Agent Based Decomposition of Optimization Problems

Johan Holmgren
Dept. of Systems and Software Engineering, Blekinge Institute of Technology, Box 520, SE-372 25 Ronneby, Sweden
johan.holmgren@bth.se

Jan A. Persson
Dept. of Systems and Software Engineering, Blekinge Institute of Technology, Box 214, SE-374 24 Karlshamn, Sweden
jan.persson@bth.se

Paul Davidsson
Dept. of Systems and Software Engineering, Blekinge Institute of Technology, Box 520, SE-372 25 Ronneby, Sweden
paul.davidsson@bth.se

ABSTRACT

In this paper, we present an agent-based approach for solving an optimization problem using a Dantzig-Wolfe column generation scheme, i.e., a decomposition approach. It has been implemented and tested on an integrated production, inventory, and distribution routing problem. We developed a decomposition model for this optimization problem, which was implemented in the Java programming language, using the Java Agent DEvelopment Framework (JADE) and the ILOG CPLEX mixed integer linear problem solver. The model was validated on a realistic scenario and based on the results, we present estimates of the potential performance gain by using a completely distributed implementation. We also analyze the overhead, in terms of communication costs, imposed by an agent based approach. Further we discuss the advantages and the disadvantages that comes with an agent-based decomposition approach.

Keywords

Agent, multi-agent system, optimization, Dantzig-Wolfe decomposition

1. INTRODUCTION

It has been argued that the strengths and weaknesses of agent-based approaches and classical optimization techniques complement each other well for dynamic resource allocation problems [5]. A number of ways to combine the approaches in order capitalize on their respective strengths were sketched and two of these were implemented and evaluated. The first approach was using an optimization technique for coarse planning and agents for operational replanning, i.e., for performing local adjustments of the initial plan in real-time to handle the actual conditions when and where the plan is executed. In the second approach, optimization was embedded in the agents. This can be seen as a further development of distributed optimization where agent technology is used, e.g., to improve coordination.

In this paper, we describe another approach, namely how an optimization approach can be "agentified" into a multi-agent system (MAS) by using principles of decomposition which is a classical optimization (reformulation) method. One purpose is to detail the concept of agentifying an optimization approach. Further, the purpose is to validate that some positive characteristics can be achieved by this type of approach. The main characteristics studied are confidentiality aspects of information, i.e. keep the information locally when possible, and the potential to achieve performance improvements by distribution and parallelization. Obviously, an agentification comes at a price, which is also discussed in this paper.

In the next section the integration of agent technology and decomposition is outlined, and a real world case is presented in section 3. The application of agent based decomposition is given in section 4, and in section 5, we present our computational experiment, some results and discussions. Finally, some conclusions and directions for future work is given in section 6.

2. DECOMPOSITION AND AGENT TECHNOLOGY

Decomposition in linear optimization, such as the Dantzig-Wolfe (DW) column generation scheme [2], were originally developed for solving large linear optimization problems. Later, decomposition approaches have been used for solving mixed integer linear problems by branch and price [8].

In DW decomposition, a mixed integer linear optimization problem is reformulated in terms of a linear master problem and a set of planning subproblems which produces new plans that are coordinated by the master problems. The linear master problem, which contains a restricted number of variables, is solved in each iteration. From the optimal dual variables obtained from the solution of the restricted master problem, the subproblems produce new plans which are added as new improving variables (or columns), to extend the restricted master problem.

In this paper, we suggest a multi-agent based DW decomposition approach illustrated by the conceptual MAS model presented in figure 1. This hierarchical model contains a coordinator agent and a set of planner agents, where the coordinator agent corresponds to a DW master problem and each planner agent corresponds to a set of DW subproblems. Note that it is possible for a planner agent to handle more than one DW subproblem. This number of subproblems depends on the organizational settings in the

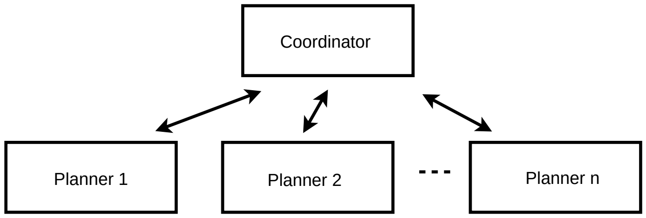

Figure 1: The conceptual MAS model.

studied scenario. A planner agent is responsible for planning some set of resources assisting the coordinator agent in finding a globally optimal solution. The planning procedure is guided by the values of the dual variables which the coordinator uses as a control mechanism to help the planner agents in finding improving plans. Hence, the coordinator is not explicitly dictating the characteristics of the plans to be produced by the planner agents, and the planner agents are thereby to some extent autonomous. The coordinator agent is responsible for coordinating the plans that are produced by the planner agents. Moreover it decides if and when the planner agents should be given some directives of partially fixating the plan, i.e. to fixate the use of some resources. This fixation of plans is necessary in cases where DW-decomposition does not guarantee any feasible (i.e. not integer) solution, which is a common characteristic of DW-decomposition. Such decisions (fixations) must of course be based on previously generated plans, and the planner agents in question are forced to obey the fixations when they create future plans. Note however, that the DW-decomposition approach allows for many different ways of defining the coordinator and the planner agents.

Traditionally, a decomposition algorithm formulated with a master problem and a set of subproblems is executed in a single process (on a single computer), which has access to all pieces of information necessary to run the optimization. This classical approach can not provide confidentiality since the coordinator must be given access to information that might be considered sensitive. With an agent based approach however, where the master problem, and subproblems are represented by different individual agents, it is often (depending on the actual application) possible to run the optimization with less need for sharing sensitive information. For instance, in the case presented in section 4, we have a master problem which coordinates production, transportation and inventory. This master problem needs access e.g. to customer demand forecasts, but it does not need any underlying details about how actual transportation plans and production plans are created, because such information belongs to the subproblems. In a classical approach, however, all such information needs to be shared with the master problem, causing decreased confidentiality.

When the agent system runs on a single computer, there is actually a loss in performance due to the use of agents, because of the increased need for communication. In the distributed solution approach, where the system runs on several computers (or a multi processor computer), there is however a potential gain in computation time and in some cases solution quality since more computing power allows for solving more complex subproblems. The reason for this is that the subproblems can be solved in parallel, which is not the case when the optimization runs on a single processor. The easiness in which an agent system can be distributed over several, geographically or non geographically distributed, computers gives another possible advantage over classical decomposition approaches. Such a decentralized solution approach further makes the system less vulnerable to single point failures of one computing node. In our approach, where the coordinator agent typically retains control of all decisions, but does not need to know any details about how solutions were generated, a failure of the coordinator agent is fatal, because the presence of the coordinator agent is absolutely necessary. However, in the case when the coordinator loose contact with some planners, the rest of the planners might still be able to produce new plans that can be considered by the coordinator. This makes the suggested approach more robust towards planner agent failures.

In section 4 we demonstrate the agent based decomposition approach by presenting a case study for an integrated production, inventory, and distribution routing problem. We have chosen a decomposition formulation which we find attractive particularly from the perspective of having a natural interpretation of dual prices and having real world correspondence to the sub-problems. In general, the decomposition scheme including e.g. master/subproblem formulations, fixations, termination criterias need to be designed to utilize the special characteristics of the studied problem type. We would like to point out that some problems can be extremely difficult to even formulate as a single mixed integer problem, and solving it can be an even more difficult task. According to Fumero and Vercellis [6], problems in this area tend to use an underlying network structure that may be exploited by decomposition approaches. Also, the studied problem class captures the difficulties with distributed decision making since typically, information and/or resources are distributed and the exact conditions, e.g., the demand and the availability of resources, are not known in advance and are changing. Another characteristic is that these problems typically are NP-hard and that they can be extremely difficult to solve, even for very small problem instances. An overview of integrated production, inventory, and distribution routing problem is provided by Lei et al [7].

3. CASE DESCRIPTION

We consider a real world problem with a producer of vegetable oils and a transport operator with a fleet of trucks handling deliveries of finished products to a number of customers. The production, and hence the planning of production and deliveries are driven by, sometimes changing and often late arriving, customer orders and the producer has limited knowledge about the contents of and arrival times of new orders. The actual production is performed in batches at a single production plant with multiple production lines which are scheduled to match the shipping times deduced from the customer orders.

Products are delivered by trucks operated by a single haulier and before loading, the finished products are stored in short-term (rather capacity limited) producer depot inventory. At delivery, products are stored in customer inventories for storage of finished products. Starting with a full (or close to full) load, at the producer, sometimes a truck visits only one customer before returning but sometimes deliveries are grouped together and the truck visits multiple customers in the same trip. The transportation cost includes time based driver cost, distance based fuel cost and vehicle wear cost.

When the transportation demand exceeds the available transportation capacity, it is possible to call in extra capacity to a higher cost. Occasionally, a vehicle can be scheduled for a non empty transport on the return trip from a customer to the producer. Furthermore, time and costs for loading and unloading are considered, and the drivers must follow the European Economic Community (EEC) regulations[1] for driving and resting hours. A typical planning horizon for the production and transport planning is usually less than a week.

A possible utilization of customer demand forecasts would allow the producer to introduce *Vendor Managed Inventory* (VMI) [3], where the producer is responsible for replenishment of customer inventories, for one or more customer. This leads to a higher flexibility in the production and transportation planning, and hopefully, an introduction of VMI can lead to a higher utilization of (often limited) production and transportation resources. We focus on the VMI situation, due to its potential of improved resource utilization, but the presented approach is not limited to VMI, since a tight specification of customer inventories mimics a non-VMI situation.

In figure 2, we present a small transport network containing a factory, inventories, customer depots and trucks, and planners who are responsible for taking decisions about the physical entities.

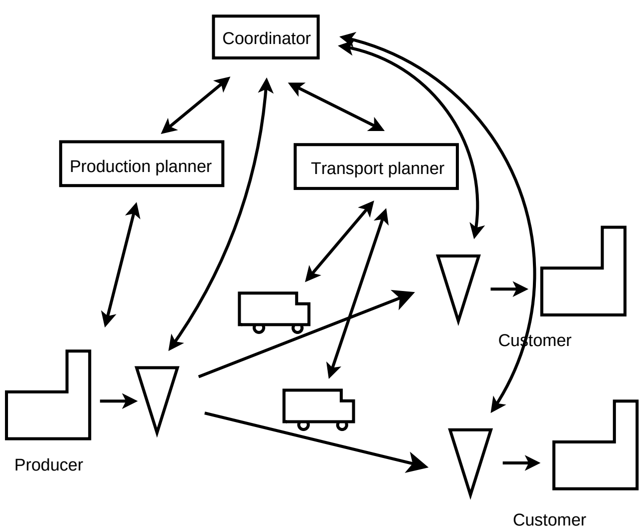

Figure 2: A transport chain with planners, coordinator and physical entities (trucks, inventories, factories and customer depots.

4. CASE SPECIFIC DECOMPOSITION FORMULATION

To illustrate our agent based approach, we designed a Dantzig-Wolfe decomposition scheme to model the integrated production, inventory, and distribution routing problem presented in the previous section. The decomposition formulation contains a master problem, a production scheduling subproblem for construction of production plans, and a transportation subproblem for construction of transportation plans.

Building upon the conceptual agent model in figure 1 we constructed an agent system containing a transport chain coordinator agent, a set of production planner agents and a set of transportation planner agents. In this decomposition formulation, the transport chain coordinator agent maps to the master problem, each production planner agent to a production scheduling subproblem and each transportation planner agent handles a set of transportation planning subproblems.

We let D^P denote the set of producer depots, D^C the set of customer depots, V the inhomogeneous fleet of vehicles, P the set of product types and L the set of production lines available in the model. The planning horizon is represented by an ordered set $T = \{1, 2, \ldots, \bar{t}\}$ of discrete time periods with uniform length τ. A *transportation plan* for a vehicle is defined as the amount of each product delivered to each customer depot and picked-up from each producer depot in each time period throughout the planning horizon. The set of all transportation plans for a vehicle $v \in V$ is denoted R_v, the cost for using r is ψ_r, and x_{dptr} represents a delivered or a picked-up amount of product p at period t by vehicle v. For a customer depot $d \in D^C$, x_{dptr} represents a delivery and for a producer depot $d \in D^P$ it represents a pickup. Similarly, we define a *production plan* for a production line as the amount of each product produced in each time period throughout the planning horizon. We let S_l denote the set of all valid production plans for production line $l \in L$, and ω_s denotes the cost for using plan $s \in S_l$. For production plan $s \in S_l$, located in some depot $d = d(s)$, we let y_{dpts} represent the amount of product p produced in period t. Furthermore, parameter ϱ_{dpt} denotes the demand of product $p \in P$ at customer depot $d \in D^C$ at time period $t \in T$.

Each depot $d \in D$ has an inventory where variable z_{dpt} is used to decide the inventory level of product p in depot d at period t. For depot d, the inventory level of product p at time period t must not fall below a lower bound $\underline{z}_{dpt}$ (which typically corresponds to a safety stock level) and must not exceed a maximum capacity of $\overline{z}_{dpt}$ units. An inventory cost ϕ_{dp} is charged for each unit of product $p \in P$ in stock in depot $d \in D$ between two subsequent periods.

Furthermore, binary decision variables are used to determine which delivery plans and production plans will be used (exactly one transportation plan for each vehicle and exactly one production plan for each production line must be used). Therefore, decision variable v_r determines if transportation plan $r \in R_v$ is used ($v_r = 1$) or not ($v_r = 0$) and w_s if production plan $s \in S_l$ is used ($w_s = 1$) or not ($w_s = 0$).

We introduce variable u_{dpt} to represent how much the storage level of product p in depot d falls below the safety stock level in period t, and q_{dpt} expresses how much the storage level exceeds the corresponding maximum allowed storage level. To avoid violating the safety stock levels and maximum allowed storage levels, or in other words minimizing the usage of the u:s and q:s we introduce penalty costs M^q and as shown in equation (1).

Our Dantzig-Wolfe Master Problem (MP) can now be formulated as to minimize

$$\sum_{p\in P, d\in D, t\in T} (\phi_{dp} z_{dpt} + M^q q_{dpt} + M^u u_{dpt}) + \sum_{r\in\bigcup_{v\in V} R_v} \psi_r v_r + \sum_{s\in\bigcup_{l\in L} S_l} \omega_s w_s, \quad (1)$$

[1]EEC 3820/1985

subject to customer depot inventory balance constraints

$$z_{dpt-1} + \sum_{r \in \bigcup_{v \in V} R_v} v_r x_{dptr} + u_{dpt} - q_{dpt} - \varrho_{dpt} = z_{dpt},$$
$$t \in T, d \in D^C, p \in P, \quad (2)$$

producer depot inventory balance constraints

$$z_{dpt-1} - \sum_{r \in \bigcup_{v \in V} R_v} v_r x_{dptr} + \sum_{s \in \bigcup_{l \in L} S_l} w_s y_{dpts} + u_{dpt} - q_{dpt} = z_{dpt},$$
$$t \in T, d \in D^P, p \in P, \quad (3)$$

storage level constraints

$$\underline{z}_{dpt} \leq z_{dpt} \leq \overline{z}_{dpt}, d \in D, p \in P, t \in T, \quad (4)$$

and convexity constraints

$$\sum_{r \in R_v} v_r = 1, v \in V, \quad (5)$$

$$\sum_{s \in S_l} w_s = 1, l \in L, \quad (6)$$

where $v_r \in \{0,1\} \ \forall \ r \in \bigcup_{v \in V} R_v$, $w_s \in \{0,1\} \ \forall \ s \in \bigcup_{l \in L} S_l$ and $z_{dpt} \geq 0 \ \forall \ d \in D, p \in P, t \in T$.

Before we define the restricted (Dantzig-Wolfe) Master Problem RMP we need to introduce subset $R'_v \subseteq R_v$ of all currently known transportation plans for vehicle $v \in V$ and $S'_l \subseteq S_l$ of all currently known production plans for production line $l \in L$. RMP is identical to MP, except for that we replace all occurrences of R_v with R'_v, all occurrences of S_l with S'_l, and all integer restrictions on variables $\mathbf{v_r}$ and $\mathbf{w_s}$ by constraints

$$0 \leq v_r \leq 1, r \in \bigcup_{v \in V} R_v, \quad (7)$$

$$0 \leq w_s \leq 1, s \in \bigcup_{l \in L} S_l. \quad (8)$$

In other words, RMP is the LP-relaxation of MP, with restrictions on the R_v and S_l sets.

In RMP, we let λ denote the dual variables for constraint group (2), μ the dual variables for group (3), δ the dual variables for constraints (5) and θ the dual variables for constraints (6). A value $\bar{\lambda}_{dpt} > 0$ means that an increased delivery of product p to depot d in period t will improve the objective function value. Also, $\bar{\mu}_{dpt} > 0$ is interpreted to mean that we want to increase the production of product p in depot d in period t. A value $\bar{\lambda}_{dpt} < 0$ on the other hand, means that we want decrease the production or reducing the inventory by transporting products away from depot d.

Figure 3 presents the main algorithm flow for the decomposition approach. RMP is resolved before each request for new production plans or transportation plans is send because we want all new plans to be based on as fresh dual variables as possible. However, we have not tested whether this design option is appropriate or not, but we find it reasonable to believe that generating plans based on fresh dual variables will give fast convergence since that means that all new plans are up-to-date. In a multi processor approach, it is possible (and probably preferable) to send the transportation and production requests in parallel to allow the

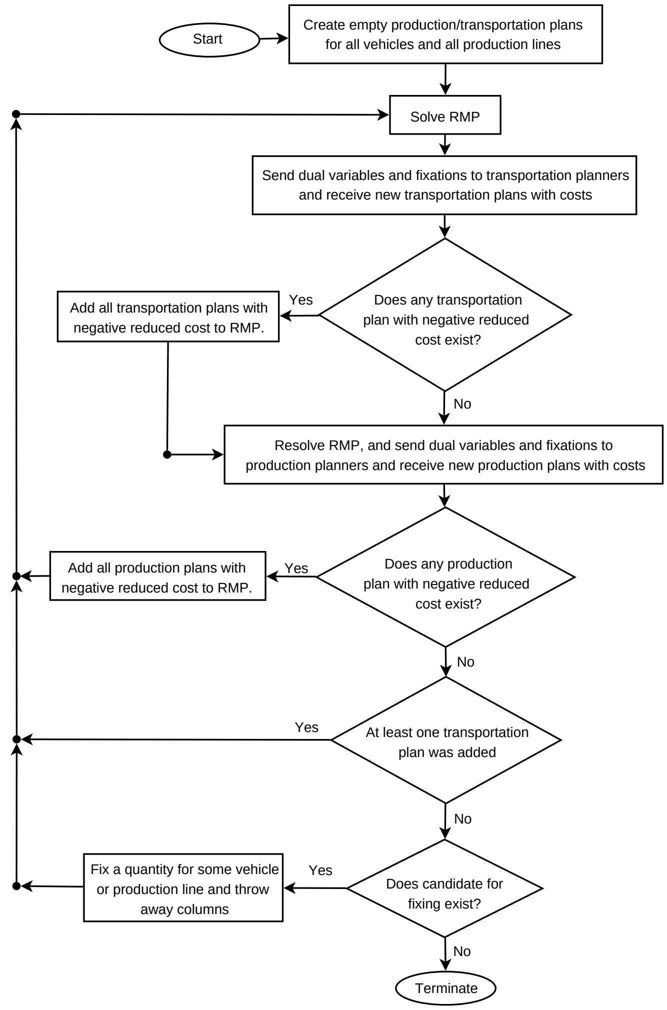

Figure 3: Master problem (coordinator agent) algorithm flow chart.

subproblems to be solved simultaneously. A diagram showing how the coordinator communicates with the transport planners and the production planners in our decomposition approach is presented in figure 4.

The described decomposition approach is for integrated production and transportation, where inventory levels between production and transportation, and between transportation and customer usage are considered. This is a rather general type of problem and it is independent of the choice of production and transportation subproblems. The subproblems must, however, be able to take the dual variables as input and be able to produce new production or transportation solutions.

4.1 Fixations and Termination

In each master problem iteration, the subproblems generate new transportation and production plans based on the current values of the λ, μ, δ, and θ dual variables. Plans with negative reduced cost are added as improving columns/variables to RMP, and when none of the subproblems can produce any new improving column, the optimal solution of RMP is reached. This optimal solution typically

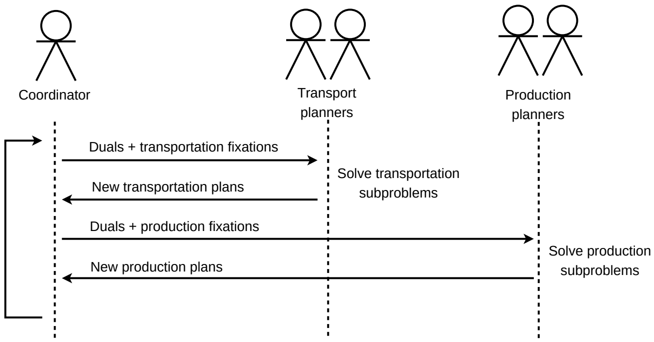

Figure 4: Communication diagram.

gives fractional combinations (see constraint sets (7) and 8)) of plans, which of course is infeasible in integer problem MP. To be able to find an optimal or, at least a heuristically "good" integer solution to MP, some delivery/pickup (depot, period, product, and vehicle) or production (period, product and production line) must be restricted (fixated) to some integer quantity, whenever a fractional optimal solution of RMP is found. Fixations are determined by RMP and communicated to the subproblems to restrict them from violating the fixations when subsequent plans are generated. During the procedure of the algorithm, more and more quantities are fixated, and eventually, when the solution space of the subproblems are too much restricted by the fixations, the algorithm has no other option than to terminate with an integer solution. See [8, ch 11.4] for a general discussion on branching and variable fixations in integer problem column generation algorithms.

In our heuristic fixation strategy, we let $F^T = \{f^T_{dptv} : d \in D, p \in P, t \in T, v \in V\}$ represent the set of all transportation fixations (deliveries and pickups), and $F^P = \{f^P_{ptl} : p \in P, t \in T, l \in L\}$ represent all production fixations available in the model. For convenience, and future reference, we let F^T_v denote all fixations for vehicle $v \in V$, and F^P_l all fixations for production line $l \in L$. In other words, we have one fixation for each depot, product, period and vehicle, and one fixation for each product, period and production line. Initially all fixations are set to 0, meaning that there exist no fixations, but we allow re-fixations to higher values as will be described below. A production fixation of a quantity f^P_{ptl} for some product p, period t, and production line l means that all subsequent production plans for l must contain a production of f^P_{ptl} units or more of product p in period t. Similarly, a transportation fixation f^T_{dptv} for some vehicle v means that all subsequent plans for v must contain a delivery or pickup (pickup if $d \in D^P$ or delivery if $d \in D^C$) of at least f^T_{dptv} units of product p at depot d in period t. In other words, it is possible to re-fixate production, delivery, and pickup quantities to higher values as long as the capacities of the vehicles and production lines are kept, and after each fixation, all columns violating the fixation must be thrown away from RMP. E.g. if we fixate a quantity f^P_{ptl} for production line l, product p, and period t, then all columns for l not containing a production of f^P_{ptl} units for product p in period t must be removed from S'_l.

Our fixation strategy does not guarantee that the decomposition algorithm converges to the optimal integer solution of MP. In order to guarantee this, we would need to create a search tree, where we branch on the fixations. In such a branching tree, we create one branch of f to a higher integer value and one branch of f to a lower integer value, whenever a fixation f is indicated by a fractional value (of delivery/pickup or production). Such a branching tree can however only be generated in theory since we are dealing with an NP-hard problem and therefore, the branching tree will grow exponentially.

In order to decide what to fixate (production, delivery or pickup), we calculate

$$(d', p', t', v') = \underset{d\in D, p\in P, t\in T, v\in V}{\arg\max} \sum_{r\in R'_v} v_r \Lambda^T_{dptr}, \tag{9}$$

where we introduce

$$\Lambda^T_{dptr} = \begin{cases} 0 & \text{if } x_{dptr} = 0 \\ 1 & \text{if } x_{dptr} \neq 0, \end{cases} \tag{10}$$

to restrict equation (9) to merely consider columns with nonzero coefficients. Otherwise, all sums in (9) would take on value 1, since, according to constraint set (5), all transportation convex combinations must sum up to 1. To avoid fixating deliveries/pickups which are represented in all columns in RMP, and more trivially, to disregard those deliveries and pickups that are not represented at all, we require

$$0 < \sum_{r\in R'_{v'}} v_r \Lambda^T_{d'p't'r} < 1. \tag{11}$$

Further, it is only interesting to consider deliveries/pickups that are represented by a higher quantity than previously fixated. Therefore, equation

$$\left| \sum_{r\in R'_{v'}} v_r x_{d'p't'r} \right| > f^T_{d'p't'v'} \tag{12}$$

must hold. From the transportation plans, the most represented vehicle, depot, product and period, which because of equation (12) has a fixation strictly lower than currently represented by RMP is determined. With the same arguments as for the transportation plans, we calculate

$$(l', p', t') = \underset{l\in L, p\in P, t\in T}{\arg\max} \sum_{s\in S'_l} w_s \Lambda^P_{dpts} \tag{13}$$

where

$$\Lambda^P_{dpts} = \begin{cases} 0 & \text{if } y_{dpts} = 0 \\ 1 & \text{if } y_{dpts} \neq 0, \end{cases} \tag{14}$$

and

$$0 < \sum_{s\in S'_l} w_s \Lambda^P_{d'p't's} < 1, \tag{15}$$

and

$$\sum_{s\in S'_{l'}} w_s y_{d(l')p't's} > f^P_{l'p't'} \tag{16}$$

must hold for the production lines. As for the transportation plans, we find the currently most represented production line, product and period. If only (d', p', t', v') exists, or if

$$\sum_{r \in R'_{v'}} v_r \Lambda^T_{d'p't'r} \geq \sum_{s \in S'_{l'}} w_s \Lambda^P_{d(l')p't's}, \tag{17}$$

then $f^T_{d'p't'v'}$ is refixated for vehicle v' according to

$$f^T_{d'p't'v'} = \left\lceil \left| \sum_{r \in R'_{v'}} v_r x_{d'p't'r} \right| \right\rceil. \tag{18}$$

If on the other hand, only (l', p', t') exists, or if

$$\sum_{r \in R'_{v'}} v_r \Lambda^T_{d'p't'r} < \sum_{s \in S'_{l'}} w_s \Lambda^P_{d(l')p't's}, \tag{19}$$

then $f^P_{l'p't'}$ will be refixated for production line l' according to

$$f^P_{l'p't'} = \left\lceil \sum_{s \in S'_{l'}} w_s y_{d(l')p't's} \right\rceil. \tag{20}$$

The absolute values in equations (12) and (18) are introduced since pickups at producer depots are represented by negative numbers in the transportation plans, and all fixations are by definition non negative.

If neither (d', p', t', v') or (l', p', t') could be found, there might still exist any two columns $r', r'' \in R'_{v'}$ or $v' \in V$, or $s', s' \in S'_{l'}$ (for a vehicle $v' \in V$ or a production line $l' \in L$) with different coefficients in the optimal solution of RMP. This scenario can occur if equation (11) or (15) equals 1 for some delivery, pickup or production and if $x_{d'p't'r'} \neq x_{d'p't'r''}$ or $y_{d'p't's'} \neq y_{d'p't's''}$ for any $d' \in D$, $p' \in P$ and $t' \in T$. Then, we fixate

$$f^T_{d'p't'v'} = \left\lceil \left| \sum_{r \in R'_{v'}} v_r x_{d'p't'r} \right| \right\rceil \tag{21}$$

or

$$f^P_{l'p't'} = \left\lceil \sum_{s \in S'_{l'}} w_s y_{d(l')p't's} \right\rceil, \tag{22}$$

to be able to converge towards a integer solution of MP.

For the fixated object (v' or l') we remove all columns ($r \in R'_{v'}$ or $s \in S'_{l'}$) with parameter

$$x_{d'p't'r} \neq \left\lceil \sum_{r \in R'_{v'}} v_r x_{d'p't'r} \right\rceil \text{ for } v', \tag{23}$$

if we fixate either a delivery or a pickup, or

$$y_{d(l')p't's} \neq \left\lceil \sum_{s \in S'_{l'}} w_s y_{d(l')p't's} \right\rceil \text{ for } l', \tag{24}$$

if we fixate a production.

4.2 Subproblem formulations

In our integrated production, inventory, and distribution routing problem, we define two types of subproblems, one single vehicle transportation subproblem and one single production line production scheduling problem. Each vehicle (and each production line) has its own problem instance of the same type. Since each agent constitutes an autonomous entity, possibly with individual restrictions and requirements, it is however natural to solve different types of subproblems for different vehicles and production lines.

4.2.1 *Transportation subproblem*

We formulate a transportation subproblem, using a hierarchical approach, with two separate problems; a routing subproblem and a product assignment subproblem. The routing subproblem is formulated as a side constrained shortest path problem and it determines which depots (the route) the vehicle will visit throughout the planning horizon. Given the optimal route (output from the routing problem), a product assignment problem is formulated to decide how much of each product will be sent on each link, thus determining how much will be picked up and delivered for each visit in the route.

The routing subproblem RP_v for vehicle v can be formulated as standard minimum cost flow problem as to minimize

$$\sum_{(n_i,n_j) \in \mathcal{A}_v} (c^{\text{RP}_v}_{(n_i,n_j)} - \eta_{(n_i n_j)}) x^{\text{RP}_v}_{(n_i,n_j)} \tag{25}$$

subject to

$$\sum_{n_k:(n_k,n_i) \in \mathcal{A}_v} x^{\text{RP}_v}_{(n_k,n_i)} - \sum_{n_j:(n_i,n_j) \in \mathcal{A}_v} x^{\text{RP}_v}_{(n_i,n_j)} = b_i, n_i \in \mathcal{N}_v, \tag{26}$$

where $\mathcal{N}_v$ is a set of nodes and $\mathcal{A}_v$ is a set of arcs in a directed graph representing a time expanded transportation network. Variable $x^{\text{RP}_v}_{(n_i,n_j)} \in \{0, 1\}$ determines the usage of arc $(n_i, n_j) \in \mathcal{A}_v$ such that $x^{\text{RP}_v}_{(n_i,n_j)} = 1$ if arc (n_i, n_j) is used in the solution and $x^{\text{RP}_v}_{(n_i,n_j)} = 0$ otherwise. The "dual variable" representation $\eta_{(n_i,n_j)}$ for arc $(n_i, n_j) \in \mathcal{A}_v$, which is defined as a function of the actual dual variables μ and λ, describes an extra cost/discount which is applied to the actual cost $c^{\text{RP}_v}_{(n_i,n_j)}$ of (n_i, n_j). Constraint set (26) specifies the node balance constraints where b_i follows the rules in equation (27). Node $n_s \in \mathcal{N}_v$ denotes the node where v is situated at the beginning of the planning period and node a is an artificial end node that allows v to be at any node at the end of the planning period. The node balance parameter b_{n_i} for node n_i is defined as

$$b_{n_i} = \begin{cases} -1 & \text{if } n_i = n_s \\ 1 & \text{if } n_i = a \\ 0 & \text{otherwise.} \end{cases} \tag{27}$$

Given the optimal route O_v determined by RP_v (defined as a set of *outbound trips*), a product assignment problem is formulated to decide how much of each product will be sent on each link, thus determining how much will picked up and delivered for each visit in the route. Actually, the problem of how to assign products to the shortest paths separates into one subproblem (ASP_o) for each outbound trip $o \in O_v$. We define an outbound trip for a vehicle as a trip that starts in a producer depot, visits a number of customers and ends when the vehicle returns to some producer, i.e. the last customer depot before returning is included in the trip.

We introduce binary decision variable $x^{\text{ASP}_o}_{ijp} \in \{0, 1\}, p \in P, i \in I_p, j \in J_o \setminus \{1\}$ assuming value 1 if the i :th item of product p is delivered to the j:th depot in outbound trip o and 0 otherwise. Here $I_p = \{1, \ldots, \Phi_p\}$ is an ordered index set over the number of products that can be loaded on v, and J_o is an index set over the visits in outbound trip o, starting with index 1 at the producer depot. From the maximum capacity (weight capacity φ_v^{weight} and volume capacity φ_v^{volume}) of v, we get for each $p \in P$ the maximum number

$$\Phi_p = \left\lfloor \min\left\{ \frac{\varphi_v^{weight}}{weight(p)}, \frac{\varphi_v^{volume}}{volume(p)} \right\} \right\rfloor \quad (28)$$

of items (of product p) that can be loaded on v. Note that Φ_p is restricted either by the volume capacity or by the weight capacity of vehicle v.

We formulate our product assignment problem ASP_o as maximize

$$\sum_{p \in P} \sum_{i \in I_p} \sum_{j \in J_o \setminus \{1\}} \left(\lambda_{d(\Delta_j)pt(\Delta_j)} - \mu_{d(\Delta_1)pt(\Delta_1)} \right) x_{ijp}^{\mathrm{ASP}_o}, \quad (29)$$

subject to

$$\sum_{p \in P} \sum_{i \in I_p} \sum_{j \in J_o \setminus \{1\}} weight(p) \cdot x_{ijp}^{\mathrm{ASP}_o} \le \varphi_v^{weight}, \quad (30)$$

$$\sum_{p \in P} \sum_{i \in I_p} \sum_{j \in J_o \setminus \{1\}} volume(p) \cdot x_{ijp}^{\mathrm{ASP}_o} \le \varphi_v^{volume}, \quad (31)$$

$$x_{i+1jp}^{\mathrm{ASP}_o} \le x_{ijp}^{\mathrm{ASP}_o}, p \in P, i \in I_p \setminus \{\Phi_p\}, j \in J_o \setminus \{1\}. \quad (32)$$

Constraint (30) and (31) expresses the weight and volume restrictions on v and constraints (32) says that item i of a product must be delivered to a depot before item $i+1$ is delivered. ASP_o assigns products to the optimal route O_v, and the result is a transportation plan.

The optimal objective function value of the transportation subproblem for vehicle v is now given as

$$\sum_{(n_i,n_j) \in \mathcal{A}_v} c_{(n_i,n_j)}^{\mathrm{RP}_v} x_{(n_i,n_j)}^{*\mathrm{RP}_v} + \sum_{o \in O_v} \mathrm{ASP}_o^* - \delta_v \quad (33)$$

where $x_{(n_i,n_j)}^{*\mathrm{RP}_v}$ is the optimal value of variable $x_{(n_i,n_j)}^{\mathrm{RP}_v}$, and ASP_o^* denotes the optimal objective value of ASP_o.

For transportation, a quantity can be fixated either at a producer depot or at a customer depot, and the two cases are handled differently in the routing problem and in the product assignment problem. We let $d(f) \in D$ denote the depot, and $t(f) \in T$ the period, and $p(f)$ the product represented by a fixation $f \in F_v^T$. Now, in routing problem RP_v we instead add constraint

$$\sum_{n_j : d(n_j)=d(f), t(n_j)=t(f)} \sum_{n_k : (n_k,n_j) \in \mathcal{A}_v} x_{(n_k,n_j)}^{\mathrm{RP}_v} = 1, \quad (34)$$

for each customer depot fixation $f \in F_v^T : d(f) \in D^C$. For each production fixation $f \in F_v^T : d(f) \in D^P$, we add a constraint

$$\sum_{n_i : d(n_i)=d(f), t(n_i)=t(f)} \sum_{n_k : (n_i,n_k) \in \mathcal{A}_v} x_{(n_i,n_k)}^{\mathrm{RP}_v} = 1. \quad (35)$$

Constraint (34) says that vehicle v must arrive at depot $d(f)$ in period $t(f)$, and constraint (35) mean that v must leave $d(f)$ in period $t(f)$. Note that these fixation constraints remove the integer properties of the LP relaxation of the routing problem.

For the product assignment problems, we add a constraint

$$\sum_{i \in \{1, \ldots \Phi_{p(f)}\}} \sum_{j \in J_o \setminus \{1\}} x_{ijp(f)}^{\mathrm{ASP}_o} \ge f \quad (36)$$

to some assignment problem ASP_o (representing the fixation f) for each producer fixation $f \in F_v^T : d(f) \in D^P$. This forces v to pickup at least f units of product $p(f)$ from producer depot $d(f)$ in period $t(f)$. For each customer fixation $f \in F_v^T : d(f) \in D^C$, we instead add a constraint

$$x_{fjp(f)}^{\mathrm{ASP}_o} = 1, \quad (37)$$

which together with constraint set (32) guarantees that a minimum of f units of product $p(f)$ will be delivered to customer depot $d(f)$ in period $t(f)$.

4.2.2 *Production scheduling subproblem*

A production scheduling subproblem for a production line $l \in L$, aims at finding improving production plans for l based on some current values of the μ dual variables, which are subtracted from the actual production costs in the objective function (38). We let c_{lp}^{prod} denote the cost for production line l to produce one unit of product $p \in P$ and c_{lp}^{setup} denote a fixed setup cost for each period product p is produced.

We let decision variable $x_{pt}^{\mathrm{PSP}_l} \in \mathbb{Z}^+$ determine the amount of product p to produce in period t. All products produced in a period t is assumed to be available for pickup at the same period (t). Binary variable $y_{pt}^{\mathrm{PSP}_l} \in \{0,1\}$ is used to indicate whether there will be a production ($y_{pt}^{\mathrm{PSP}_l} = 1$) of product p in period t or not ($y_{pt}^{\mathrm{PSP}_l} = 0$). Furthermore, we let U denote the maximum number of different product types that can be produced in one period. The production scheduling problem PSP_l can now be defined as minimize

$$\sum_{p \in P} \sum_{t \in T} \left(c_{lp}^{\mathrm{prod}} - \mu_{pt} \right) x_{pt}^{\mathrm{PSP}_l} + \sum_{p \in P} \sum_{t \in T} c_{lp}^{\mathrm{prod}} y_{pt}^{\mathrm{PSP}_l} - \theta_l \quad (38)$$

subject to

$$\sum_{p \in P} t_{lp}^{\mathrm{prod}} x_{pt}^{\mathrm{PSP}_l} \le \tau, \qquad t \in T, \quad (39)$$

$$t_{lp}^{\mathrm{prod}} x_{pt}^{\mathrm{PSP}_l} \le \tau y_{pt}^{\mathrm{PSP}_l}, \qquad p \in P, t \in T, \quad (40)$$

$$\sum_{p \in P} y_{pt}^{\mathrm{PSP}_l} \le U, \qquad t \in T, \quad (41)$$

Constraints (39) models the capacity constraints, where t_{lp}^{prod} denotes the time needed for production line l to produce one unit of product p, and τ denotes the time period length. Constraint set (40) forces each $y_{pt}^{\mathrm{PSP}_l}$ variable to value one whenever the corresponding $x_{pt}^{\mathrm{PSP}_l}$ is greater than 0 to be able to model setup costs. Finally constraint set (41) says that no more than U different product types can be produced in any period. Also note that we subtract the convexity constraint dual variable θ_l from the objective function, and that the inventory balance constraints normally included in production scheduling problems, here belongs to the master problem which is controlled by the transport chain coordinator agent, and therefore, the production scheduling problems separate over time.

For each production fixation $f \in F_l^P$, we add

$$x_{p(f)t(f)}^{\mathrm{PSP}_l} \ge f \quad (42)$$

to the constraint set of subproblem PSP_l, which forces production line l to include a production of no less than f units of product $p(f)$ in period $t(f)$ in all subsequent plans.

5. COMPUTATIONAL EXPERIMENTS

We implemented our optimization model inside a multi-agent based simulation tool (TAPAS [4]) to be able to utilize a previously developed agent system. The software agent system in TAPAS is implemented in the Java Agent DEvelopment Framework (JADE) platform [1] and the optimization problems were solved using the ILOG CPLEX[2] 8.0 mixed integer linear problem solver.

Currently, TAPAS runs on a single computer but a distributed implementation (with different agents running on different computer) is possible. As discussed above, a distributed approach would increase the communication overhead, while potentially reducing the overall solution time because the availability of more processors. In our experimental implementation, JADE causes overhead time for coding and decoding of messages. We could however not perform any real experiments with a parallelization of the computations, since we do not have access to the necessary amount computers and ILOG CPLEX licenses.

Because of the slow convergence speed caused by the complexity of the studied problem, we found it unreasonable to solve RMP to optimality before fixating delivery, pickup and production quantities. Instead fixations are performed whenever the average relative improvement of a certain number of the latest generated plans fall below some required improvement level. Of course, this breaking criteria imposes a restriction on the solution space of RMP, but hopefully we end up with a heuristically good, integer solution to MP.

5.1 Scenario Description

We solved our problem on a set of 5 realistic scenarios using a transportation network consisting of two producer depots d_1 (with production line l_1) and d_7 (with production line l_2) and 6 customer depots $d_2, \ldots, d_6$ and d_8, and with a planning horizon of 72 time periods of uniform length 2 hours. The scenarios generated randomly from a set of parameters, and they differ only in the average customer demands, the minimum and maximum storage levels, and the storage levels at the beginning of the planning period. Customers $d_2, \ldots, d_6$ are VMI customer, while d_8 is a non VMI customer, which means that delivery times and delivery quantities are fixed for d_8, but not for the rest of the depots. Producer depot d_1 provides customer depots $d_2, \ldots, d_6$ with products and the purpose of d_7 and d_8 is to model possible non empty return transports for a small portion of the vehicles traveling back from $d_2, \ldots, d_6$ to d_1. The depot are connected with direct links, and from depot d_1 it is possible to travel to depots $d_2, \ldots, d_7$, from depots $d_2, \ldots, d_6$ all depots except depot d_8 can be reached, from d_7 it is possible to travel to d_1 and d_8, and from d_8, all transports must go to depot d_1.

We have a fleet $v_1, \ldots, v_9$ of 9 identical vehicles, where v_8 and v_9 represents third party transport capacity which can be called in with a transportation cost approximately 10% higher than the cost of $v_1, \ldots, v_7$. In our scenario, all vehicles are planned by the same transport planner, which means that we have one transport planner agent and two production planner agents (one for depot d_1 and one for depot d_7). Moreover, we have 4 different products $p_1, \ldots, p_4$, where p_4 is used to model return transports, and we assume p_4 is produced only in d_7 and consumed only in d_8. For storage in each node, we have a min storage level (safety) and a max storage level, and penalty costs for breaking these levels. Each node also has an initial storage levels, somewhere in between the minimum level and the maximum level, for each product.

Furthermore, we have expected forecasted consumptions (in units) per time period for each customer and each product. These forecasts are known to the coordinator agent, which also knows about the number of vehicles, products, production lines and depots, and the penalty costs for breaking the storage constraints. The coordinator does however not need to know anything about e.g. routes, transportation costs and production costs.

5.2 Results and Discussion

In each iteration, the coordinator agent sends one plan request message to each planner agent (in our scenario; one transport planner and two production planners), and each planner returns a response message to the coordinator. A response message can either contain a generated plan or some kind of failure notification indicating that no plan could be generated at the time of the request. For our scenario, which used an average of 3576 master problem iterations to reach the final solution, in average 17467 message had to be sent. The exact number of iterations and messages for each of the 5 scenarios is detailed in table 1. The number of messages can be used as a measure of the overhead in the system, compared to a non agent-based implementation approach. The size of each message depends on the actual application and the problem size of course, since a bigger problem e.g. means that more dual variables must be communicated.

Table 1: Number of master problem iterations and number of communicated messages for each of the 5 scenarios.

Scenario	1	2	3	4	5
Iterations	2239	3910	2375	3289	2740
Messages	13440	23466	14256	19734	16440

For each of the 5 scenarios, we estimated an upper bound of the overhead imposed by the agentification of our decomposition approach. This upper bound is calculated as the total running time minus the estimated time for performing decomposition related tasks. For the 5 scenarios, we measured an average total running time of 16576 seconds, and an average lower bound estimation of the time spent in the actual decomposition algorithm of 12583 seconds. This gives an average estimated overhead of approximately 27.1% of the total running time, with a standard deviation of 6.5 percentage units. Here the average estimated overhead is taken as the average over the overhead of the 5 scenarios.

For the 5 scenarios, we estimated the expected performance improvement that can be achieved when all subproblems are distributed to, and solved in parallel on, different computers. This is one of the purposes with the agent based decomposition approach, and for the scenarios, we got average estimated total solving times of 5734 seconds for the transportation subproblems and 23.6 seconds for the production scheduling subproblems. For each of the 5 scenarios, table 2 gives some interesting solving times and the estimated performance overhead imposed by the agentification.

[2]http://www.ilog.com/

Assuming all problems of the same type need the same solv-

Table 2: The total running time (Total), estimated decomposition algorithm time (Alg.), estimated times for solving transportation (Transp.) and production (Prod.) subproblems, and estimated overhead (Over.) for each of the 5 scenarios.

Scenario	1	2	3	4	5
Total (s)	10043	34449	11674	14785	11931
Alg. (s)	7032	29038	8412	10283	8149
Transp. (s)	5349	8077	4997	5403	4846
Prod. (s)	18	33	19	26	22
Over. (%)	30.0	15.7	27.9	30.4	31.7

ing time, we get average times of 0.22 and 0.0040 seconds respectively, for solving one transportation subproblem and one production scheduling subproblem. Here, the average transportation subproblem solve time is calculated as the average, taken over the 5 scenarios, of the quotient with the estimated total transportation subproblem solve time in the numerator and 9 (number of transportation subproblems) times the number of master problem iterations in the denominator. The average production subproblem solve time is calculated similarly, except for that the numerator is replaced with the total production subproblem solve time and denominator with 2 (number of production subproblems) times the number of iterations.

In theory, ignoring the increased communication time imposed by a parallelization, an average potential time reduction from $(5734 + 23.6) \approx 5758$ seconds to 637 seconds can be achieved. The reduced time for a scenario is calculated as the number of master problem iterations times the maximum taken of the average solve time of one transportation subproblem and the average solve time of one production subproblem. In our scenario, this would constitute a time reduction of in average $5758 - 637 = 5121$ seconds of the total running time. The standard deviation of this time reduction is 1188 seconds. Here, we further assume a usage of 12 computers, 9 for transportation subproblems, 2 for production scheduling subproblems and 1 for the master problem. We see that solving the production scheduling subproblems and the master problem on the same computer would give the same improved running time, but confidentiality of information would be weakened. Note that we cannot use the main algorithm as it is displayed in figure 3 to utilize a complete parallelization. Instead dual variables must be sent to the transportation planners and the production planners in parallel without resolving RMP in between. The actual optimal (or actually heuristic solution) of our scenario is however unimportant for the purpose of this paper since it is identical to the solution of the corresponding non agent-based algorithm, and therefore, we have chosen not to present any such data.

6. CONCLUSIONS AND FUTURE WORK

From our experiments, we see that it is possible to use a multi-agent system to implement a decomposition based optimization approach. There are some rather obvious positive effects of the approach, such as the possibility to provide increased confidentiality and distributed computing. However, there are some negative effects experienced in terms of processing and communication overheads.

Future work includes implementation of a traditional non agent based approach to allow us to perform an exact calculation of the performance overhead imposed by the agent-based implementation. Today the performance overhead is estimated by an approximated upper bound, which we indeed think is a decent indication of the overhead. Unfortunately, it is only possible to quantity the negative effects, in terms of messages and extra computing time, of an agent-based decomposition approach. Positive effects, e.g. in terms of increased confidentiality and robustness, have been discussed throughout this paper and we believe our approach has big potentials.

There are also potential positive effects connected to the different planners involved in the planning, in our case study, the production and transportation planners and the coordinator. Their tasks could be better supported by a distributed agent based decomposition approach than by a centralized approach. As mentioned, the confidentiality is one, another is the ability to locally specify and modify the problem. Further, the local agent could always provide the most recent subproblem solution to local planners (even though based on old dual variables). Also the local planner could manually intervene by developing (modifying) some solutions and send to the coordinator.

In order to capitalize on the use of the multi-agent based approach for achieving performance improvements, we find it interesting to experiment with a distributed system where each agent runs on its own computer, for example in a GRID system. Such a distributed approach is a natural model of the real world, where the actors (agents) are geographically separated and each agent typically only has access to its own data and its own computer. The estimations presented in section 5.2 indicates that the potential benefits from a parallelization of our approach is quite interesting. Also, a parallelization would allow us to solve more complicated subproblems thus making it possible to catch more real world details, which is important to be able to take the step from using the model in a laboratory environment to solving "real" problems. For this purpose we need to invest in more computers and for our optimization problems, we either have to use an open source mixed integer linear problem solver or buy more CPLEX licenses.

Furthermore, we plan to perform a tighter integration of our optimization ideas in TAPAS. TAPAS can, for instance, be used to evaluate the results returned by the "optimization" algorithm. An evaluation of the results can be useful because an optimization model in some meaning always constitutes an estimation of the real world. Also, real world evaluations of the results are typically hard to carry out since they tend to become rather expensive.

From a modeling perspective, there might be a need to improve how visit fixations are applied to the routing subproblems. The removal of the integer properties, as discussed in section 4.2.1, may cause too much performance overhead, which might need to be addressed. We will perform a computational analysis to measure the actual performance loss caused by this potential bottleneck.

7. ACKNOWLEDGMENTS

This research was primarily carried out within the projects "Integrated Production and Transportation Planning within

Food Industry"[3] and "Effects of Governmental Control Policies on Transportation Chains: A Micro-Level Study"[4] with fundings from the Swedish Knowledge Foundation and the Swedish Agency for Innovation Systems (VINNOVA). Further, the municipality of Karlshamn is in part financially supporting this research. We would like to thank our project partners, in particular AarhusKarlshamn and FoodTankers for their invaluable assistance in developing and verifying our optimization model.

8. REFERENCES

[1] F. L. Bellifemine, G. Caire, and D. Greenwood. *Developing Multi-Agent Systems with JADE (Wiley Series in Agent Technology)*. John Wiley & Sons, 2007.

[2] G. Dantzig and P. Wolfe. The decomposition algorithm for linear programs. *Econometrica*, 29:767–778, 1961.

[3] P. J. Daugherty, M. B. Myers, and C. W. Autry. Automatic replenishment programs: an empirical examination. *Journal of Business Logistics*, 20(2):63–82, 1999.

[4] P. Davidsson, J. Holmgren, J. A. Persson, and L. Ramstedt. Multi agent based simulation of transport chains. In *Proceedings of the 7th International Joint Conference on Autonomous agents and multiagent systems (AAMAS'08)*, Estoril, Portugal, 2008.

[5] P. Davidsson, J. A. Persson, and J. Holmgren. On the integration of agent-based and mathematical optimization techniques. In *Agents and Multi-Agent Systems: Technologies and Applications*, volume 4496 of *Lecture Notes in Artificial Intelligence*, pages 1–10. Springer, 2007.

[6] F. Fumero and C. Vercellis. Synchronized development of production, inventory, and distribution schedules. *Transportation Science*, 33(3):330–340, 1999.

[7] L. Lei, S. Liu, A. Ruszczynski, and S. Park. On the integrated production, inventory, and distribution routing problem. *IIE Transactions*, 38(11):955–970, 2006.

[8] L. A. Wolsey. *Integer Programming*. John Wiley & Sons, Ltd, New York, USA, 1998.

[3] http://www.ipd.bth.se/fatplan/
[4] http://www.ipd.bth.se/stem/