Self-regulation of Agents Using Individual Profit Expectations in Multi-agent Scheduling for Supply Management

Alexander Tsarev
Smart Solutions, Ltd.
Samara, Russia
mail@identifier.at

Petr Skobelev
Smart Solutions, Ltd.
Samara, Russia
petr.skobelev@gmail.com

Igor Mayorov
Smart Solutions, Ltd.
Samara, Russia
mayorov@smartsolutions-123.ru

Abstract

The paper focuses on the development of methods of self-adaptation of agent societies in multi-agent scheduling systems allowing them to achieve higher scheduling quality in changing environments. The paper introduces a self-regulation method based on the agent expectation of achievable profit and on the level of truthfulness in agent interactions. It is described how the profit expectations and the declaration of these expectations affect the scheduling quality in different cases. The paper shows the importance of coordinated behaviour of agents. The approach to coordinated self-regulation and to proactive schedule improvement is proposed. Finally, the results of real data scheduling using the proposed approach are given.

1 Introduction

Growing complexity and dynamics of modern global market demand new paradigms in resource management [1][2]. New revolutionary approach to increase efficiency of business is associated today with real-time economy, which requires adaptive reaction to events, ongoing decision making on resource scheduling and optimization and communication results with decision makers.

Multi-agent technology is considered as a new design methodology and framework to support distributed problem solving methods in real-time scheduling and optimization of resources [3][4].

The main feature of real-time scheduling and optimization methods is to produce a result in the specified moment of time or time interval, reacting to unpredictable external and internal, constructive or destructive events (new order coming, order is cancelled, resource unavailable, etc.).

The quality and efficiency of decision making in resource scheduling and optimization process can be influenced by the number of factors: the intensity of events flow, the number and current state of resources, individual specifics of orders and resources, time interval between the events and processing time for events, productivity of resources and many others.

A big challenge is to ensure that certain quality of scheduling results is achieved in a short time after the event to make it possible to finish the processing before the next event and to always have a valid schedule needed for decision-making.

In this paper, we propose a method to improve the quality of multi-agent supply scheduling based on self-regulation of the agents.

2 Supply scheduling task description

The aspects of agent self-regulation are researched based on the supply scheduling task. The supply network is defined as a set of sites interconnected with channels. There may be a demand for specific product and quantity at specific sites, and there may be a limited availability of the product at specific sites. The channels may be used to relocate the products between sites, but each channel may have a cost applied to each item transported using it. In general, the availability may be limited due to a production capacity, delivery constraints, lead times, channel and storage capacities, but we consider a simplified case ignoring the time issues and actual causes of the limited availability. The supply schedule in this simplified case is just a set of instant relocations (source, destination, product, and quantity) to be done in order to satisfy the demand. Although this simplification is too significant to consider all the details of the supply scheduling from such simplified point of view, in our understanding, it does not affect the aspects of scheduling that we discuss in this paper. Specifically, the high-level and very generic mechanisms allowing agents to achieve better scheduling results.

In order to measure the quality of the scheduling result, we use profit. Each individual demand has a revenue associated. The total profit of the result is calculated as a sum of all revenues of satisfied orders deducted by the sum of all costs

978-1-943436-00-2 / copyright ISCA, CAINE 2015
October 12-14, San Diego, California, USA

(costs of relocations in this case). This quality measurement is represented by the following total quality equation.

$$Q = \sum_{i \in satisfied} revenue_i - \sum_{i \in scheduled} cost_i$$

Further in the paper, the description is based on the following example of the network structure (Figure 1).

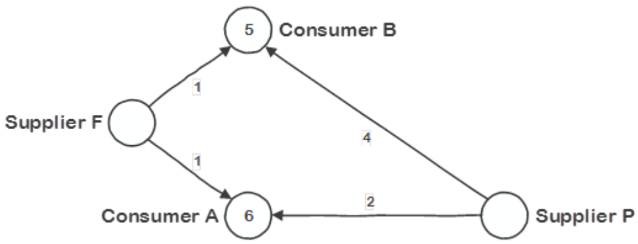

Figure 1: Supply network example

In this example, both consumers have a demand for one item of an abstract product. Consumer A generates revenue of 6 virtual coins if its demand is satisfied, while consumer B generates 5.

The costs of delivery of the product item between the suppliers and the consumers are presented in the Table 1.

Table 1: Delivery costs

Source	Destination	Cost per item
F	A	1
F	B	1
P	A	2
P	B	4

Suppliers in the network have a limited availability of the product. Thus, the scheduling task is to achieve as much quality (Q) as possible by choosing the right supplier for each demand.

In the more generic case, which we consider in practice (out of scope of this paper), product transformations, delivery durations, and constrained storages are considered.

$$\sum_p \sum_t \sum_m \begin{pmatrix} \sum_d D^{revenue}{}_{pmtd} D_{pmtd} - \\ -\sum_l T^{cost}{}_{pmtl} T_{pmtl} - \\ -\sum_n \sum_c R^{cost}{}_{pmnct} R^{in}{}_{pmnct} \end{pmatrix} \to max$$

Where p is a product, t is time interval, m is the specific site in the network. $D^{revenue}{}_{pmtd}$ and D_{pmtd} are the revenue and the volume of satisfied demand d. $T^{cost}{}_{pmtl}$ and T_{pmtl} are the cost and volume of production batch l. $R^{cost}{}_{pmnct}$ and $R^{in}{}_{pmnct}$ are the cost and volume of relocation from site n to site m using channel c.

This and the related constraints make the task more computationally complex. I this case, it becomes very time consuming to get the optimal solution using conventional methods (e.g. simplex algorithm). This is the reason, why we use the multi-agent approach to schedule supply in complex networks.

For the purposes of this paper, it is more convenient to consider the simplified definition, as the described approach to self-regulation remains the same.

3 Multi-agent supply scheduling

In multi-agent approach, the solution is found via negotiations between agents representing different entities of the task.

The scheduling process is triggered either by a new data context (initial data) being supplied to the scheduling engine or by a change set describing external events.

When the engine is supplied with the initial data via API, the scheduling dispatcher creates site agents and distribute data to them. Each site gets data about its own parameters, consumption demands, connected channels and related sites.

In case of events (data changes supplied via API) the scheduling dispatcher first checks if there are changes in the network structure. It creates or deletes site agents if needed and then send site level changes to them.

Sites do not know about the whole network, they work asynchronously and coordinate with each other using messages. Thus, the logic of the supply network scheduling consists of 2 parts:

- Site scheduling, which is the logic of scheduling within each site.
- Network scheduling, which is the logic of coordination between sites in the network.

Within each site, the scheduling is driven by the demand defined on that site. “Demand” is the request of a specific product quantity at a specific time.

If a site does not have unprocessed demands, it does not do anything. It is idle and waits for the requests from other sites or for new external events. On the other hand, if the site does have demand that is unfulfilled, the scheduling process is initiated.

Common practice at the start of scheduling is that the input data only contain demand at the sinks in the network. Thus, the agents of nodes that are upstream from the sinks. So in a network where shops are sinks, upstream factories and DCs are idle until the shops have created an internal schedule by processing the demands.

Each site gets the following input data.

- Product information, including prices, volume and weight.
- Current stock levels.
- Storage cost function.
- Any transformation processes, product compatibility constraints
- Calendars containing times where the facility is operating.
- Incoming channels, including channel constraints and transportation cost function.
- Consumption demands, including products and quantities.

The result of the intra-site scheduling is a planned set of production operations and deliveries (including time, channel, products and quantities) needed to satisfy the consumption demands, meet the limitations and keep the profit as high as possible.

After the site is fully scheduled, the site agent creates messages to coordinate with other sites.

For each delivery in the intra-site schedule, the site agent sends a message to the upstream site(s) with the requested quantity of products, time when they should be sent and the maximal cost (reward) the site can afford.

For each lost sale generated during the intra-site scheduling, the site agent sends a message to the downstream site that the requested demand is not possible to fulfil at this time for the proposed reward. If the site itself is an end-node, i.e. sink, nothing will happens except that the lost sales are recorded in the KPIs.

If a demand was successfully scheduled within the proposed reward, the site agent sends the information about the "actual" cost to the requesting site to make it possible to review the decisions and improve the quality of the schedule. The difference between "actual" costs and "temporary" is that the "actual" cost requires all intra-site schedules to be completed, which can change based on the effective utilization within the site.

If a site gets a message that a delivery is not possible from another site, the scheduled delivery is cancelled and the demands that depend on its scheduled arrival will attempt to find other options during the intra-site scheduling. For example, a site can first schedule a delivery using a cheap channel, but when the delivery request is processed by the upstream site it finds out that the supplying site does not have enough stock and therefore the site cannot supply a downstream site with the stock it has available. Then the downstream sites request is rejected, whereby it will attempt to request fulfilment of its demand using a more expensive channel.

If a site gets a feedback about the "actual" higher cost of supply, the site may also revise the decisions and try to identify another option to reduce costs.

In this paper, we consider primarily the interaction between the site agents (suppliers and consumers) and the ways to improve their decisions via self-regulation to achieve better results in the limited time.

4 Success expectation and agent "mood"

When a consumer agent negotiates with a supplier agent, it declares a certain reward that it is ready to pay for the resource allocation (e.g. for delivery of the product). This reward is a part of the profit that the consumer agent is able to generate, if the demands are satisfied. The higher the reward, the more chances are that the supplier chooses this specific consumer to deliver to it instead of another one, if the resources are limited.

The maximum reward that the agent can declare is obviously the revenue of the demand itself (assuming there are no explicit costs reducing the profit). Declaring the reward, the demand agent considers that its proposal will probably compete with the proposals of other demand agent. Considering this, it seems reasonable that each agent declares the maximum reward in order to be as competitive as possible. Then, the demand with higher profit has more chances to win the competition. If this is the case, in the network example above, the consumer A proposes supplier F a reward of 5=6-1, and the consumer B – a reward of 4=5-1.

But this approach does not necessarily lead to the best possible solution, as it is shown below. In some sense, unnecessary competition may lead to worse results. If there is no lack of resources, it becomes more important to coordinate between agents to find a better resource distribution. From this point of view, we suggest to calculate the reward considering the internal agent mood and known alternative delivery options. Thus, if a demand can be satisfied using one of two options, it will request the resources using the cheaper option first, but it will propose only the difference between the two options as the reward. In this case, the supplier F gets a proposal of 1=(6-1)-(6-2) from A and 3=(5-1)-(5-4) from B.

This means that the demand does not compete that much for the cheaper option, if it has an alternative. Of course, this behaviour is reasonable only if all other consumers requesting the same supplier are also "truthful", and propose reduced rewards considering their options. Then, the resource is provided to those who are not just more profitable, but also more limited in options. Still, having another option does not mean it is possible to negotiate it successfully. It means that the proposal of the reduced reward is a risk for the consumer agent. It can accept this risk if it is in "optimistic mood", or reject it being in "pessimistic mood", or choose a compromise. The "mood", in its turn, may depend on the previous agent experience, approximate resource availability analysis, etc. Being in a more "optimistic mood", the agent expects higher chances that the options it has may be successfully used. In order to reflect its expectation during negotiations, the demand agent reduces the proposed reward by the profit it can obtain using the next potentially possible option. This shows that the demand is not interested that much in this specific resource, and gives more chances to the demands with worse alternative options. If the reality matches the agent expectations, the total profit should be higher.

Let us consider the result that we can obtain in the example above (section 2) with different resource availability and different agents "mood". In the trivial case, if the resources are not limited, both consumers are supplied from F regardless the rewards they propose.

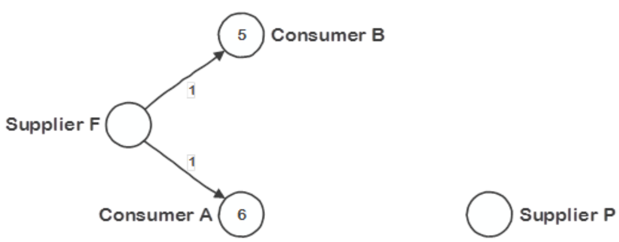

Figure 2: Deliveries in case of high product availability

Then, the total schedule quality is Q=(6-1)+(5-1)=9.

If only one item is available for supply from F and one from P, the first scheduling result depends on the agents "mood". In this situation, the most profitable solution is to supply A from P, and B from F. In order to come to this result with a minimum of negotiations, the consumer agents should propose F the rewards based on optimistic estimations. "Optimistic" means they expect that in the worst case it is still possible to get the product from P.

Thus, A proposes F the reward of 1=(6-1)-(6-2), B proposes F the reward of 3=(5-1)-(5-4). The supplier F having only one item available chooses to supply to B to get higher reward. Then A gets a rejection and asks P for supply. Finally, we get a result with total profit Q=(6-2)+(5-1)=8.

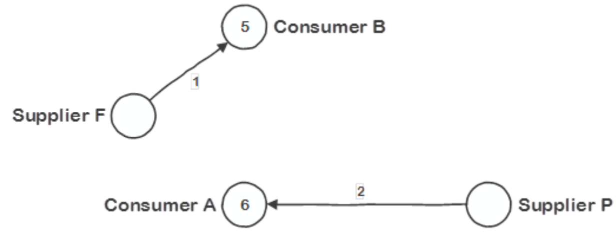

Figure 3: Deliveries in case of normal product availability with optimistic mood

The scheduling process go differently, if the agents are in pessimistic mood. Then, they do not expect that they can satisfy the demand in other ways and consider the supply from F as a last chance. A proposes F the highest reward of 5=6-1, B proposes F the reward of 4=5-1. F chooses A to supply it. Then B is supplied from P. The total profit of this solution is Q=(6-1)+(5-4)=6, which is worse than in optimistic case.

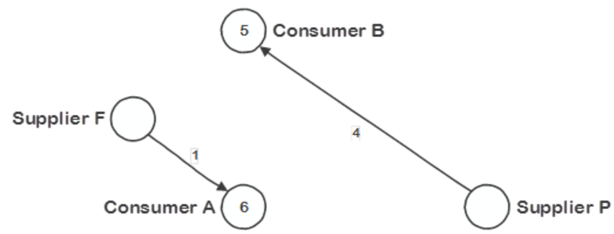

Figure 4: Deliveries in case of normal product availability with pessimistic mood

Next, let us consider the case when only one item of the product is available in the network at supplier F. The scheduling logic in this case is the same except the demand rejected by F in the first run is left unsatisfied because no product can be supplied from P. Then, for optimistic case, we get Q=6-2=4, while for pessimistic case Q=6-1=5.

We can see that in this simple example the higher quality is achieved when the agent expectations meet the resource availability, which looks reasonable.

Table 2: Scheduling quality

	Low availability	Normal availability	High availability
Pessimistic expectations	5	6	9
Optimistic expectations	4	8	9

5 Proactivity and self-regulation

The described effects can be used to achieve better results in scheduling using less negotiations. In this section, the proposed self-regulation method is described.

Each agent has its own mood value in the interval [0;1], which may change during the agent life cycle. When the scheduling is started with new data, the default mood for all agents is set to zero. After the first result is achieved, the agents analyse how well their expectations fitted into the reality.

It can happen that an agent (being in pessimistic mood) is forced to explore other options during the scheduling, and it finds out that the demand can be satisfied with one of them. In this case, it changes its mood to the more optimistic value for ($mood = (mood + 1)/2$).

In an opposite case, an agent can discover that no option (e.g. no supplier) can satisfy its demand. Then it changes the mood to the more pessimistic one ($mood = mood/2$).

It is important to note that the mood is not related to a specific scheduling option. The agent expectation of successful scheduling is a more general "feeling" about overall situation, and characterizes the agent even if the set of options change from run to run (e.g. different deliveries are formed).

If the agent does not get any new information about available options for K runs, it switches the mood to a random value in order to explore if it allows agent to achieve other results.

Next time the re-scheduling based on events is started, the agents behave according to their new mood. It is assumed that events do not change the situation significantly, so the new mood is mostly valid for the changed environment. If no new events come to the system, the re-scheduling is started proactively with the new mood in order to try to find a better solution.

During the scheduling, the agents should be coordinated with each other. The purpose of this coordination is to establish the same effective mood in the groups of agents based on the suppliers they negotiate with. The effective agent mood is also in the interval [0;1] and is calculated every time as the average of moods of agents in the group G_S of agents communicating with the same supplier S.

$$mood^{i}_{effective} = \frac{\sum_{i \in G_S} mood^{i}}{N_{G_S}}$$

N_{G_S} – number of agents in the group.

The effective mood affects the reward proposed by the agent to specific supplier according to the formula below.

$$reward = revenue - cost_{best} -$$

$$-mood_{effective} * (revenue - cost_{next})$$

The point is that if one of the agents behave based on the pessimistic mood while others are in optimistic mood, it gets more attention from the supplier. As experiments show, such egoistic behaviour prevents the agent society from finding a better solution.

The method of self-regulation was tested using a network of 20 suppliers and 200 consumers with random distribution of costs and available products. The chart below (Fig. 5) shows how the average agents mood achieved by the scheduling engine after several re-schedules depends on the availability of resources (0 - no resources, 1 - resource availability matches demand). Re-schedules were initiated by random events of demand and resource availability changes in random places of the network. It is clear from the chart that a higher mood is achieved in higher resource availability conditions.

$$mood_{average} = \frac{\sum_i mood^{i}}{N}$$

N – total number of agents.

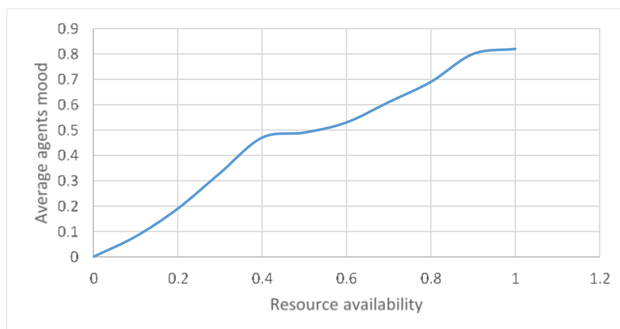

Figure 5: Average agents mood achieved for different product availability

The next chart (Fig. 6) shows how the self-regulation improves the quality achieved in reactive mode. It is the quality of the first consistent scheduling result, which agents achieve just after the event. It is usually not very optimal and requires proactive improvements. We can see from the chart, that with each processed event, due to the self-regulation that happens during the processing, the next event is processed with higher quality.

The quality on this chart is normalized (divided by the maximum that can be achieved). The events in this experiment do not change the ratio between demands and resources. If the demand is increased at a random site in the network, we also increase available resources at a random supplier to keep the ratio constant.

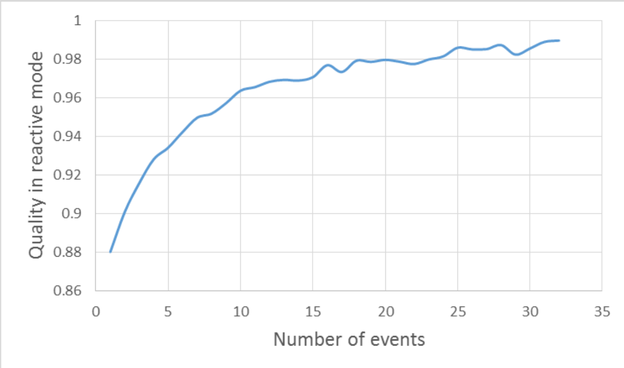

Figure 6: Average scheduling quality in reactive mode after the specific number of events

6 Conclusion

The proposed method of self-regulation of agents in multi-agent supply scheduling system is based on the agent expectations regarding the profit it can achieve in the current situation. This expectation is formalized as an agent "mood" property, which is tuned based on the agent experience. This allows to achieve better results when the new events are processed.

The further work will be focused on more sophisticated regulation of the mood of the agents in case of complex interrelations between the agents in the supply network.

Acknowledgement

This work was carried out with the financial support of the Ministry of Education and Science of the Russian Federation (contract 14.579.21.0084, project unique id RFMEFI57914X0084).

References

[1] G. Rzevski and P. Skobelev, Managing complexity, 2014, WIT Press, Boston.

[2] A. Park, G. Nayyar, and P. Low, Supply Chain Perspectives and Issues, A Literature Review, April 21, 2014, Fung Global Institute and World Trade Organization.

[3] A. Mohammadi and S. Akl, Scheduling Algorithms for Real-Time Systems, Technical Report, 2005, no. 2005-499, School of Computing Queen's University, Kingston.

[4] M. Joseph, Real-time Systems: Specification, Verification and Analysis, Prentice Hall, 2001.

[5] M. Pinedo, Scheduling: Theory, Algorithms, and Systems, Springer, 2008.

[6] J. Leung, Handbook of Scheduling: Algorithms, Models and Performance Analysis, CRC Computer and Information Science Series, Chapman & Hall, 2004.

[7] S. Binitha and S. Sathya, "A survey of bio inspired optimization algorithms", Int. Journal of Soft Computing and Engineering, 2012, vol. 2, issue 2, pp. 2231-2307.

[8] S. Sun and J. Li, "A two-swarm cooperative particle swarms optimization", Swarm and Evolutionary Computation, 2014, vol. 15, pp. 1-18. Elseiver.

[9] M. Tasgetiren, M. Sevkli, Y. Liang, and M. Yenisey, "Particle swarm optimization and differential evolution algorithms for job shop scheduling problem", International Journal of Operational Research, 2008, vol. 3, no. 2, pp. 120-135.

[10] V. Vittikh and P. Skobelev, "Multiagent interaction models for constructing the demand-resource networks in open systems", Automation and Remote Control, 2003, vol. 64, issue 1, pp. 162-169.

[11] Y. Chevaleyre, et al. "Issues in Multiagent Resource Allocation", https://staff.science.uva.nl/u.endriss/MARA/mara-survey.pdf, retrieved: March 2015.

[12] M. Barbuceanu and M. S. Fox, "Coordinating multiple agents in the supply chain", Proceedings of the fifth workshops on enabling technology for collaborative enterprises, WET ICE'96, IEEE Computer Society Press, 1996, pp. 134-141.

[13] T. Stockheim, M. Schwind, O. Wendt, and S. Grolik. "Coordination of supply webs based on dispositive protocols", 10th European Conference on Information Systems (ECIS), Gdañsk, 6-8 June 2002.

[14] A. Oliinyk, "The multiagent optimization method with adaptive parameters", Artificial Intelligence journal, 2011, no. 1, pp. 83-90.