

AGENT SYSTEM APPLICATION IN HIGH-VOLUME PRODUCTION MANAGEMENT

Martin Reháč¹, Petr Charvát² and Michal Pěchouček¹

¹Gerstner Laboratory
Czech Technical University in Prague
Technická 2, Prague 6, 166 27 CZECH REPUBLIC
{pechouc,rehakm1}@labe.felk.cvut.cz

²CertiCon, a.s.
Václavská 12, Prague 2, 120 00 CZECH REPUBLIC
charvat@certicon.cz

Based on actual industrial project on which the Gerstner laboratory has collaborated, we present a multiple-level scheduling approach as a mean to efficiently apply agent-based planning systems in high-volume production environment. Brief description of efficient and reconfigurable high-level scheduler based on linear programming, as well as design elements of low-level agent-based planning are included

1. INTRODUCTION

Agent systems are currently predominantly used in project-oriented production management (Pěchouček et al., 2002) where they offer a significant competitive advantage by easily adopting to naturally very dynamic environment. In this work, we study an application of multi-agent systems in highly specialized and high volume manufacturing plant. Underlying research is an extension of real industrial application project executed for major automotive production plant in Eastern Europe.

First, we shall specify the client requirements on system function and behavior. Then, we shall introduce a concept of multi-level partially distributed planning realized in cooperation between dedicated high-level planning agent and agents representing real-world physical entities. High-level planning agent, its model, algorithm and current implementation is presented in section 4 and low-level planning and production management in the section 5. Conclusions are drawn in the section 6, together with future work directions.

2. CLIENT REQUIREMENTS AND PROBLEM STATEMENT

Automotive industry operates in high volumes and on very low margins, thus it focuses a lot of attention on process optimization. Such optimization can be specified by following generic requirements, derived from the project specific requirements drawn by our client. For through motivation of these criteria, see (Goldratt, 1990).

- Minimize the stock through the production chain, thus decreasing the financial and storage costs.
- Maximize the production uniformity, to be able to use the industrial means in an efficient manner and to avoid overtime cost.
- Minimize the unnecessary handling of products between successive steps of the production process to further reduce human resources and other manipulation related costs.
- Allow the integration with production surveillance and management tools.
- Allow real-time or almost real-time re-planning in case of demand changes or production anomalies.
- Allow easy and straightforward process reconfiguration in the future (strengthening the bottlenecks of the production process)

It is interesting to note that the quantitative (first three) criteria listed above are completely contradictory and that a satisfactory optimum is their weighted, context dependent combination.

Problem statement

Factory in question contains three serially organized production lines, as shown on the figure.

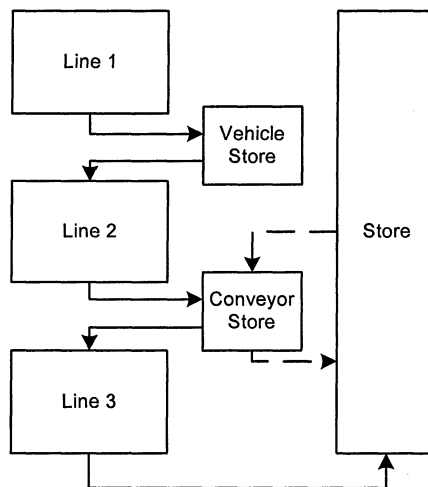


Figure 1 – Factory Outline

The factory production means can be described as three lines in a series, with two buffer stores and one main store used for final product storage before delivery

and for the intermediary product storage as well. The intermediary product can be bought from outside or shipped directly to clients, as it may be a part of deliveries described below. Material store is not represented on drawing, as the material is delivered to different positions on production lines when required. Demand is formalized as a matrix defining how many products of a given type shall be delivered on a given day.

3. MULTI-LEVEL PLANNING

In order to successfully plan in this environment, we shall note that parallelism is limited to using pre-completed parts instead of completing them locally, or to producing partly assembled components for the completion in other plants. Such production is represented by dashed paths in the figure.

Therefore, the use of classic negotiation techniques is highly constrained due to the fact that there are almost no alternative paths and the whole planning problem is reduced to pure scheduling in a static environment, featuring an enormous solution space. In this case, a single, dedicated planning agent has a considerable advantage of global knowledge that allows it to use a heavyweight but efficient scheduling methods. On the other hand, a today's user will typically require a rapid process reconfiguration ability to follow process or manufacturing equipment evolutions and a close integration with on-line production surveillance & analysis tools. This is the area where agent based distributed and adaptive systems have significant advantages (Bussmann et al., 2001).

The solution we propose is based on a compromise between these two approaches. We have divided the scheduling process to two distinctive phases and distributed different scheduling tasks between these phases. *High level scheduling* agent is responsible for the fulfillment of the first two requirements, minimum stock and maximum production uniformity, together with ensuring that the deliveries are feasible. Output of this planning stage is the size of lots to produce on different lines during the given day.

Low level scheduling agents process the output of high level scheduling and organize the work on their lines during the appropriate time period by distributing the lots of products into appropriate timeslots. In this phase, we handle the production continuity, ensuring that the line 2 immediately consumes the product produced on line 1 and that the same applies for Lines 2 and 3. Production surveillance may be connected to this process for dynamic rescheduling.

In the scope of our case, two levels of planning are completely sufficient. With increasing problem scope, or while extending it to extra-enterprise environment (Mařík et al., 2002) we will probably need to add extra layers of planning above the current high-level scheduling. This will allow an effective choice of production site for each task. As a negotiation or auction methods are probably best suited for this layer, an additional future requirement on high-level scheduling is its speed, allowing us to answer the bids for production instantly.

The planning agents may be directly integrated with the embedded holonic systems in production equipment, provided by manufacturers. This will empower both the high-level and low-level scheduling by giving it more information about the equipment and allowing it not only to react to problems when they happen but also to proactively predict problems and include contingency in the plan.

4. HIGH-LEVEL SCHEDULING

This section describes the model and algorithm used by high-level scheduler to determine how many products to produce in a given time interval. First, we will formally restate the problem (simplified) solved by high-level scheduler, explicitly specifying the limitations to respect and the parameters to optimize. We will also briefly discuss some interesting aspects of the solution and implementation.

Formal Problem Statement

Let's have J products groups ($j = 1, 2, \dots, J$). The time horizon is divided into T time intervals ($t = 1, 2, \dots, T$). The production can be scheduled on P processors ($p = 1, 2, \dots, P$). Let's also have the expedition demand d_{jt} , for each product group j and time interval t . Further, we denote q_{pjt} the production volume of the product group j in the time interval t on the processor p and I_{jt} as stock of the product group j at the end of the time interval t . Mark I_{j0} the initial stock of the product group j . We intuitively request all elements of Q to be positive and integer.

Denote $z_{jp} \in \{0, 1\}$ the ability to produce the product group j on the processor p . The production volume of the product group j in the time interval t on all processors is then

$$Q_{jt} = \sum_{p=1}^P z_{jp} q_{pjt}$$

The production relationships is described by square matrix $S_{J \times J}$. For the production of one item of the product group x we need $s_{x,y}$ items of the product group y . The production demand is then

$$v_{jt} = \sum_{x=1}^J s_{xj} Q_{xt} \quad \forall j, t$$

We can describe the relation among the production, stock and demand as

$$I_{j,t-1} + Q_{jt} - I_{jt} = d_{jt} + v_{jt} \quad \forall j, t$$

Let the C_{pt} is the capacity of the processor p in the time interval t . For production of one item of the product type j $r_{jp} > 0$ units of processor p are necessary. Then we can describe the constraint on the production induced by the capacity of the processor with following relation

$$\sum_{j=1}^J r_{jp} z_{jp} q_{pjt} \leq C_{pt} \quad \forall p, t$$

The L_{pjt}^{\min} is the minimal lot-size. We also request the lot-size (number of products of type j produced in day t on processor p) to be either zero or more then this minimal lot size value due to material handling efficiency issues and non-zero switching times.

Limitations do apply also on the stock of the products produced and ready for shipment. Typically, a certain amount of pieces of each product (c_{jt}^{\min}) is kept in reserve in order to be able to replace the non-produced or incorrectly produced pieces of this type. On the other side, the capacity of the store is physically limited

to certain amount of pieces of all products, weighted by their respective space consumption. This capacity is denoted M_t .

Our goal is to satisfy all orders, respect all the limitations described above and to minimize the stock and production variability. Stock minimization can be described as

$$\sum_{t=1}^T \sum_{j=1}^J I_{jt} \rightarrow \min$$

and production variability can be described using the following relations:

$$\sigma_p = \frac{1}{T} \sum_{t=1}^T |\omega_{pt} - \bar{\omega}_p| \rightarrow \min$$

with

$$\bar{\omega}_p = \frac{1}{T} \sum_{t=1}^T \omega_{pt} \quad \text{where} \quad \omega_{pt} = \frac{\sum_{j=1}^J r_{jp} z_{jp} q_{pjt}}{C_{pt}} \in \langle 0,1 \rangle.$$

Problem Solution Elements

After careful deliberation and several experiments, linear programming was chosen for implementation of high-level scheduler agent. The main factors beyond this choice were its speed, robustness and an ability to detect the constraint preventing us from achieving our goal. However, the application of this method in our case is not straightforward, because the problem as specified is not entirely linear. We had to resolve following issues:

1. the lot-size has to be an integer,
2. the production uniformity relation is non-linear,
3. lot-size has to be either zero or more than the minimum value.

Integer values of lot sizes are an issue that is easy to resolve. Either we can use integer extension of standard LP algorithm, which is NP-hard and complicates the solution, or we can simply ignore this issue and round the results of LP algorithm. The rounding error caused by this approach is (in most cases, as well as in our case) insignificant compared to total number of products.

Non-linearity of the production uniformity relation is an issue that is much harder to resolve. We have opted for an alternative approach that modifies the conditions of the original model by requesting the production not to divert from the average required production value by more than certain percent. In practice, we replace the condition presented above by following inequalities.

$$\sum_{j=1}^J r_{jp} z_{jp} q_{pjt} \leq C_{pt} \Rightarrow B_p^{low} C_{pt} \leq \sum_{j=1}^J r_{jp} z_{jp} q_{pjt} \leq B_p^{high} C_{pt}$$

Average load used in the relation to determine the boundary values can be calculated per processor, per processor group or for the whole plant, depending on client preferences.

Third problem, minimum lot size issue was solved by iterated runs of the algorithm. In the first run, we use the model as described above and we use the resulting

production matrix Q to modify the conditions for the second run according to the following condition.

$$q_{pjt} \leq \alpha L_{pjt}^{\min}$$

If the condition is satisfied, then q_{pjt} is fixed to 0. Otherwise, we require it to be bigger than the minimum lot size. The value of the parameter α can either be set manually by operator, or deduced by the system from its previous experience.

Implementation

For the implementation of our current high-level scheduler, a free third party LP solver was used, together with communication and data transformation wrapper. The whole scheduling takes less than 1 second on standard PC (with 28 days, 50 products and 3 processors), thus completely satisfying the performance requirements resulting from frequent re-planning and future possible requirements resulting from the integration with another negotiation-based planning layer.

In the case that the solution cannot be found, we can use the solver with modified problem to identify the critical limitation and to communicate it to operator or other appropriate system component.

Linear programming model is also rather easy to extend or modify in case of plant reconfiguration. We can simply add new production lines or products, together with their properties and modify the Z matrix, describing the ability of processor to produce different products.

In the next part, we will propose equally reconfigurable solution for low-level scheduling and production surveillance as an extension to today's traditionally less flexible solutions.

5. LOW-LEVEL SCHEDULING AND PRODUCTION CONTROL

Low level scheduling will process the required daily quantities determined by preceding high-level scheduling. Its task is to order the lots on the lines during the given day to minimize the manual manipulation and material handling related to product switches. In our model, we associate a cost to switching between two types of products on a line. This cost is relatively low for products that share a major part of their components, but grows with increasing differences between products.

Single processor problem statement

For a single line (processor), we may describe the succession of different products as an oriented graph, where the nodes represent different lots and the evaluations of edges connecting them represent the cost of succession of these particular lots in the order determined by the orientation of edges (See figure for simple example).

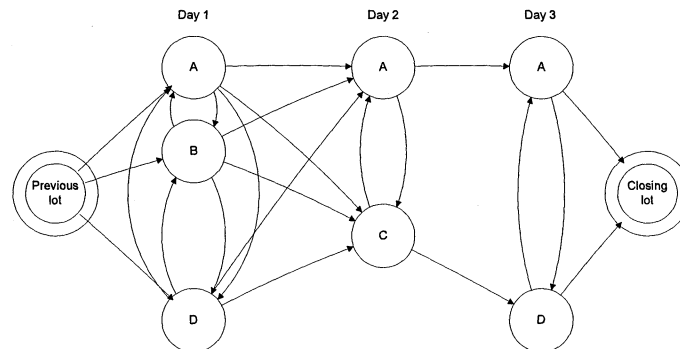


Figure 2 – Single line lot ordering problem

Note that the lots scheduled for one day are connected in both directions, as any of them may precede another, while the connections between successive days are only from past to the future. When we establish such graph for all the lots during the planning period, we may note that finding an optimum order is equivalent to finding a Hamiltonian path (a path passing through all nodes) through this graph, connecting the first and last lot. Such task is NP complete (see (Blazewicz et al., 2001) for alternative prove) and there is no trivial way of finding an optimal solution. However, the frequent changes of delivery orders, together with non-uniformities of the production process make the complete and uninterrupted execution of the plan highly unlikely. Therefore, we have opted for the use of greedy algorithm, which seeks the local optimum and selects the batch with locally cheapest transition, expecting that future gain from accepting locally suboptimal decision is highly unlikely to be collected anyway.

Extension to multiple processors

In reality, few industrial processes are executed by single processing unit and our case is no exception to this rule. This poses us in front of another obstacle, as we must ensure the continuity between successive processors. These processors may have incompatible preferences concerning the products to manufacture in a given moment. In this stage, the agent nature of low-level scheduler may prove to be advantageous, as the negotiations between different processor-representing agents would be able to efficiently organize the lot order in a given day.

In accordance with principles defined by the theory of constraints, we may see that within a single day scope, our bottleneck is actually defined by the high-level scheduling output. Therefore, throughout the negotiations, we shall prioritize the agents representing the components with load closest to nominal in a given day. This shall ensure the feasibility of fulfillment of the goals specified by the high-level scheduler. Other components switching costs may be discounted compared to the bottleneck switching. This approach uses the information prepared by the high-level scheduling to make the low-level optimization more efficient and relatively straightforward.

Production surveillance and dynamic re-planning

As already stated above, a smooth execution of any plan is rarely natural and often

requires many local or even major global adjustments. Today, such adjustments are decided by human managers based on their observations of the production process and their experience, even if automatic solutions start to appear (Bussmann et al., 2001). We propose that the processor planning agents shall be also used for online process surveillance and that the information gathered by these agents may be used both to increase the experience of agents and to react immediately to current situation, using the agent experience (Mařík et al., 2002) and (Pechoucek et al., 2000). This approach would allow us to eliminate many false alarms, connected with slow production start or short-time material inaccessibility, but can sooner detect potential major problems, especially by looping back the information from quality assurance stations.

6. CONCLUSIONS AND FUTURE WORK

The work described in this article is based on an industrial application project to which the Gerstner laboratory has contributed by its experience in industrial process planning and control. Major features of the design presented in this article is the separation of high-level and low-level scheduling, enabling us to benefit both of the global view of the dedicated scheduling component and flexibility, reactivity and potential learning ability of agent based systems. We've successfully integrated a linear programming methods into the project solution and demonstrated the complementarity between this classical approach and dynamic agent systems. Even if the final implementation of this particular project will not probably directly use agent framework (due to system integration and technology transfer issues), we are looking forward to integrate the planning component and other features from this project with current, more project oriented software tools to extend their reach to high-volume manufacturing.

7. REFERENCES

1. Blazewicz, J, et al. *Scheduling Computer and Manufacturing Processes* – 2. ed. Springer Verlag, Berlin 2001
2. S. Bussmann, K. Schild: *An Agent-based Approach to the Control of Flexible Production Systems*, in: *Proc. of the 8th IEEE Int. Conf. on Emergent Technologies and Factory Automation (ETF A 2001)*, p.481-488 (Vol.2). Antibes Juan-les-pins, France, 2001.
3. Goldratt, Eliyahu M. *The Theory of Constraints*, Croton-on-Hudson, N.Y.: North River Press, 1990.
4. Mařík, V., Pěchouček, M., and Štěpánková, O.: *Social Knowledge in Multi-Agent Systems*, In: *Multi-Agent Systems and Applications (M. Luck et. al, eds.)*, LNAI 2086 Tutorial, Springer-Verlag, 2001, pp. 211-245.
5. Mařík, V. - Pěchouček, M. - Vokřínek, J. - Říha, A. *Application of Agent Technologies in Extended Enterprise Production Planning*. In: *EurAsia-ICT 2002: Information and Communication Technology*. Berlin : Springer, 2002, p. 998-1007. ISBN 3-540-00028-3.
6. Pěchouček, M. - Říha, A. - Vokřínek, J. - Mařík, V. - Pražma, V. *ExPlanTech: applying multi-agent systems in production planning*. In: *International Journal of Production Research*. 2002, vol. 40, no. 15, p. 3681-3692. ISSN 0020-7543
7. Pěchouček, M., Mařík, V., Štěpánková, O., 2000, *Role of Acquaintance Models in Agent-Based Production Planning Systems*. In M. Klusch and L. Kerschberg (eds), *Cooperative Information Agents IV – LNAI 1860*, (Heidelberg: Springer Verlag), p. 179-190