, 13 (4):255–271, 2005.

G. Dudek and H. Stadtler. Negotiation-based collaborative planning between supply chains partners. *European Journal of Operational Research*, 163(3):668–687, 2005.

M. Frisk, M. Göthe-Lundgren, K. Jörnsten, and M. Rönnqvist. Cost allocation in collaborative forest transportation. *European Journal of Operational Research*, 205(2):448–458, 2010.

A. Guinouet and M. Jordans. Jsp and hf-czechforge to bundle plastics and steel shipmentsfrom czech republic to germany. Technical report.

R. Ireland and R. Bruce. Cpfr-only the beginning of collaboration. *Supply Chain Management Review*, 4(September/October):80–88, 2000.

M.A. Krajewska, H. Kopfer, G. Laporte, S. Ropke, and G. Zaccour. Horizontal cooperation among freight carriers: request allocation and profit sharing. *Journal of the Operational Research Society*, 59(11):1483–1491, 2008.

H.L. Lee, V. Padmanabhan, and S. Whang. Information distortion in a supply chain: the bullwhip effect. *Management science*, 43(4):546–558, 1997.

N. Lehoux, S. D'Amours, and A. Langevin. A win-win collaboration approach for a two-echelon supply chain: a case study in the pulp and paper industry. *European Journal of Industrial Engineering*, 4(4):493–514, 2010.

P. Liu, Yaohua Wu, and N. Xu. Allocating collaborative profit in less-than-truckload carrier alliance. *Journal of Service Science and Management*, 3:143–149, 2010.

E. Loehman and A. Whinston. An axiomatic approach to cost allocation for public investment. *Public Finance Quarterly*, pages 236–251, April 1974.

O.-O. Özener, O. Ergun, and M. Savelsbergh. Lane-exchange mechanisms for truckload carrier collaboration. *Transportation Science*, 45(1):1–17, 2011.

B. Pal, S.S. Sana, and K. Chaudhuri. A three layer multi-item production–inventory model for multiple suppliers and retailers. *Economic Modelling*, 29(6):2704–2710, 2012.

L.S. Shapley. A value for n-person games. *Annals of Mathematics Studies*, 28:307–317, 1953.

M.M. Solomon, E.K. Baker, and J.R. Schaffer. Vehicle routing and scheduling problems with time window constraints: Efficient implementations of solution improvement procedures. In B. Golden and A. Assad, editors, *Vehicle routing: Methods and studies*, page 85–106. Elsevier Science Publishers, 1988.

D. Soons. 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.

E. Taillard, P. Badeau, Mi. Gendreau, F. Guertin, and J.-Y. Potvin. A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation Science*, 31(2):170–186, 1997.

D.J. Thomas and P.M. Griffin. Coordinated supply chain management. *European Journal of Operational Research*, 94(1):1–15, 1996.

A. Van Breedam, B. Vannieuwenhuyse, and S. Verstrepen. The making of the world's first orchestrated horizontal collaboration. Presentation at the 2nd horizontal collaboration summit, May 2011.

C. Vanovermeire, K. Sörensen, A. Van Breedam, B. Vannieuwenhuyse, and S. Verstrepen. Efficient supply chains through flexible horizontal collaboration. Working Papers 2012007, University of Antwerp, Faculty of Applied Economics, March 2012. URL `http://ideas.repec.org/p/ant/wpaper/2012007.html`.

C. Vanovermeire, D. Vercruysse, and K. Sörensen. Analysis of different cost allocation methods in a collaborative transport setting. Working papers, University of Antwerp, Faculty of Applied Economics, 2013. URL `http://EconPapers.repec.org/RePEc:ant:wpaper:2013009`.

S. Verstrepen and D. 't Hooft. Co3 case study: Retail collaboration in france. Technical report, CO^3 Collaboration Concepts for Co-modality, 2011.

K. Verweij. Horizontal supply chain collaboration: experiences from benelux case studies, November 2009.

X. Zou, S. Pokharel, and R. Piplani. Channel coordination in an assembly system facing uncertain demand with synchronized processing time and delivery quantity. *International journal of production research*, 42(22):4673–4689, 2004.

Acknowledgements

Research funded by a Ph.D. grant of the Agency for Innovation by Science and Technology (IWT).

A. Pace list

B. Full output additional tests

Table 4: 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		

Table 5: Full output additional tests

	Results without integrating the Shapley value								Results when integrating the Shapley value							
	φ_A	φ_B	φ_C	Total cost	$\sum e_a$	$\sum e_b$	$\sum e_c$	$\sum e_i$	φ_A	φ_B	φ_C	total cost	$\sum e_a$	$\sum e_b$	$\sum e_c$	$\sum e_{all}$
$A_A = A_B = A_C = 1, C_A = C_B = C_C = 10$																
Instance 1	1480.25	358.72	73.62	1912.59	2	1	1	4	1480.25	358.72	73.62	1912.59	2	1	1	4
Instance 2	338.40	350.42	543.11	1231.93	3	3	1	7	340.59	347.96	543.39	1231.93	2	4	1	7
Instance 3	880.05	227.28	242.88	1350.21	0	0	0	0	877.33	227.28	240.16	1344.76	0	0	1	1
Instance 4	236.25	293.40	851.28	1380.93	0	0	2	2	235.08	293.40	850.11	1378.58	0	0	3	3
Instance 5	52.06	570.28	758.76	1381.10	1	0	3	4	52.06	570.28	758.76	1381.10	1	0	3	4
Instance 6	589.29	590.43	11.02	1190.73	1	0	1	2	589.29	590.43	11.02	1190.73	0	1	1	2
Instance 7	684.78	229.51	229.52	1143.81	1	0	0	1	671.57	216.30	230.12	1117.99	2	1	2	5
Instance 8	348.18	154.29	281.32	783.79	1	0	0	1	340.68	146.78	274.49	761.94	3	0	1	4
Instance 9	797.73	155.73	609.73	1563.18	0	2	0	2	797.73	155.73	609.73	1563.18	0	2	0	2
Instance 10	472.90	228.43	430.72	1132.05	2	0	2	4	472.90	228.43	430.72	1132.05	2	0	2	4
$A_A = 5, A_B = A_C = 1, C_A = 50, C_B = C_C = 10$																
Instance 1-1	1480.25	358.72	73.62	1912.59	2	1	1	4	1525.28	403.74	118.65	2047.67	0	1	1	2
Instance 2-1	338.19	350.21	543.53	1231.93	2	4	1	7	408.80	416.17	543.39	1368.35	0	3	1	4
Instance 3-1	880.05	227.28	242.88	1350.21	0	0	0	0	877.33	227.28	240.16	1344.76	0	0	1	1
Instance 4-1	236.25	293.40	851.28	1380.93	0	0	2	2	235.08	293.40	850.11	1378.58	0	0	3	3
Instance 5-1	62.66	570.28	769.36	1402.30	0	0	3	3	59.13	577.35	765.83	1402.30	0	0	3	3
Instance 6-1	589.29	590.43	11.02	1190.73	0	1	1	2	589.29	590.43	11.02	1190.73	0	1	1	2
Instance 7-1	754.47	167.55	233.39	1155.41	0	0	2	2	747.63	170.97	236.81	1155.41	0	0	2	2
Instance 8-1	378.95	103.70	301.14	783.79	1	0	0	1	378.28	103.03	301.14	782.45	1	1	0	2
Instance 9-1	799.67	157.68	605.83	1563.18	0	2	0	2	799.67	157.68	605.83	1563.18	0	2	0	2
Instance 10-1	480.55	213.13	438.37	1132.05	1	0	3	4	456.53	213.13	462.40	1132.05	2	0	2	4
$A_A = A_B = A_C = 3, C_A = 21, C_B = C_C = 20$																
Instance 1-2	1480.25	358.72	73.62	1912.59	0	2	2	4	1480.25	358.72	73.62	1912.59	2	2	0	4
Instance 2-2	338.40	350.42	543.11	1231.93	0	6	0	6	340.59	347.96	543.39	1231.93	1	5	0	6
Instance 3-2	880.05	227.28	242.88	1350.21	0	0	0	0	876.89	227.28	239.72	1343.88	0	0	3	3
Instance 4-2	236.08	243.13	915.81	1395.02	0	0	1	1	230.31	231.48	902.99	1364.78	0	2	4	6
Instance 5-2	52.06	632.16	696.88	1381.10	0	0	4	4	52.06	632.16	696.88	1381.10	2	0	2	4
Instance 6-2	598.70	585.72	6.31	1190.73	0	1	1	2	660.18	612.16	64.61	1336.95	0	0	0	0
Instance 7-2	686.68	228.56	228.58	1143.81	1	0	0	1	666.53	217.55	232.16	1116.24	4	1	1	6
Instance 8-2	399.25	107.55	192.16	698.95	1	0	3	4	434.67	183.17	268.46	886.30	0	0	0	0
Instance 9-2	776.66	154.60	631.92	1563.18	0	2	0	2	791.08	169.02	603.08	1563.18	0	2	0	2
Instance 10-2	448.88	228.43	454.75	1132.05	1	0	3	4	433.60	224.72	439.47	1097.78	4	0	4	8
$A_A = A_B = A_C = 1, C_A = 11, C_B = C_C = 10$																
Instance 1-3	1480.25	358.72	73.62	1912.59	2	1	1	4	1480.25	358.72	73.62	1912.59	2	1	1	4
Instance 2-3	338.40	350.42	543.11	1231.93	2	3	2	7	340.59	347.96	543.39	1231.93	2	4	1	7
Instance 3-3	880.05	227.28	242.88	1350.21	0	0	0	0	877.33	227.28	240.16	1344.76	0	0	1	1
Instance 4-3	236.25	293.40	851.28	1380.93	0	0	2	2	235.08	293.40	850.11	1378.58	0	0	3	3
Instance 5-3	52.06	570.28	758.76	1381.10	1	0	3	4	52.06	570.28	758.76	1381.10	1	0	3	4
Instance 6-3	589.29	590.43	11.02	1190.73	0	1	1	2	589.29	590.43	11.02	1190.73	0	1	1	2
Instance 7-3	684.78	229.51	229.52	1143.81	1	0	0	1	671.57	216.30	230.12	1117.99	2	1	2	5
Instance 8-3	348.18	154.29	281.32	783.79	1	0	0	1	340.68	146.78	274.49	761.94	3	0	1	4
Instance 9-3	797.73	155.73	609.73	1563.18	0	2	0	2	797.73	155.73	609.73	1563.18	0	2	0	2
Instance 10-3	472.90	228.43	430.72	1132.05	1	0	3	4	472.90	228.43	430.72	1132.05	2	0	2	4