Towards Decentralised Agent-Based Inter-Company Scheduling

Haixiao Liu and **Jeroen Keppens** and **Michael Luck**[1]

Abstract. This paper examines inter-company scheduling in scenarios where the companies wish to collaborate with one another but are not prepared to reveal operations or sales information. For example, this type of scenario occurs when multiple small to medium sized manufacturers of the same or similar products wish to pool their resources to meet larger orders even though they remain competitors with one another. Conventional approaches to inter-company scheduling are inappropriate in such situations because they rely on companies to reveal potentially sensitive information to one another or to a third party. This paper presents a novel fully decentralised agent-based approach to inter-company scheduling that substitutes information sharing for negotiation. The performance of this approach is assessed in a range of simple scenarios consisting of only two companies.

1 INTRODUCTION

It is often advantageous for separate organisations to coordinate the planning and scheduling of their activities for a variety of reasons. For example, inter-company scheduling can allow the organisations involved to operate more cost-effectively, enable them to adapt to environmental changes more quickly or achieve objectives that they could not achieve otherwise [3][16]. Various multi-agent systems for inter-company scheduling have been proposed and these have achieved considerable success [12][7].

Conventional approaches to agent-based inter-company scheduling approaches assume that organisations are prepared to share access to potentially sensitive private information, such as orders and production capacity, with a third party entity. For example, a yellow page agent is introduced in [8] to maintain the overall capacity of involved companies. Although these approaches to inter-company scheduling are effective in many scenarios, there are circumstances where the organisations that may benefit from inter-company scheduling tend to be unwilling to share as much information with a third party as these approaches require [15][4].

We envision a scenario consisting of a number of small and medium sized manufacturers that produce similar products. On the one hand, these organisations compete with one another because they manufacture similar products. As such, much of the information that is shared with a third party in conventional inter-company scheduling approaches is commercially sensitive. Therefore, the manufacturers in our scenario are unwilling to share such information. On the other hand, in certain situations, small and medium sized manufacturers can benefit from pooling resources with their competitors to achieve economies of scale or, simply, to allow them to fulfil larger orders. This scenario requires a different paradigm of inter-company scheduling: one where resource sharing is accomplished through bilateral negotiation between the organisations involved. A potential drawback of such an approach is that globally optimal scheduling solutions cannot be achieved due to lack of information sharing.

We propose a novel fully decentralised agent-based approach to inter-company scheduling that substitutes information sharing for negotiation. This approach effectively tackles the problem for two reasons. Firstly, manufacturing companies can independently make beneficial decisions to allocate their own resources. Secondly, since manufacturers do not depend on another or a third party to share resources, they may choose not to disclose private information for their own good. Therefore, our approach can solve the inter-company scheduling problem in a decentralised manner with privacy protection.

This paper presents a first step into our investigation to decentralised agent-based inter-company scheduling, by means of a simplified scenario that involves only two manufacturers. The scenario is defined in Section 2. Next, Section 3 proposes a novel fully-decentralised agent-based inter-company scheduling approach that can deal with the scenario introduced in 2. This approach is evaluated in Section 4, where we examine to what extent our approach under-performs due to the organisations' unwillingness to share information. Section 5 reviews related approaches to inter-company scheduling are reviewed. Finally, our future work towards decentralised agent-based inter-company scheduling is described in Section 6.

2 SCENARIO

Our scenario is depicted in Figure 1. It consists of two manufacturers 1 and 2, each aiming to maximise its profit. Each manufacturer receives orders for the products it can manufacture. It also obtains income by fulfilling orders. A company's production capacity is restricted by the production resources it possesses. We make the following assumptions about inter-company scheduling:

- Order: Only one type of product is required in all customer orders. Each order is not fulfilled unless the manufacturer produces as many products as stipulated in the order. No income is received for orders that are only partially fulfilled.
- Resource: Only one type of resource is requested for production. A company can increase the number of resources available for production by buying additional resources from the other company. Conversely, a company that sells resources to the other reduces the resources available to it.
- Production: There is a linear relationship between resource input and product output.

---

[1] Department of Informatics, King's College London, United Kingdom, email: haixiao.liu@kcl.ac.uk

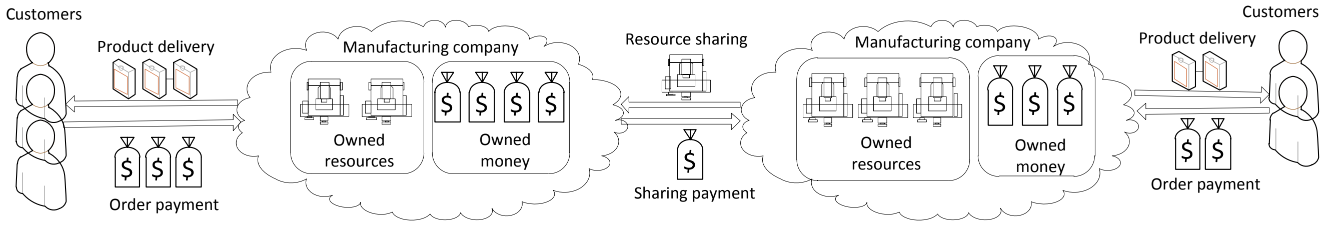

Figure 1. Our scenario

Let d be the market price for the product. The manufacturer i owns c_i units of resources and receives a set of orders O_i from customers. Each order $q \in O_i$ represents the quantity of the product required to fulfil the order. We assume that to manufacture one unit of the product, m_i units of resource are required in manufacturer i. For each order $q \in O_i$ that the manufacturer i fulfils, $q * m_i$ units of resources are allocated and an income $q * d$ is received.

A number of additional variables are introduced to represent scheduling solutions and their implications. The boolean variable ω_q describes whether or not the order $q \in O_i$ is accepted. The positive integer variable a_q expresses the number of resources assigned to order q; and the integer variable s_i indicates the number of units of resource bought or sold in manufacturer i. p_i is the price paid for sharing each unit of resource between two manufacturers. Here positive or negative values of s_i indicate different meanings as below.

- $s_i > 0$: the manufacturer i intends to purchase s_i units of resource.
- $s_i < 0$: the manufacturer i wants to sell $-s_i$ units of resource.
- $s_i = 0$: the manufacturer i does not wish to buy or sell any resources.

In our scenario, when the resource sharing occurs there is a buyer and a seller, which leads to $s_1 = -s_2$. Even when both manufacturers do not intend to share resources, this equation still holds as $s_1 = 0$ and $s_2 = 0$.

An allocation of a_q resource to production to meet order q is represented as a pair (ω_q, a_q). Let A_i be the set of all of i's production resource allocations. Then, a scheduling solution, or schedule, for manufacturer i can be expressed as a vector $x_i = \langle A_i, (s_i, p_i)\rangle$. The profit associated schedule x_i is denoted $P(x_i)$. The optimal assignment X_i is the most profitable solution of allocating resources for processing customer orders based on its own production capacity, ignoring any potential for buying or selling units of resource. Its profit P_i can be calculated from Equation 1.

$$P_i = \max\{\sum_{q\in O} q * d | O \subset O_i, \sum_{q\in O} q * m_i \leq c_i\} \tag{1}$$

If a rational manufacturer is unable to meet all its orders, it is willing to purchase them provided the cost of purchase is less than the marginal profit realised from those resources. In other words, a manufacturer is only willing to purchase resources that increases total profit. Thus, from the point of resource buyer i, then the following two constraints apply to s_i and p_i:

$$\begin{array}{l} 0 \leq s_i \leq \sum_{q\in O_i} q * m_i - c_i \\ 0 \leq p_i \leq ((s_i + c_i) * d/m_i - P_i)/s_i \end{array} \tag{2}$$

A self-interested company will only agree to sell resources instead of committing them to production if the former yields a higher overall profit than the latter. The amount of resources it can sell cannot exceed its capacity. Moreover, the profit after selling should not be less than allocating those sold resources for production or letting them remain unused. Then from the resource seller i's point of view, two constraints in an acceptable schedule x_i should be met as in Equation 3.

$$\begin{array}{l} s_i \geq -c_i \\ P(x_i) \geq P_i \end{array} \tag{3}$$

The negotiation strategy on resource sharing is used in manufacturer i to propose the quantity s_i and price p_i. In resource sharing negotiations, a prospective buyer would like to purchase resources to meet as many orders as possible for maximal income from production. Moreover, for the same amount of resource that it intends to purchase, the buyer is willing to pay as little as the seller agrees for minimal expenses.

We assume for s_i units of resources, the prospective buyer initially desires to pay b_i of marginal profit. Then the initial purchase request for prospective buyer is decided as in Equation 4.

$$\begin{array}{l} s_i = \sum_{q\in O_i} q * m_i - c_i \\ p_i = (\sum_{q\in O_i} q * d - P_i) * b_i/s_i \end{array} \tag{4}$$

The resource purchase request of manufacturer i should be updated when (s_i, p_i) is not agreed by the other. It can gradually increase the payment by g_i of marginal profit as in Equation 5 when the payment p_i does not reach the upper bound, namely $p_i < ((s_i + c_i) * d/m_i - P_i)/s_i$.

$$\begin{array}{rl} s'_i = & s_i \\ p'_i = & \min\{((s_i + c_i) * d/m_i - P_i)/s_i, \\ & p_i + ((s_i + c_i) * d/m_i - P_i) * g_i/s_i\} \end{array} \tag{5}$$

If the payment reaches the marginal profit, namely $p_i = ((s_i + c_i) * d/m_i - P_i)/s_i$, the prospective buyer may reduce the amount of resources it is willing to purchase. This process of updating (s_i, p_i) to (s'_i, p'_i) is specified in Equation 6.

$$\begin{array}{rl} s'_i = & \max\{\sum_{q\in O} q * m_i | O \subset O_i, \\ & \sum_{q\in O} q * m_i < s_i + c_i\} - c_i \\ p'_i = & (s'_i + c_i) * d/m_i - P_i)) * b_i/s'_i \end{array} \tag{6}$$

The objective of companies is to maximise their profit from fulfilling orders and sharing resources as formalised in Equation 7.

$$\max \sum_{q\in O_i} \omega_q * q * d - s_i * p_i \tag{7}$$

In order to achieve this objective, the manufacturer has to satisfy the resource and process constraints. The former one as shown in Equation 8 means that the amount of resources assigned to orders and shared with the other company should not exceed its capacity. Equation 9 indicates the process constraint that any order is not met unless the complete needed resources are allocated. In addition, the valid range of variables are specified in Equation 10. X_i can be obtained by using these equations with $s_i = 0$ and $p_i = 0$.

$$\sum_{q \in O_i} a_q - s_i \leq c_i \tag{8}$$

$$\forall q, a_q \geq \omega_q * q * m_i \tag{9}$$

$$\omega_q \in \{0, 1\}; a_q \in N. \tag{10}$$

3 APPROACH

In our scenario, each manufacturer is an autonomous decision maker. For this reason, each manufacturer is represented by an agent in our approach. Following the scenario, an agent i is defined by a tuple $\langle c_i, m_i, O_i \rangle$, where c_i represents i's resource capacity, m_i i's production process and O_i is the set of customer orders of i.

Each agent must make decisions on how to allocate resources. A unit of resource can be allocated to production, sold to the other agent or remain unused. An agent may also decide to attempt to increase the number of resource units available to it by purchasing them from its competitor and negotiate a price for this transaction. It is assumed that each agent is driven by profit maximisation.

The decision making process of a single agent in the decentralised agent-based inter-company scheduling system is shown in Algorithm 1. The algorithm consists of two phases.

In the first phase, an agent i searches for the optimal resource assignment X_i through the simplex algorithm. The profit P_i of this assignment is used as the lower limit for any decisions regarding resource sharing. If an agent is unable to fulfil all its orders by means of the resources it possesses, then it generates an initial resource purchase request $rq = (i, s_i, p_i)$. This request contains a proposal for the quantity of required resources and payment, both of which are decided according its negotiation strategy as in Equation 4. The simplex algorithm is utilised here for corresponding resource allocation for production A_i as well.

In the second phase, the agents negotiate potential resource sharing. In our simple scenario, the way in which the negotiation proceeds depends on the circumstances of the agents. If neither of the two agents sends a purchase request, both commit to allocating their resources following the optimal assignment X_i. If one sends a purchase request and the other does not, then further rounds of negotiation take place. The recipient of the message answers with a reply $rp = (answer, rq)$ where $answer \in \{accepted, declined\}$. It declines the proposal rq if agreeing to it does not enable it to increase its profit. This decline leads to the sender updating its request according to its negotiation strategy as in Equations 5 and 6. This continues unless the negotiation terminates. If both agents make rent requests, then they simultaneously send out their purchase requests until the negotiation terminates. Here agents only send out requests to purchase resources and replies to agree or disagree to the corresponding requests.

The negotiation process terminates when either an agreement is reached or when the agents discover it is not possible to reach an agreement. The latter situation occurs when an agent reaches one of its boundary conditions during the negotiation process as in Equation 2.

Algorithm 1 Decentralised agent-based scheduling approach

```
sharing_agreed = false
$P_i$ = get_profit_bound() //calculate the profit lower bound as in Equation 1
$X_i$ = get_schedule(0, 0) // use the simplex algorithm to find out the schedule $X_i$ and its profit $P_i$ in Equations 7, 8, 9 and 10 with $s_i = 0$ and $p_i = 0$
if $\sum_{q \in O_i} q * m_i < c_i$ then
    $s_i$ = initialise_quantity()
    $p_i$ = initialise_payment() // set variables as in Equation 4
    $x_i$ = get_schedule($s_i, p_i$)
    $P(x_i)$ = get_profit($s_i, p_i$) // use the simplex algorithm to find out the schedule $x_i$ and its profit $P(x_i)$ in Equations 7, 8, 9 and 10 with just initialised variables $s_i$ and $p_i$
    $rq \leftarrow (i, s_i, p_i)$
else
    $x_i \leftarrow X_i, P(x_i) \leftarrow P_i, rq \leftarrow (i, 0, 0)$
end if
repeat
    send($rq$)
    receive($msg$)
    if $msg$ is the reply $rp = (answer, rq)$ then
        if $answer = accepted$ then
            commit to $x_i$
            sharing_agreed = true
        else
            $s''_i$ = update_quantity($s_i, p_i$)
            $p''_i$ = update_payment($s_i, p_i$) // update variables as in Equations 5 or 6
            send($rq'' = (i, s''_i, p''_i)$)
            if $s''_i = 0$ then
                commit to $x_i$
                sharing_agreed = true
            end if
        end if
    else if $msg$ is the request $rq' = (i', s_{i'}, p_{i'})$ then
        if $s_i = 0 \land s_{i'} = 0$ then
            commit to $x_i$
            sharing_agreed = true
        else
            $s'_i \leftarrow -s_{i'}, p'_i \leftarrow -p_{i'}$
            $x'_i$= get_schedule($s'_i, p'_i$)
            $P(x'_i)$ = get_profit($s'_i, p'_i$)
            if $P(x'_i) \geq P(x_i)$ then
                commit to $x'_i$
                sharing_agreed = true
                send($rp = (accepted, rq')$)
            else
                if $s_i = 0$ then
                    send($rp = (declined, rq')$)
                else if $s_i > 0$ then
                    $s''_i$ = update_quantity($s_i, p_i$)
                    $p''_i$ = update_payment($s_i, p_i$)
                    send($rq'' = (i, s''_i, p''_i)$)
                end if
            end if
        end if
    end if
until sharing_agreed
```

4 ANALYSIS

In our scenario, a manufacturer may be able to increase its profits through resource sharing. However, the extent to which resource sharing is possible is constrained by the amount of information that the companies are willing to share. In this section, we compare the performance of our approach (denoted DS or decentralised scheduling) with two alternative approaches:

- Local scheduling (LS): the two agents do not exchange any information and do not share any resources as a result. In other words, two agents are isolated entities and no negotiation occur between them. This approach constitutes a baseline on the performance of DS.
- Global scheduling (GS): the resources and orders of both agents are pooled into a single agent and its optimal resource allocation is computed. In our scenario, this approach is equivalent to full information sharing. It constitutes an upper bound on the potential performance of DS.

This paper examines the effect of different features of our scenario on overall profitability under DS, compared to LS and GS, by generating random instances of our scenario on a set of input parameters. The effects of the following five input parameters are examined:

- Overall demand (OD): the total volume of products required by customers of both companies. Formally,

$$\sum_{i=1,2} \sum_{q \in O_i} q = OD \tag{11}$$

- Demand distribution (DD): the ratio of the whole demand of one company to that of the other company. Formally, $DD = x : y$ iff

$$\frac{\sum_{q \in O_1} q}{\sum_{q \in O_2} q} = \frac{x}{y} \tag{12}$$

- Requirement distribution (RD): the probability distribution used to partition the overall demand into individual order quantities. In this work, a normal distribution with a given average μ and given standard deviation σ is used. Thus, $RD = \mathcal{N}(\mu, \sigma^2)$.
- Overall capacity (OC): the ratio of the total volume of resources owned by two companies to that needed to meet the overall demand. Formally, $OC = x : y$ iff

$$\frac{c_1 + c_2}{\sum_{i=1,2} \sum_{q \in O_i} m_i * q} = \frac{x}{y} \tag{13}$$

- Resource distribution (ReD): the ratio of the volume of resources owned by one company to that of the other company. Formally, $ReD = x : y$ iff

$$\frac{c_1}{c_2} = \frac{x}{y} \tag{14}$$

4.1 Experimental Set-up

In our experiments, the following parameters are used: market price $d = 120$ and the number of resources required for production for each manufacturer $m_1 = m_2 = 4$. It is assumed that an agent requesting to purchase resources makes an initial offer of a payment of $b = 60\%$ of the marginal profits. If this offer is rejected, the agent increases the payment by $g = 20\%$ until it reaches 100% of marginal profits.

We have run five set of experiments whose input parameters are shown in Table 1. For each set of input parameters, 50 random scenario instances have been generated. The results shown are the averages over 50 experiments.

Set	Fixed Parameters	Variable Parameters
A	$DD = 1:9$, $RD = \mathcal{N}(10,1)$, $OC = 1:1$, $ReD = 9:1$	$OD = 60$, 70, 80, 90, 100, 110, 120, 130, 140, 150
B	$OD = 100$, $RD = \mathcal{N}(10,1)$, $OC = 1:1$, $ReD = 5:5$	$DD = 1:9, 2:8, 3:7, 4:6, 5:5, 6:4, 7:3, 8:2, 9:1$
C	$OD = 100$, $DD = 4:6$, $OC = 1:1$, $ReD = 5:5$	$RD = \mathcal{N}(5,1)$, $\mathcal{N}(10,1)$, $\mathcal{N}(15,1)$, $\mathcal{N}(20,1)$, $\mathcal{N}(25,1)$, $\mathcal{N}(30,1)$
D	$OD = 100$, $DD = 4:6$, $RD = \mathcal{N}(10,1)$, $ReD = 5:5$	$OC = 1:10, 2:10, 3:10, 4:10, 5:10, 6:10, 7:10, 8:10, 9:10, 10:10, 11:10, 12:10$
E	$OD = 100$, $DD = 4:6$, $RD = \mathcal{N}(10,1)$, $OC = 1:1$	$ReD = 1:9, 2:8, 3:7, 4:6, 5:5, 6:4, 7:3, 8:2, 9:1$

Table 1. Experiment Settings

4.2 Results

The results of five experiments are shown in Table 2. In order to compare the performance of three scheduling approaches, the overall profit of LS and DS are presented as the percentage of GS. For each set of input parameters, the mean overall profit is displayed.

Table 2. Results of three scheduling approaches in experiments.

Set	Value	LS	DS	Value	LS	DS
A	60	0.142	1.000	110	0.199	1.000
	70	0.157	1.000	120	0.195	1.000
	80	0.192	1.000	130	0.193	1.000
	90	0.200	1.000	140	0.191	1.000
	100	0.200	1.000	150	0.192	1.000
B	1:9	0.599	1.000	6:4	0.898	1.000
	2:8	0.700	1.000	7:3	0.799	1.000
	3:7	0.799	1.000	8:2	0.698	1.000
	4:6	0.899	1.000	9:1	0.598	1.000
	5:5	1.000	1.000			
C	$\mathcal{N}$(5,1)	0.900	1.000	$\mathcal{N}$(20,1)	0.815	1.000
	$\mathcal{N}$(10,1)	0.899	1.000	$\mathcal{N}$(25,1)	0.874	1.000
	$\mathcal{N}$(15,1)	0.867	1.000	$\mathcal{N}$(30,1)	0.714	1.000
D	1:10	0.296	0.887	7:10	0.839	0.965
	2:10	0.458	0.988	8:10	0.990	1.000
	3:10	0.422	0.929	9:10	0.967	0.996
	4:10	0.982	0.999	10:10	0.924	0.999
	5:10	0.925	0.988	11:10	0.921	0.985
	6:10	0.859	0.995	12:10	0.990	1.000
E	1:9	0.698	1.000	6:4	0.799	1.000
	2:8	0.796	1.000	7:3	0.697	1.000
	3:7	0.895	1.000	8:2	0.599	1.000
	4:6	1.000	1.000	9:1	0.500	1.000
	5:5	0.897	1.000			

These results show that in our scenario and under our experimental conditions, the decentralised scheduling obtains largely globally optimal solutions solely by negotiating the price and quantity of resources. The substantially weaker total profit results for local scheduling show that genuine resource sharing benefits are being achieved in these scenarios. For example, Figure 2 illustrates this by plotting the overall profit achieved under LS and DS, relative to GS, under the experimental conditions of set A.

Figure 3 plots the overall profit of the LS and DS approaches as

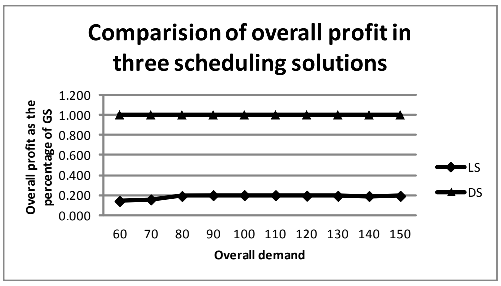

Figure 2. Results of experiment set A

the percentage of that of GS, for experiment set B. It shows that DS and LS generate the same profit result in scheduling instances with $DD = ReD$. Additionally, the bigger the difference of demand are distributed in two companies, the better DS works than LS when OD and ReD remain the same.

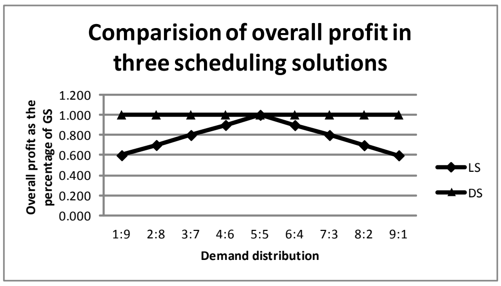

Figure 3. Results of experiment set B

There are circumstances where decentralised scheduling does not manage to produce a globally optimal solution as shown in Figure 4, which depicts the results of experiment set D. This is because in our approach the prospective purchaser initialises the resource request at the point with highest quantity and lowest payment. When the request is declined, it gradually increases the payment at first and then decreases the quantity when the payment reaches its upper bound.

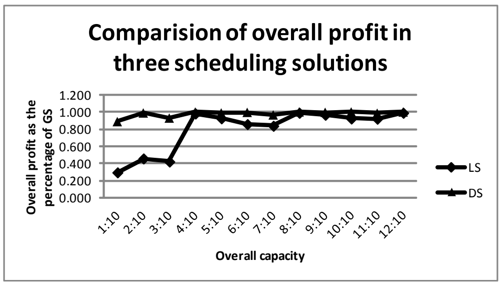

Figure 4. Results of experiment set D

5 RELATED WORK

An increasing number of papers examine developing agent-based inter-company scheduling methods in manufacturing. In them, the auction protocol has widely been used to guide agent negotiation behaviours [11]. Generally, the manager agent announces available resource capacities or production tasks and other agents reply with bids for prospective rewards [6].

Oliveira and Rocha [9] applied the auction protocol to develop an agent-based inter-company scheduling approach for scheduling production tasks as shown in Figure 5. In this approach, one customer task is not given to an individual manufacturer as private information, but published in an electronic market system. Moreover, internal scheduling decisions for production subtasks in each company should be submitted to the central market agent or delivery to other companies in order to resolve potential conflicts for processing the whole production task.

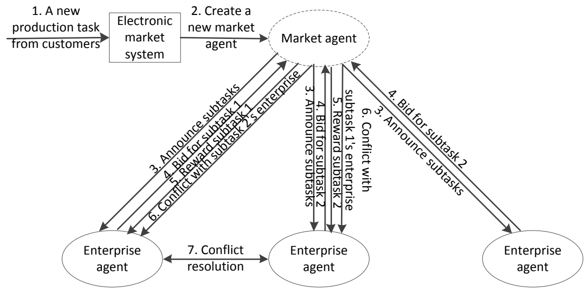

Figure 5. Example of an agent-based inter-company scheduling method

Some inter-company scheduling approaches allow individual companies to keep their production processes private, but require customer orders to be managed by third-party agents [2][13]. For example, the work presented in [8] introduces a yellow page agent maintaining the overall capacity of all involved manufacturers. The software agent common provider (SACP) representing a trusted third party is used in [1] to make scheduling decisions, which maintains a centralised repository of all available resources and a central scheduler to generate a schedule plan.

Another supply chain master planning approaches manage operational activities of organisational units for fulfilling customer demands [14]. They aim to enhance the overall competitiveness of the supply chain, especially to minimise the total relevant costs. Centralised methods may be only appropriate for static structures composed of isolated entities [10]. Although negotiation mechanism has been applied between buyer and supplier partners for collaborative planning, the private information is also exchanged (e.g. local cost effects of proposed resource supply [5]). This may be suited when both buyer and supplier come from the same organisation but not when they still compete with each other in the market.

6 CONCLUSIONS AND FUTURE WORK

This paper has presented a novel decentralised agent-based inter-company scheduling approach that operates through negotiation rather than sharing of potentially commercially sensitive information. This is useful in scenarios where the companies that need to coordinate their schedules are competitors and, therefore, reluctant to share explicit information on their orders, capacity and production plans with one another or with third party agents. We have examined the performance of our approach in a simple, two-company scenario. Through a set of experiments, we have shown that in this setting, the benefits of resource sharing can be achieved and mostly globally optimal solutions are produced.

The work presented in this paper is a first step towards the development of fully decentralised inter-company scheduling approaches. As our approach is largely negotiation based, future work will elaborate the negotiation protocol and negotiation strategies. On the one hand, the use of richer communication language will allow the agents to negotiate more efficiently. This facilitates the discovery of globally optimal solutions. On the other hand, smarter negotiation strategies will limit the amount of (potentially commercially sensitive) information that is communicated to other agents. This prevents one agent from gaining insights into the privacy of the other. However, the agents' ability to discover globally optimal solutions may be limited as well.

Obviously, the scenario employed in this paper is a simple one. The scenario was kept simple with a view to be able to study it more effectively. Future work will examine the effect of such extensions. In particular, the introduction of a temporal dimension has many implications, not only for the simulation but the negotiation strategies as well. For instance, in such a setting, different types of resources (capital resources, resources that are consumed in production and perishable resources) raise different considerations. Also, the scheduling problem becomes a dynamic optimisation problem. Furthermore, in such a setting an agent's effectiveness is affected by its ability to predict future events (e.g. by learning from past experience).

Another important generalisation of the work is the introduction of additional agents. Again, this affects the negotiation in a variety of ways. In particular, adding agents makes negotiations multilateral instead of bilateral, which has implications on both the negotiation protocol and an individual agent's negotiation strategies. However, the computation complexity has to be considered when scaling up the inter-company scheduling problem.

REFERENCES

[1] A. Aerts, N. Szirbik, and J. Goossenaerts, 'A flexible, agent-based ICT architecture for virtual enterprises', *Computers in Industry*, **49**(3), 311–327, (2002).

[2] L. Camarinha-Matos, 'Multi-agent systems in virtual enterprises', in *Proceedings of the 11th International Conference on AI, Simulation and Planning in High Autonomy Systems*, (2002).

[3] L. Camarinha-Matos and H. Afsarmanesh, 'Virtual enterprise modeling and support infrastructures: Applying multi-agent system approaches', in *Multi-agent systems and applications*, eds., M. Luck, V. Marik, O. Stepankova, and R. Trappl, 335–364, Springer, (2001).

[4] P. Davidsson, S. Johansson, J. Persson, and F. Wernstedt, 'Agent-based approaches and classical optimization techniques for dynamic distributed resource allocation: A preliminary study', in *AAMAS03 workshop on Representations and Approaches for Time Critical Decentralized Resource/Role/Task Allocation*, (2003).

[5] G. Dudek and H. Stadtler, 'Negotiation-basedcollaborativeplanning between supply chains partners', *European Journal of Operational Research*, **163**(3), 668–687, (2005).

[6] S. Kraus, 'Automated negotiation and decision making in multiagent environments', in *Multiagent Systems and Applications*, eds., M. Luck, V. Marik, O. Stepankova, and R. Trappl, 150–172, Springer, (2001).

[7] L. Monostori, J. Vancza, and S. Kumara, 'Agent-based systems for manufacturing', *CIRP Annals - Manufacturing Technology*, **55**(2), 697–720, (2006).

[8] T. Norman, A. Preece, Chalmers. S., N. Jennings, M. Luck, V. Dang, T. Nguyen, V. Deora, J. Shao, W. Gray, and N. Fiddian, 'Agent-based formation of virtual organisations', *Knowledge-Based Systems*, **17**(2-4), 103 – 111, (2004).

[9] E. Oliveira and A. Rocha, 'Agents advanced features for negotiation in electronic commerce and virtual organisations formation processes', in *Agent Mediated Electronic Commerce*, eds., F. Dignum and C. Sierra, 78–97, Springer, (2001).

[10] R. Pibernik and E. Sucky, 'An approach to inter-domainmasterplanning in supplychains', *International Journal of Production Economics*, **108**(1-2), 200–212, (2007).

[11] J. Reaidy, P. Massotte, and D. Diep, 'Comparison of negotiation protocols in dynamic agent-based manufacturing systems', *International Journal of Production Economics*, **99**(1-2), 117–130, (2006).

[12] W. Shen, Q. Hao, H. Yoon, and D. Norrie, 'Applications of agent-based systems in intelligent manufacturing: An updated review', *Advanced Engineering Informatics*, **20**(4), 415–431, (2006).

[13] W. Shen, F. Maturana, and D. Norrie, 'MetaMorph II: An agent-based architecture for distributed intelligent design and manufacturing', *Jounal of Intelligent Manufacturing*, **11**(3), 237–251, (2000).

[14] H. Stadtler, 'Supply chain management and advanced planning-basics, overview and challenges', *European Journal of Operational Research*, **163**(3), 575–588, (2005).

[15] G. Weiss, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, MIT Press, 2000.

[16] M. Wooldridge, *An introduction to multiagent systems*, Wiley, 2009.