Association for Information Systems AIS Electronic Library (AISeL)

PACIS 2008 Proceedings

Pacific Asia Conference on Information Systems (PACIS)

July 2008

AN AGENT-BASED COOPERATIVE MECHANISM FOR INTEGRATED PRODUCTION AND TRANSPORTATION PLANNING

Yike Hu *Technical University Kaiserslautern, Germany,* hu@wiwi.uni-kl.de

Michael Schwind *Technical University Kaiserslautern, Germany,* schwind@wiwi.uni-kl.de

Oliver Wendt Technical University Kaiserslautern, Germany, wendt@wiwi.uni-kl.de

Follow this and additional works at: http://aisel.aisnet.org/pacis2008

Recommended Citation

Hu, Yike; Schwind, Michael; and Wendt, Oliver, "AN AGENT-BASED COOPERATIVE MECHANISM FOR INTEGRATED PRODUCTION AND TRANSPORTATION PLANNING" (2008). PACIS 2008 Proceedings. 238. http://aisel.aisnet.org/pacis2008/238

This material is brought to you by the Pacific Asia Conference on Information Systems (PACIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in PACIS 2008 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

AN AGENT-BASED COOPERATIVE MECHANISM FOR INTEGRATED PRODUCTION AND TRANSPORTATION PLANNING

Yike Hu, Michael Schwind, Oliver Wendt

Business Information Systems and Operations Research

Technical University Kaiserslautern, Erwin-Schrödinger Str., 67653 Kaiserslautern, Germany {hu, schwind, wendt}@wiwi.uni-kl.de

Abstract

This paper presents a decentralized cooperative economic scheduling mechanism for a supply chain environment. For this purpose, we design autonomous agents that minimize the production or transportation and outsourcing costs incurred by the external execution of a task. The decentralized cooperative scheduling approach comprises two parts: the individual optimization of an agent's local schedule and the cooperative contract optimization, either by outsourcing the task or by (re-)contracting the release time and due time with the contract partners aiming to maximize their total profits. A negotiation mechanism based on trust accounts is employed to protect the agents against systematic exploitation by their partners.

Keywords: decentralized scheduling, cooperative agents, production and transportation integration, supply chain networks

1 INTRODUCTION

Approaches to cooperative management of inter-organizational supply chains have attracted a great deal of attention in recent years. But their focus on synchronizing supply and demand in order to collaboratively reduce stock and idle time of production capacity has aggravated the problem of low truck loading rather than contributed to sustainable logistics.

In short-term production scheduling cooperative approaches have not yet become standard, at least not in the sense that the opportunity cost of the logistics services concerned (whose preferred pickup and delivery times might vary) would be accounted for in the production scheduling process. Production and logistics scheduling tend to be planned successively: Due dates of the supplier are synchronized with prospective release times of the customer and then provided as exogenous data to the logistics planner (or external service provider), whose task is to meet the given pick-up and delivery times and not to renegotiate them. Such negotiation, although it would be crucial for the simultaneous optimization of the complete supply chain, comprised by production and logistics, is found hardly anywhere, neither in practice nor in literature.

Since real-life supply chain management (SCM) is closely related to problems caused by the divergent interests of the actors (enterprises) and the distributed structure of the underlying optimization (scheduling), one natural way to address this constellation is to employ a holonic multi-agent system (MAS), in which the process structure is mapped onto single agents (Eymann 2001). Each agent - operating a single production facility or vehicle fleet - maximizes its own profit by determining an optimal internal schedule. To streamline the production and transportation process and avoid penalty

costs due to processing bottlenecks or late delivery, agents carry out economic scheduling. Accordingly, they employ 'outsourcing contracts' to reduce their production or transportation load and optimize their schedules. The negotiation has to be seen as an inter-agent schedule optimization process, leading to the social contract equilibrium of the MAS based on prices calculated according to the production and transportation load. Due to the fact that the calculation of these negotiated prices cannot be directly monitored by the contract partners, a trust protocol has been included to foster truthful bidding. Moreover, by implementing an incentive compatible mechanism to avoid exploitation by competitors, our decentralized supply chain optimization system is able to deal with other agency problems like the unwillingness of actors to reveal sensitive but (in terms of system optimization) valuable information. We address the revelation issue by introducing a trust account mechanism, which helps to prevent individual long-term exploitation. The trust mechanism is directly integrated into the schedule optimization procedure, similar to Padovan et al. (2002).

In this paper we address the question of how logistics costs may be reduced by exploiting scheduling flexibilities of the supply chain partners. For this purpose we extend the "DISPOsitive WEB" protocol of Stockheim et al. (2005). Whereas the DISPOWEB protocol described by Stockheim et al. focuses solely on the negotiation of Pareto-efficient delivery contracts for a production plan between production agents, while avoiding the exploitation of negotiation partners by the use of trust accounts, this work also integrates the negotiation of transportation planning agents into the process. We think that, by modelling our optimization problem as an integrated agent-based planning process, we are able to guarantee a more appropriate mapping of real world situations in SCM (Caridi et al. 2004) and for this reason the DISPOWEB mechanism is able to provide a higher solution quality for our scenario than traditional centralized operations methods can. However, we have to prove this by testing our MAS model with real world data, as will be described in the last section.

The paper is organized as follows: The next section summarizes the literature relevant to our research, followed by the formal description of the *agent-based integrated production and distribution planning problem (ABIPDP)*. Section four presents the negotiation mechanism that is used to find feasible solutions for the ABIPDP. The last section summarises the simulation experiments that will be conducted in order to evaluate the quality of the ABIPDP model we propose here.

2 LITERATURE REVIEW

A primary goal of our model is to integrate the planning of production and distribution processes with a very flexible formulation of optimization approaches in order to provide maximum adaptability in our approach to real world scenarios. To our knowledge, there are no satisfactory solutions to be found in literature on *integrated production and distribution planning*, so we will therefore be brief. Our approach also integrates decentralized scheduling; because we are especially interested in *economic variants of scheduling* we will discuss this in the second part of our review. Finally we touch the vehicle routing problem, because it will play a crucial role in the further development of the DISPOWEB model.

2.1 Integrated Production and Distribution Planning

In the recent years the seamless *integration of production and distribution planning* has become a focus of operations research. However, due to the complexity of the combined problem there are few papers that give a reasonable answer to how to couple both optimization problems.

Sequential integration approaches solve the integrated production and distribution planning by a two phase process: the optimization results of the production planning are used as an input for the optimization of the distribution process. *Centralized planning* is chosen by Ertogral et al. (1998) who combine a pickup-and-delivery model with time windows (Li & Lim, 2002), with a multi-level multi-item dynamic capacitated lot sizing problem (Stadtler, 2003). By employing a Lagrange decomposition approach, they are able to find solutions for the combined problems. *Decentralized*

planning is chosen by Jung & Jeong (2005), who introduce an agent-based production and distribution planning system which generates feasible operational plans based on the negotiation process between production and distribution agents that incorporate individual objective functions for the each optimization goal in a decentralized supply chain structure. The focus of this approach is mainly on sequentially determining the optimal lot size in a multi-period model. It does not integrate route planning. Lau et al. (2006) also present a decentralized, agent-based supply chain scheduling model that employs the contract net protocol (Davis & Smith, 1983) in a distributed negotiation process to construct cost minimal production plans with respect to an objective function that integrates geographically dependent transportation costs, production costs, and costs for lateness.

Simultaneous integration approaches are taken by Boudia et al. (2005), who design a centralized production and distribution planning system that has the objective of periodically calculating the appropriate production quantity and the delivery routes associated with the product dispatching, in order to minimize the total cost of production and distribution. The resulting combined problem formulation is solved using a greedy heuristic combined with a local search. Eksioglu (2002) examines an integrated production and distribution planning problem in a two-stage supply chain consisting of a number of facilities, all capable of producing the final product, and a number of retailers. Their model assumes that the retailers' demands are known deterministically and there are no production or transportation capacity constraints. After formulating the problem as a network flow model with fixed charge costs, they solve the NP-hard problem using a primal-dual based heuristic. Lei et al. (2006) present an approach that is closely related to the previous one and operates on the production, inventory, and distribution routing problem. They are able to solve the combined production and distribution planning problem by a two phase process using mixed-integer programming and a heuristic algorithm approach. Sarimento & Nagi (1999) give an extensive analysis and categorization of integrated production and distribution planning problems that pursue the simultaneous simulation approach. It provides a comprehensive introduction to the problem class we are considering in our work.

2.2 Weighted Job Interval Scheduling Problem and Economic Scheduling

The Weighted Job Interval Scheduling Problem (WJISP) was introduced by Garey & Johnson (1977) and Potts & Van Wassenhove (1988). It is defined as the scheduling of a set of jobs, subject to *release dates* and *due dates* on a single machine, which can process at most one job at a time. The objective is to minimize the weighted number of late jobs, or equivalently to maximize the weighted number of early jobs (Koulamas & Kyparisis, 2001). This problem is shown to be NP-hard if the tasks are assumed not to be preemptive (Gordon & Kubiak, 1998). Meta-heuristics like genetic algorithms (GA) tabu search (TS) and simulated annealing (SA) are often used to solve the job-shop scheduling problem (JSSP) (Vaessens et al., 1996; Yamada & Nakano, 1996). They can easily be adapted for the WJISP.

2.3 Capacitated Vehicle Routing and Scheduling Problem

The quantitative exploration of the vehicle routing problem (VRP) was introduced by Dantzig et al. (1959). The VRP has prime place in logistics management and belongs to the class of NP-hard problems too. In practice the precedence relations between customers are often pre-determined because of exact service time or service time windows given by the customers. This kind of VRP with time restrictions is termed Vehicle Routing and Scheduling Problem (VRSP) cp. e.g. Desrochers et al. (1990) and has also been addressed by meta-heuristics as for example in Gambardella et al. (1999).

3 DESCRIPTION AND FORMALIZATION OF THE PROBLEM

In general, the production facilities of the different tiers (and the end customers) of a supply chain or network are distributed geographically. Obviously, the due time of a producer is corresponds to the earliest pickup time for producers on subsequent tiers, and the release time of a producer or an end customer corresponds to the latest delivery time of the supplier. The pickup and delivery time intervals are defined by the contracts between the production and transport agents. In fact, the end customers in a supply network can be modelled as "degenerated" production agents, which have no customers. Thus, in the following we only consider two different agent types in the supply chain network, namely production agents (PA) and transport agents (TA).

Although there is an abundance of production scheduling problems with various assumptions about the available resources and the order in which tasks may be processed by single resources or by bundles of them, we have chosen the classic Job Shop Scheduling Problem (JSSP) to illustrate our approach. Note that this is for illustrative ease only and the inter-agent negotiation would not change if we extended this to resource-constrained (multi-)project scheduling or any other production model with a bundle of fixed resources, release times and due dates for tasks and outsourcing opportunities: The JSSP is defined by several jobs and each of these jobs is closed by several tasks that have to be sequentially executed in different shops (seen as resources or bundles of resources able to process only a single task at a time). We identify each production agent with a shop. Furthermore, we extend the JSSP by introducing a valuation function that attributes constant economic weights to each of the single tasks in a job, instead of a valuation function for the jobs combined or the complete schedule. This way, for each production agent we obtain a weighted single machine scheduling problem. Figure 1 depicts how the weighted JSSP can be broken down into (highly interdependent) subproblems, in which each production agent is exclusively concerned with the execution of its set of tasks. The resulting subproblems of the WJSSP are instances of the Weighted Job Interval Scheduling Problems (WJISP) as defined by Elendner (2003). The intervals of a PA's WJISP are in accordance with the time interval between the supplier's delivery time for the required preliminary products (or raw materials) from the PA's predecessor, and the pickup time of the products provided to the PA's successor. The execution time of production tasks is up to the PA but is restricted by the supply and delivery contracts. In contrast to the standard JSSP, now the goal is to minimize not so much the make span as the overall cost of outsourcing (as defined by the tasks' weights, if they cannot be scheduled within their interval) while meeting all contract times.



Figure 1: Decomposition of the Weighted JSSP into WJISPs.

Similarly, we introduce constant weights for the single pickup and delivery tasks of a vehicle's transportation tour. Each transport agent (TA) represents a capacitated vehicle and has to plan its optimal route for executing the pickup and delivery tasks. The individual pickup and delivery time at the PAs can be negotiated between the TAs and their contracting PAs. In analogy to the WJISP we introduce a weight for each transportation task and allow for outsourcing them by incurring that cost whenever the time window cannot be met. We label this "economic" extension of the CVRSP the Weighted Pickup and Delivery Interval Routing and Scheduling Problem (WPDIRSP). For the transport agents the optimization task is more challenging since vehicle routing decisions always involve sequence dependent setup times in their scheduling decisions.

The negotiation of pickup and delivery times in our WJISP and WPDIRSP framework results either in a feasible situation without changing the cost structure or in an infeasible situation, i.e. one in which it would not be possible to execute all tasks in the given intervals. For this case we introduced an implicit outsourcing option which enables the agents to negotiate the external execution of a job in return for a payment. The following section summarizes the assumptions of our model.

3.1 Assumptions of Production and Transportation Model

Each job and tour respectively is composed of a number of subtasks to be executed sequentially (other inputs are not critical): Our agent-based WJISP adopted the assumptions of the classical JSSP, besides assuming that each task has a set of predecessors. The relaxation requires the WJISP's release times to be defined as the maximum of all contract times over all suppliers. In our WPDIRSP-Model each contract defines either a pick-up task or delivery task. But, as a producer or consumer several different contracts can be executed sequentially.

Closed model with deterministic jobs: We assume every task to be known to the agents, i.e. they only renegotiate existing contracts. Extending to a dynamic model with emerging tasks is straightforward, however: Whenever a new service or product request has to be priced, the agent's WJISP and WPDIRSP defined by its current portfolio of *2n* contracts is extended by two additional contracts.

Existence of primary contract between producer and customer: Long-term relationships between production facilities are assured by primary contract which we do not have to deal with in our model.

Pickup and delivery: Given the external demand for the products over a discrete time horizon, the pickup time at producer and delivery time at customer are negotiable by contracting between PA and TA. The negotiation mechanism applied by two agent types is identical. For the transport agents, the assumptions of our Model are based on that of the CVRP, but a heterogeneous vehicle fleet without stores is taken into consideration. To execute each corresponding pickup and delivery task the TA must pick up the products at first and then deliver them (precedence relations).

Static costs for task outsourcing: Currently we assume an outsourcing option to be available for all tasks and every agent at all times, meeting whatever deadline the customer will require. While this simplification reduces the problem's complexity, it is true that in most economies for almost any product or service a substitute will be available at any time for a price below infinity.

Unlimited compensation budgets: We assume each agent to have unlimited financial resources for side payments. Since the side payments only serve to compensate for economic value generated by relaxing the agent's scheduling constraints or to collaboratively escape suboptimal plans, the sheer number of re-contracting steps applied keeps the probability of persistent financial loss very low.

3.2 Agents' Knowledge

Denote by $A := \{A_p \cup A_T\}$ the set of all agents consisting of the set of PAs A_P and the set of TAs A_T . Define $T := \{0, 1, 2, ..., \overline{T}\}$ to be the discrete planning horizon of our model. Each agent $a \in A$ disposes of the common knowledge, namely A and T.

3.2.1 Knowledge of Production Agents

Contracts: PAs have two types of contract with their suppliers.

Each supply contract C_k^{ij} of $i \in A_p$ with $k \in A_T$ defines

- $d_j \in :$ *Demand* for items (raw materials, intermediate products or final products) that must be produced by *i* and picked up by *k* in order to be delivered to *j*.
- $T_{pd}(C_k^{ij})$ $(0 < T_{pd}(C_k^{ij}) \le \overline{T})$: Production due time of *i* (equiv. to the pickup time of *k* at *i*).

Note that in a WJISP for a supply contract C_k^{ij} of *i* with *k*, exactly one production task is to be done by *i*. Let ${}^s := \bigcup_{k \in A_T} C_k^{ij}$ be the set of supply contracts of *i* with all $k \in A_T$, and Γ be the set of production

tasks γ of *i*. Each task γ is defined by exactly one single contract $C_k^{ij} \in s^{ij}$ of *i*. That is $|\Gamma| = |s|^{ij}$.

- $p_{\gamma} \in$: *Processing duration* of task γ .
- $w_{\gamma} \in : Outsourcing \ costs \ of \ task \ \gamma \ of \ contract \ C_k^{ij}$.

We suppose that for execution of task γ a set of input items are required and delivered by transport agent $l \in A_{\tau}$. Then, each *delivery contract* C_{l}^{hi} between *i* and *l* specifies

- $d_i \in :$ Demand of *i* for input items that must be delivered from $h \in A_p$ to *i* by *l*.
- $T_{dd}(C_l^{hi})$ $(0 < T_{dd}(C_l^{hi}) < T_{pd}(C_k^{ij}))$: Desired *delivery due time* of l at i.
- $C^d := \bigcup_{\substack{k \in A_r}} C_k^{hi}$ defines the set of delivery contracts of *i* with *k*.

The sequence dependent setup time between task β and γ is denoted by $\mathcal{G}_{\beta\gamma} \in \mathcal{G}$.

Resource capacity: In WJISP the resource capacity $cap_i \in of a_i$ amounts to 1 unit. The demand $r_{i\gamma}$ of the resource by task γ is either 0 or 1, because each task requires exactly one resource if executed.

3.2.2 Knowledge of Transport Agents

TAs have the following contract pairs:

A pickup contract C_i^{ij} of $k \in A_T$ with $i \in A_p$ consists of

- $-d^{C_i^{ij}} \in (i.e. d_i \in \mathbb{Z}_{\leq 0})$: *Demand* for items that must be picked up by *i* then delivered to *i*'s customer $j \in A_n$.
- $T_{pick}(C_i^{ij})$ $(0 < T_{pick}(C_i^{ij}) \le \overline{T})$: *Pickup time* of k at *i* (equiv. of production due time of *i*).

Symmetrically, k has a *delivery contract* C_i^{ij} which is closed with j. C_i^{ij} defines

- $d^{C_j^{ij}} \in : demand$ of j for items produced and picked up at prod. i, then delivered to cust. j.
- $T_{deliv}(C_j^{ij})$ $(T_{pick}(C_i^{ij}) << T_{deliv}(C_j^{ij}) \le \overline{T})$ indicates the delivery due time of k at j.

Let C^i represent a pickup or delivery contract of the transport agent with any production agent *i* which either producing or receiving the products specified by contract C^i , i.e. $C^i \in \{C_i^{ij}, C_i^{ki}\}$ $(i, j, k \in A_p \ i \neq j, i \neq k.$). $prod(C^i)$ and $cust(C^i)$ declare the identity of *i* appropriate to C^i . Thus, *i* is either a producer or customer for the transported products. For example, we number the production agents serially. $C_1^{13}, C_1^{14}, C_3^{13}, C_4^{14}$ are contracts between $k \in A_T$ and PA 1, 3, 4, where PA 3, 4 are customers of PA 1. These contracts indicate that the items are to be picked up at PA 1 and then delivered to PA 3, 4. Therefore, $C^1 \in \{C_1^{13}, C_1^{14}\}, C^3 \in \{C_3^{13}\}, C^4 \in \{C_4^{14}\}, prod(C_1^{13}) = prod(C_1^{14}) =$ $prod(C_3^{13}) = prod(C_4^{14}) = 1$, $cust(C_1^{13}) = cust(C_3^{13}) = 3$, etc.

Furthermore, a contract C^i has to comprise:

- $p_i^{C^i} \in :$ Processing (service) duration of a pickup or delivery task at *i* according to C^i .
- $W_{i,c^i} \in :$ Outsourcing costs of contract C^i with producer or customer *i* specified by C^i .

Additional information the TAs have available is as follows: $\tau_{ij} \in$ defines the travel time from *i* to *j*, $c_{ii} \in$ the appropriate travel costs. The maximum load capacity of *k* is limited to $cap_k \in$.

Summarized, the Relationships between production and transport agents in our supply chain can be shown as follows: $\dots h_{\epsilon A_p} \leftarrow \frac{C_l^{hi}}{C_h^{hi}} > l_{\epsilon A_p} \leftarrow \frac{C_l^{hi}}{C_i^{hi}} > i_{\epsilon A_p} \leftarrow \frac{C_k^{ij}}{C_i^{ij}} > k_{\epsilon A_p} \leftarrow \frac{C_k^{ij}}{C_j^{ij}} > j_{\epsilon A_p} \cdots$ It presents the relationships defined by contracts between PAs *h*, *i*, *j* and TAs *l*, *k* (cp. Figure 2). The notations above the arrows denote the delivery and supply contracts of PA with TAs, and notations under the arrows describe the pickup and delivery contracts of TAs with PAs.

3.3 Agents' Costs and Schedule Optimization

The common objective of each agent is to minimize the total operating and outsourcing costs of its tasks by determining an optimal production schedule or a transportation route and schedule. The agent's internal scheduling optimization mechanism is described below.

3.3.1 Internal Optimization of Production Agents

The PA's production planning consists of determining a single-machine schedule which minimizes the internal production and outsourcing costs, satisfies the external demands for intermediate or final products, fulfils the capacity constraints of resources and complies to the agreed production due time and delivery time of the supplier respectively. We suppose the production costs caused by processing the tasks on the shop to be constant. Then, only the delivery contracts and the outsourcing costs of them are relevant to the objective function.

We assume that production agent *i* has $n \ge 1$ delivery contracts or production tasks to plan for. For production start and completion we introduce the dummy production tasks 0 and n+1. That is $p_0 = p_{n+1} = 0$. Let $S_{\gamma} \ge 0$ denote the completion time of task $\gamma \in \Gamma(\Gamma = \{0, 1, ..., n+1\})$ and $S_0 := 0$, S_{n+1} the end time of production. Task γ can be dispatched if all input items required by processing the task γ have been delivered. That means the release time of task γ is $rt_{\gamma} := \max \{T_{dd}(C) \mid C \in d\}$. The estimated production due time dt_i ($rt_i < dt_i \le \overline{T}$) is given by a negotiation process. Let $y_{\gamma} \in \{0,1\}$ be a decision variable for task γ and $x_{\beta\gamma} \in \{0,1\}$ be a sequence assignment between tasks β and γ . y_{γ} equals one exactly if task γ is completed. $x_{\beta\gamma}$ equals one exactly if task β is processed immediately before γ . The WJISP of each production agent can be formulated as follows, with an objective function of outsourcing costs minimization.

$$Min. \quad \sum_{\gamma \in \Gamma \setminus \{0,n+1\}} w_{\gamma} (1 - y_{\gamma}) \tag{1}$$

Subject to

$S_{\gamma} - S_{\beta} \ge \mathcal{P}_{\beta\gamma} + p_{\gamma} - \overline{T} (1 - x_{\beta\gamma})$	() $\forall \gamma, \beta \in \Gamma$	(2)
$S_0 = 0$		(3)
$(rt_{\gamma} + p_{\gamma})y_{\gamma} \le S_{\gamma} \le dt_{\gamma}$	$\forall \gamma \in \Gamma$	(4)
$S_{\gamma} \ge y_{\gamma}$ and $S_{\gamma} \le \overline{T} \cdot y_{\gamma}$	$\forall \gamma \in \Gamma$	(5)
$\sum x_{\beta\gamma} = y_{\gamma}$	$\forall \gamma \in \Gamma \setminus \{0\}$	(6)
$\sum_{\gamma \in \Gamma \setminus \{0\}}^{\beta \in \Gamma \setminus \{n+1\}, \beta \neq \gamma} x_{\beta \gamma} = y_{\beta}$	$\forall \beta \in \Gamma \setminus \{n+1\}$	(7)
$x_{\alpha}, y_{\alpha} \in \{0,1\}, y_{\alpha} = y_{\alpha} = 1$		
$x_{\gamma\gamma} = x_{\gamma 0} = x_{n+1,\gamma} = 0$	$\forall \gamma \in \Gamma$	

Constraint (2) says if task γ is executed immediately after β then the difference between the end of two processes can not be less than the minimal time lag $g_{\beta\gamma} + p_{\gamma}$. Time constraint (4) indicates that if task γ is released and processed, i.e. $y_{\gamma}=1$, then the completion time S_{γ} must comply with the time windows given by $[rt_{\gamma} + p_{\gamma}, dt_{\gamma}]$. S_{γ} and y_{γ} become interdependent because of constraint (4) which says: if task γ is released and completed, i.e. $S_{\gamma} > 0$ then y_{γ} must equal one; if y_{γ} equals zero then $S_{\gamma}=0$. Tasks are sequentially processed, meaning each task γ has exactly one predecessor and one successor. This condition is enforced by restrictions (6)-(7).

3.3.2 Internal Optimization of Transport Agents

 $\begin{aligned} x_{ij}^{C^{l}C^{j}}, y_{iC^{i}} \in \{0,1\}, y_{0C^{0}} = y_{n+1,C^{n+1}} = 1\\ ld_{i}^{C^{i}} - d^{C^{i}} \ge ld_{j}^{C^{j}} - cap(1 - x_{ij}^{C^{l}C^{j}}) \end{aligned}$

The formulation of the scheduling optimization model of transport agents is more complex. Note that for the demand $-d^{C_i^{ij}} \ge 0$ of producer *i* defined by C_i^{ij} (with $i=prod(C_i^{ij})$, $j=cust(C_i^{ij})$) and the demand $d^{C_j^{ij}} \ge 0$ of customer *j* defined by C_j^{ij} (with $i = prod(C_j^{ij})$, $j = cust(C_j^{ij})$) it holds that $-d_i = d_j$, i.e. all the products are to be picked up and delivered exactly once. Analogous to the WJISP, each task or contract in a WPDIRSP which minimizes the travel costs and outsourcing costs is also to be dispatched and carried out only once. Each vehicle must start and end its tour at the depot. In addition, we require all items being picked up to be delivered, in other words, if pickup contract C_i^{ij} has to be fulfilled then the corresponding delivery contract C_i^{ij} must be executed, too. C^0 and C^{n+1} with $p_0^{C^0} = p_{n+1}^{C^{n+1}} = 0$ are fictitious contracts between each transport agent and depot 0 and n+1, which represent the start and end points of transport services, respectively. Given $n \ge 1$ pickup and delivery contracts with producer and customer, let $V := \{0, 1, ..., n+1\}$ be the set of locations to be travelled to, $:= \{C^0, C^{n+1}\} \cup \{C^1, \dots, C^n\}$ denote the set of transport contracts. and let

$$Min. \qquad \sum_{i} \sum_{C^{i}} \sum_{j} \sum_{C^{j}} x_{ij}^{C^{i}C^{j}} \cdot c_{ij} + \sum_{C^{i} \in i \in prod(C^{i}), cust(C^{i})\}} W_{iC^{i}}(1 - y_{iC^{i}})$$
(8)
Subject to

$$\begin{cases} C^{i} - S_{i}^{C^{i}} \ge \tau_{ij} + p_{j}^{C^{j}} - \overline{T}(1 - x_{ij}^{C^{i}C^{j}}) & \forall i, j \in V; C^{i}, C \in \\ \end{cases}$$
(9)

$$(T_{pick}(C^{i}) + p_{i}^{C^{i}})y_{ic^{i}} \leq S_{i}^{C^{i}} \leq T_{deliv}(C^{i}) \qquad \forall i \in V, C^{i} \in (11)$$

$$(10) \qquad (10) \qquad (11) \qquad (11)$$

$$S_{i}^{C^{i}} \ge y_{iC^{i}} \quad \text{and} \quad S_{i}^{C^{i}} \le \overline{T} \cdot y_{iC^{i}} \qquad \forall i \in V, C^{i} \in$$

$$\sum_{i \in V, C^{i}} \sum_{i \in V, C^{i} \in V$$

$$\sum_{\substack{C^{i} \in \ \backslash C^{n+1}, C^{i} \neq C^{j} \ i \in \{prod(C^{i}), cust(C^{i})\} \\ \sum_{\substack{C^{j} \in \ \backslash C^{0}, C^{i} \neq C^{j} \ j \in \{prod(C^{i}), cust(C^{i})\} \\ x_{ij}^{CC} = x_{0j}^{C^{0}C} = x_{i,n+1}^{C^{n+1}} = 0 \\ x_{k0}^{CC} = x_{n+1,k}^{C^{n+1}C^{k}} = x_{kk}^{C^{k}C^{k}} = 0 \\ \forall C^{i} \in \ \backslash \{C^{n+1}\}, i \in \{prod(C^{i}), cust(C^{i})\} \\ \forall C^{i} \in \ \backslash \{C^{n+1}\}, i \in \{prod(C^{i}), cust(C^{i})\} \\ \forall C^{i} \in \ \langle C^{n+1}\}, i \in \{prod(C^{i}), cust(C^{i})\} \\ (14)$$

$$\forall C \in , i = prod(C), j = cust(C)$$
(15)

$$\forall k \in V; C^k, C \in \tag{16}$$

$$\forall i, j \in V; C^i, C^j \in , C^i \neq C^j$$
(18)

$$\forall i \in V; C' \in \tag{19}$$

$$\begin{aligned} & la_i - d \ge la_j - cap(1 - x_{ij}) & \forall i, j \in V; C, C \in C, C \neq C \end{aligned}$$

$$\begin{aligned} & d_i^{C^i} \le ld_i^{C^i} \le cap & \forall i \in V; C^i \in C \end{aligned}$$

$$\begin{aligned} & V_i \in V; C^i \in C \in C \\ & (19) \end{aligned}$$

$$\begin{aligned} & \sum_{C_j^{ij} \in C} d^{C_j^{ij}} \cdot y_{jC_j^{ij}} \le -\sum_{C_i^{ij} \in C} d^{C_i^{ij}} & \forall i \in V; C^i \in C \end{aligned}$$

$$\begin{aligned} & \forall i \in V; C^i \in C \\ & \forall i \in V; C^i \in C \\ & (19) \end{aligned}$$

The formulation of WPDIRSP of transport agents as shown above with decision variables $x_{ii}^{C^{i}C^{j}}$, $y_{iC^{i}} \in \{0,1\}$, $S_{i}^{C^{i}} \ge 0$ ($S_{0}^{C^{0}} = 0$) and integer auxiliary variable $ld_{i}^{C^{i}}$ representing the load of a vehicle at the arrival time at *i* in order to execute contract C^i . $x_{ij}^{C^iC^j}$ defines the sequence of transport services to execute contract C^i at *i* and C^j at *j*. $x_{ij}^{C^iC^j}$ takes the value one exactly if *i* is served to complete contract C^{i} immediately before j to carry out C^{j} . $y_{iC^{i}}$ is equal to one exactly if i is visited to execute C^{i} . $S_{i}^{C^{i}}$ defines the completion time of C^{i} at *i*.

Condition (9) enforces the minimal time lags between the ends of two different service tasks which are carried out sequentially. Restriction (10) gives the service time windows of contracts in the same way as restriction (4) in the WJISP-Model. (11) is equivalent to (5). Inequations (12)-(17) ensure that each contract C^i to be completed at *i*, has exactly one predecessor and successor respectively, if C^i is carried out. Moreover, each C^i will be dispatched and executed at most once. The customer *j* indicated by contract C_j^{ij} cannot be visited before the producer *i* of C_j^{ij} . The travel tour must be started at depot 0 and closed at n+1, i.e. C^0, C^{n+1} must be performed. Constraints (18)-(19) guarantee that the maximum load capacity *cap* of vehicle will not be exceeded during the total service time at any producer or customer. The equality of the product quantity which is offered by producer *i* and required by his customers can be assured by formula (20).

4 NEGOTIATION MECHANISMS

In addition to the optimization of the PAs' and TAs' internal schedules, agents can try to relax their problems by increasing their temporal flexibility in order to maximize the utility of it's individual contract situation, i.e. renegotiating for later contract dates with the subsequent tiers of the supply chain or earlier dates with the suppliers or delivering TAs in return for payment. However, since this relaxation or the agent's own problem leads to a more constrained problem for the contracting agent, compensation payments will have to be negotiated to find an appropriate trade-off.

4.1 Agents' Cooperative Contract Optimization

Let us assume the consumer, i.e. the supply web's last tier, is willing to pay for earlier deliveries. A total schedule with less idle time will thus result in earlier delivery times to the consumer and may generate additional profit that can be distributed among the supply web's PAs and TAs.

Each Agent has a schedule- (i.e. time-)dependent price function U^* , which maximizes the utility of contract situation. U^* is defined as follows: $U^*(C \subset) = \max_{S \in S} U(C, S)$ with the subset C of contracts

set C and the corresponding optimal schedule $S=(S_0,...,S_{n+1})$ from the set S of schedules.

For every given schedule *S*, any PA or TA agent can calculate the opportunity cost incurred or additional benefit gained when moving the contract time (by defining either the PAs due date and TAs earliest pickup time, or the TAs latest delivery time and the PAs release time, as being equal). Since widening of an interval always leads to a relaxed WJISP or WPDIRSP these new problems always have the same or lower cost, while narrowing the time interval always generates a WJISP or WPDIRSP with the same or higher cost.



Figure 2: Time-dependent cost functions serving as a basis for re-contracting

Let us illustrate this from the transport agents point of view (i.e. there is transportation at zero cost and thus PAs negotiate directly) and consider PA b in figure 2 (left): Re-contracting the release time of job 1 (blue) from t = 20 to t = 23 or even later would render the scheduling of this task impossible. On the other hand, the resources freed by this would allow for the scheduling of the second task in an optimal solution which would lead to a total cost of 15 instead of 14 MU, i.e. a cost increase of 1 MU. Relaxing the contract to t = 18 or earlier would allow the scheduling of both the first and the second task, thus yielding an additional profit of 6. For agent 1 a contract time of 17 or earlier renders job 1 (the upper one) impossible but in turn allows for scheduling the third task (third from top) causing a

total cost increase of 6 MU. When relaxing the deadline to (at least) 21 however, all tasks can be scheduled by agent 1 (starting with the third one).

Assuming both agents had agreed on a price x for t = 20, adding the respective cost deltas indicated in figure 2 (left) would define time-dependent price functions representing the agents' opportunity cost or benefits. By communicating this function to the partner, each agent could calculate an optimal recontracting step. In our case, agreeing that t = 18 would lead to a total surplus of 6 MU that could be shared by the two agents.

When calculating the price functions showing the optimal contract time for a task, we have to assume all other contracts are kept constant. This in turn means the re-contracting operation for task 1 now leads to outdated price functions for all other contracts of agent 1 and agent 3, i.e. updating would be required to determine whether their contracting time (i.e. the schedule dependent on it) is still optimal. This raises the question of whether it really is efficient to have the agent calculate the price functions for all points in time before communicating them, especially when the whole system of interdependent negotiations is still far from any equilibrium.

Having this "social contract optimum" calculated by a central planner run into two problems: First, the agents lack any incentive to reveal the level of their internal knowledge to this central planner, and second, solving the global mixed integer optimization problem becomes prohibitive when we don't want to consider just toy-size supply networks: such a toy example with 5 PAs and one single TA has already made ILOG CPLEX run for 1.13 seconds, and generate 976 nodes in 14024 iterations.

In DISPOWEB we use a "memory-free" alternative, randomly choosing a time offset and then proposing this shift to the contracting party, making both agents estimate the implications of this one specific change only (by solving their respectively modified WJISPs or WPDIRSPs). Although this comes with the disadvantage of not finding the bilaterally optimal contract time for a given contact in one search step, it drastically reduces the number of WJISPs and WPDIRSPs to be solved by each agent.

If the agents agree on a new delivery time (e.g. t = 18 in our scenario) and one agent profits more from the new delivery time than the other agent's additional cost, the total profit is assumed to be shared equally. In the example: agent 3 makes a side payment of 3 MU to agent 1, leading to a situation where both agents profit from the change of the delivery contract.

The extension of the model by logistics agents is now straightforward: Since PAs and TAs have the same "external interfaces" in the sense that they only re-negotiate time points of existing contracts, the interaction protocol does not have to depend on the agent type at all! Thus, between every two PAs we can simply "plug in" the according TA as though it were just an intermediate PA of the supply chain. Whether it has to solve a WPDIRSP internally instead of a WJISP (or any other economic scheduling problem with time windows) does not matter. What does matter, however, is the monotonicity of the scheduling problem's function with respect to the time windows. Imagine all transport from PA a to PA b has been contracted to the same transport agent (TA 1), as depicted in figure 2 (right): If the trip from PA a to PA b takes 4 hours one way, the TA would have to outsource the second transport in order to be able to meet all his contracts if we assume all transports to require the full capacity of the truck. Be renegotiating the pickup time for the first task to be 2h earlier and the delivery time for the third to be 3h later, the logistics agent could save these outsourcing costs without incurring any additional expenses, either for PA a or for PA b.

4.2 Altruistic Negotiation and Avoiding Exploitation

When evaluating this decentralized economic scheduling mechanism for a pure production scheduling network without logistics (Stockheim et al., 2005), we realized that this mechanism, when employed on heuristically generated initial schedules and contracts, would increase welfare by repeated renegotiation, but due to the strong interdependence of the schedules this process converges into sub-optimal equilibrium sets of contracts. Although far from providing globally optimal schedules, no pair of agents has the opportunity to benefit from bilateral re-contracting.

A modification of the re-contracting mechanism, extending the idea of simulated annealing to a collaborative optimization process, significantly improves the contracting equilibria reached: Instead of the "selfish" negotiation in which agents only accept those transition proposals which lead to a profit increase for the accepting agent (and, of course, for the proposing agent, too, otherwise it would never propose the contract modification and consequent side payment) we used an "altruistic" negotiation in which agents even accept contract changes which leave them worse off! The probability of such a negative proposal being accepted, however, is – as is common in simulated annealing – a decreasing function of the extent of the loss and the time or number of steps, the negotiation process is already running. As we see from the comparison in figure 3, after an initial phase of agreeing on "really bad contracts", the altruistic agents usually generate much better contract equilibria when, with time, the probability of non-beneficial changes has decreased.



Figure 3: Welfare increase of selfish (left) versus altruistic re-contracting (right)

Unfortunately, the introduction of altruism makes the agents extremely vulnerable to exploitation. While the defecting agent may generate extraordinary profits (by always exaggerating the cost and under-reporting the benefits) the other agents may end up worse-off compared to their initial plans, most of their side payments going to the defecting agent, with no returns for themselves.

The good news is that this problem may easily be circumvented by the introduction of so-called "trust accounts" (Stockheim et al., 2005) which limit each agent's altruism towards a specific other agent: Whenever agent A's initial credit for contractor B is used up, A will only accept those re-contracting proposals of B that A will find beneficial. Our simulations showed that very small credit limits are sufficient protection against exploitation without sacrificing much of the additional benefits from altruistic re-contracting.

5 CONCLUSION AND FURTHER RESEARCH

In this paper we proposed a decentralized cooperative scheduling mechanism for a supply chain. Our mechanism is motivated by the assumption that agents who plan cooperatively in an integrated production and transport planning scenario are able to reduce supply chain costs significantly, compared with a centralized planning process that relies on traditional operations research methods where production and transportation planning are optimized separately. Our model relies on a twofold process: the supply chain partners optimize their internal process schedule on an inner level between themselves to minimize the in- and outsourcing cost of the overall production and transportation process by involving outsiders. To achieve this, production and transportation agents can improve their total profits by the re-negotiation of contract execution time with their partners in the supply chain. To prevent the permanent exploitation of agents who grant advantages to other contract partners in the negotiation process, we additionally propose the use of trust accounts. Our future work has to prove the advantages of our decentralized integrated production and transportation scheduling model over a centralized solution by conduction simulation studies based on real-world data. Further research issues could be the replacement of the WJISP with an economic version of a resource-constrained project scheduling model and the use of a multi-depot instead of a single-depot model.

References

- Boudia, M., Louly, M. A. O. and Prins, C. (2005) Global optimization of production and distribution. In Proceedings of the International Conference on Industrial Engineering and Systems Management, Marrakech (Morocco).
- Caridi, Maria, Cavalieri, Sergio (2004), Multi-agent systems in production planning and control: an overview, *Production Planning & Control*, 15 (2), 106-118
- Davis, R. and Smith, R. G. (1983) Negotiation as a metaphor for distributed problem solving. *Artificial Intelligence* 20, 63-109.
- Desrochers, M., Lenstra, J. K. and Savelsbergh, M. W. P. (1990) A classification scheme for vehicle routing and scheduling problems. *European Journal of Operational Research* 46 (3), 322-332.
- Eksioglu, S. (2002) Optimizing integrated production inventory and distribution problems in supply chains. PhD-Thesis, University of Florida.
- Elendner, T. (2003) Scheduling and combinatorial auctions: Lagrangean relaxation-based bounds for the WJISP. Manuskripte aus den Instituten für Betriebswirtschaftslehre, Christian-Albrechts-Universität zu Kiel.
- Ertogral, K., Wu, S. D. and Burke, L. I. (1998) Coordination production and transportation scheduling in the supply chain. Department of Industrial & Mfg. Systems Engineering.
- Eymann, T. (2001) Co-evolution of bargaining strategies in a decentralized multi-agent system. In AAAI Fall 2001 Symposium on Negotiation Methods for Autonomous Cooperative Systems, North Falmouth, MA, US.
- Gambardella, L. M., Taillard, E. and Agazzi G. (1999) MACS-VRPTW: A multiple ant colony system for vehicle routing problems with time windows. In D. Corne, M. Dorigo, and F. Glover, editors, New Ideas in Optimization. McGraw-Hill, London, UK, 1999.
- Garey, M. R. and Johnson, D. S. (1977) Two-processor scheduling with start-times and deadlines. SIAM Journal on Computing 6 (3), 416-426.
- Gordon, V. and Kubiak, W. (1998) Single machine scheduling with release and due date assignment to minimize the weighted number of late jobs. *Information Processing Letters* 68 (3), 153-159.
- Jung, H. and Jeong, B. (2005) Decentralised production-distribution planning system using collaborative agents in supply chain network. *The Int. of Journal Advanced Manufacturing Technology* 25, 167-173.
- Koulamas, C. and Kyparisis, G. J. (2001) Single machine scheduling with release times, deadlines and tardiness objectives. *European Journal of Operational Research* 133 (2), 447-453.
- Lau, J. S. K., Huang, G. Q., Mak, K. L. and Liang, L. (2006) Agent-based modelling of supply chains for distributed scheduling. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems & Humans* 36 (5), 847-861.
- Lei, L., Liu, S., Ruszczynski, A. and Park, S. (2006) On the integrated production, inventory, and distribution routing problem. *IIE Transactions* 38, 955-970.
- Li, H. and Lim, A. (2002) Local search with annealing-like restarts to solve the vehicle routing problem with time windows. In *Proceedings of the 2002 ACM symposium on Applied Computing*, pp 560-565, ACM Press, Madrid, Spain.
- Mui, L., Mohtashemi, M. and Halberstadt, A. (2002) A computational model of trust and reputation. pp 188-197, Big Island, Hawaii.
- Padovan, B., Sackmann, S., Eymann, T. and Pippow, I. (2002) A prototype for an agent-based secure electronic marketplace including reputation tracking mechanisms. *International Journal of Electronic Commerce* 6 (4), 93-113.
- Potts, C. N. and Van Wassenhove, L. N. (1988) Algorithms for scheduling a single machine to minimize the weighted number of late jobs. *Management Science* 34 (7), 843-858.
- Sarimento, A. M. and Nagi, R. (1999) A review of integrated analysis of production-distribution. *IIE Transactions* 31, 1061-1074.
- Stadtler, H. (2003) Multilevel lot sizing with setup times and multiple constrained resources: Internally rolling schedules with lot-sizing windows. *Operations Research* 51 (3), 487-503.
- Stockheim, T., Wendt, O. and Schwind, M. (2005) A trust-based negotiation mechanism for decentralized economic scheduling. In 38th Hawaiian International Conference on System Sciences, (HICSS-38), p 35a, Hilton, Waikoloa Village, Big Island, Hawaii.
- Vaessens, R. J. M., Aarts, E. H. L. and Lenstra, J. K. (1996) Job shop scheduling by local search. *INFORMS Journal on Computing* 8 (3), 302-317.
- Yamada, T. and Nakano, R. (1996) Job-shop scheduling by simulated annealing combined with deterministic local search. In *Meta-heuristics: Theory and applications* (Osman, I. H. and Kelly, J. P., Eds), pp 237– 248, Kluwer, Dordrecht, Netherlands.