A Model for Multi-Plant Production Planning Coordination

Conrad Oboulhas, Xu Xiaofei, and Zhan Dechen
Department of Computer Science and Engineering, Harbin Institute of Technology, China

Abstract: *This paper deals with the problem of coordinating production planning process that becomes more complex in today's highly competitive manufacturing environment. Our main concern is to develop a model for coordinating the multi-plant production planning when the demand at any plant can be satisfied by producing at any other plant. The proposed model uses the multi-agent system and linear programming technique to achieve minimized production cost, increased flexibility and faster response.*

Keywords: *Multi-plant production planning, coordination, linear programming, multi-agent system.*

Received January 26, 2004; accepted August 22, 2004

1. Introduction

The intensive global competition and rapid development of new products today increase the difficulty of production planning coordination for multi-plant companies. Under these circumstances, the multi-plant companies need to take appropriate actions so that they could remain competitive in the global market situation. Multi-plant organization is a type of those supply chain, which behaves like a single company through strong coordination and cooperation towards a mutual goal. The production planning coordination of activities across a network of plants becomes critical to quickly respond to dynamic market conditions, reduce costs, faster response times more rapidly and effectively. To maximize coordination in a complex and heterogeneous multi-plant production environments, an agent-based technology approach provides a natural way to streamline communication among plants. In this paper, we describe the total production cost minimization problem faced by a production planning center of a multi-plant company. We formulate the problem as a linear programming problem and we propose an agents system to coordinate the production.

The remainder of this paper is organized as follows. In section 2, we introduce the notion of the multi-plant production planning. The problem is formulated and linear programming technique is proposed, which is shown to be more effective in cost minimization problem. In section 3, we present the Multi-Agent System (MAS) and show that agent-based approach potentially offers many advantages for the development of distributed intelligent manufacturing systems. The system architecture and negotiation process among agents are proposed. Section 4 presents the prototype implementation of the multi-agent system. Section 5 briefly presents a case study, while the conclusion is made in section 6.

2. Multi-Plant Production Planning

In recent years, the practice of coordination for multi-plant production environment is extremely common in many manufacturing industry settings, such as electronics assembly, semiconductor and automobile manufacturing industries. The need for coordination is expected to grow due to the following factors and trends: Changing customer profile, globalization, unique and special requirements, etc. Companies can profit from this coordination, improving their competitiveness, through a better estimation of the demand and costs' reductions from planning improvements. Production planning process in a multi-plant facility is a large-scale problem that many industries with complex operations face on a continual basis [11] and different computational models have been proposed.

Bhatnagar and Chandra [1] defined the property of a coordinated multi-plant production planning problem as seeking to link together the production plans of hierarchical integrated manufacturing plants. Therefore, this problem intends to coordinate the production processes of several manufacturing plants under hierarchical supply chain structure in an efficient manner. In other words, the objective of coordination is to achieve better performance for the entire network, such as minimizing total relevant costs, shortening the orders manufacturing flow-time, ect.

Moon *et al.* [8] further investigates an integrated process planning and scheduling problem for a multi-plant supply chain network. Their problem is formulated as a multiple Travelling Salesman Problem

(TSP) model with alternative machines, sequence-dependent setup, and distinct due dates. The objective is to find an optimal schedule for minimizing total tardiness by analyzing the machines and operation sequences in multi-plant supply chain. In order to obtain a good near-optimal solution, genetic algorithm-based heuristic approach is developed. Because the features of alternative machine, sequence dependent set, transfer batch and process batch are all considered in their model, their research is rich in the concepts. However, most of these researchers on coordinated multi-plant production planning environment have focused on optimization models using Linear Programming (LP) or Non-Linear Programming (NLP) techniques [10]. These models often provide a good solution but are not always the best because of the limitations of the methods [6]. They can be difficult to use and are technically insufficient when processing a high volume of what-if scenarios. Since the efficiency of a supply chain is affected by a large number of interacting factors, the knowledge of how the factors combine to influence the chain is very valuable. In this paper, we describe the production cost minimization problem faced by Production Planning Center (PPC) of a multi-plant company when the demand at any plant can be satisfied at any other plant.

2.1. Problem Formulation

We consider a company operating several plants. The PPC receives the demand for the set of products that the plants manufactured. Each plant P can manufacture all the products and satisfy demand d from another plant. For each product the bill of materials is known, i. e., the set of parts and materials that compose each product reference. Due to the differences in technology, the production cost/time, holding cost and setup cost/time were different at different plants. The PPC must decide which products to be produced in each plant and their quantities.

2.2. Mathematical Model

The following is a mathematical programming formulation.

Sets:

p = the index denoting the plant, p=1,…,n.

k = the index denoting the period, k=1,…,i.

Decision variables:

X_{pk} = Quantity to be produced at plant p during period k.

I_{pk} = Inventory to be carried at plant p at the end of period k.

$$Y_{pk} = \begin{cases} 1, \text{ there if there is setup at plant p during k} \\ 0, \text{ otherwise} \end{cases}$$

Parameters:

d_{pk} = Amount demanded at plant p during period k.

M_{pk} = Marginal cost of producing a unit if product at plant p during period k.

S_{pk} = Cost of setup at plant p during period k.

C_{pk} = Cost of carrying a unit in inventory at plant p during period k.

Objective function:

The aim is to minimize the total production cost of the company over the given period (horizon):

Minimize

$$\sum_{p} \sum_{k} [\, M_{pk} \,.\, x_{pk} + C_{pk} \,.\, I_{pk} + S_{pk} \,.\, Y_{pk}]$$

Subject to

$$x_{pk}, I_{pk} \geq 0$$

3. Multiple Agent System for Multi-plant Production Planning

3.1. Motivation

Over the past few years, multi-agent systems have received an increasing amount of attention. Indeed, they were used for the development of wide range classes of applications such as: Distributed systems, distributed problem solving, modeling and simulation of complex systems. A multi-agent system is one in which a number of agents cooperate and interact with each other in a complex, widely distributed and ever-changing manufacturing environment to achieve particular, possible common goals. MAS approach offers significant advantages to overcome the problems of complexity (i. e., large volume of data, production capacity distributed among resources having conflicting objectives, etc.) and uncertainty (i. e., noises coming from the field, lead times, etc.). According to Wooldridge and Jennings [14] an agent is a computer system that has the following properties: Autonomy, social ability, reactivity and pro-activeness. Basically an agent is intelligent, if it perceives its environment and is capable of reasoning its perceptions, solve problems, and determine actions depending on its environment and tasks, which were given to it by its user [4, 13]. It is possible for intelligent agents to learn as they communicate with their users or other agents. Jennings *et al.* [5] provided a detailed review of the field. Many multi-agent approaches have been proposed to carry out supply chain and enterprise integration platforms implementation. Sikora and Shaw [12] provided a multi-agent framework for the coordination and integration of information systems. Yan *et al.* (2000) [15] developed a multi-agent based negotiation support system for distributed electric power transmission cost allocation based on the network flow model and Knowledge Query and Manipulation Language (KQML).

3.2. Framework

The architecture of MAS for multi-plant planning coordination is shown in Figure 1. In this architecture, each interface agent is linked with mediator agent to find other agents in the complex and heterogeneous multi-plant environment. After receiving a demand for the set of products, PPC sends order production requests to competent plants for products production; a plant receives a production order request, finds some competent agent for further task execution; LCA negotiates with resource agents for machining operations, awards machining operation tasks to suitable resource agents, and then sends related information back to PPC and PPC can make decision (products to be produced in each plant and their quantities, etc.) in order to minimize the total production cost. If the related factories are not able to produce the requested products before the due date, they need to contact the customer for negotiating a new due date.

The basic agents in this architecture are described as follows:

- *Mediator Agent (MA)*: MA acts to coordinate the activities of other agents and resolve conflicts between them. It generates it's own commitments based on the messages received from other agents and what it has learned from the performance of the learners then finds the "all-agreed" solution in a short time period.
- *Interface Agents (IA)*: IA are software that aims at assisting users in performing computer-based tasks [7]. Such agents learn by "watching over the shoulder" of the user and detecting patterns and regularities in the user's behaviour. We consider that interface agents may have the following characteristics: communicative, semi-autonomous, collaborative, reactive, pro-active, adaptive, self-aware, and mobile.
- *Knowledge Management Agents (KMA)*: KMA are able to retrieve information for a specific application from knowledgebases and databases via intelligent querying mechanisms such as data mining and knowledge discovery. They are also able to store, transform, and in some cases (e. g., mobile agents) transport information. The key issue to developing knowledge management agents is to develop efficient mechanisms for knowledge acquisition, representation, learning and reasoning.
- *Local Coordinator Agents (LCA)*: LCA represent agents in their area and help them to initiate agent-to-agent interactions. Areas are required to have exactly one local area coordinator, which is an agent that acts as a facilitator for other agents within its area. Agents will use the services of local coordinator agent to access other agents in the system. Agents requesting the use of materials in an area need to go through a registration process with the LCA in that area. Agents succeeding in their registration are expected to surrender some of their autonomy in exchange for the use of resources.
- *Resource Agents (RA)*: RA are manufacturing resources (raw materials, parts, manpower, machines, transport and stocking facilities, etc.) and software programs. Resource agents are coordinated by LCA.

The descriptions of agents can be illustrated in Table 1.

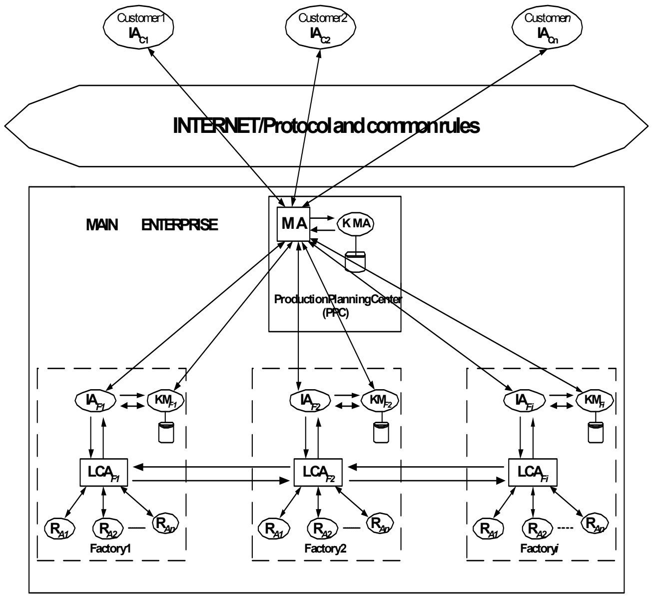

Figure 1. Agent-based architecture of multi-plant production coordination system.

3.3. Negotiation Among Agents

The two most widely used Agent Communication Languages (ACL) are the KQML and the Foundation for Intelligent Physical Agents (FIPA). These languages are highly versatile and therefore can be used in conversations and not only for message passing. The newest language at this moment is the Java-based Knowledge Query and Manipulation Language (JKQML). Negotiation is the communication process of a group of agents in order to reach a mutually accepted agreement on some matter [2]. The cooperative negotiation strategy can be applied to MAS where the individual agents are cooperative and will collaborate in order to achieve a common goal, for the best interest of the system as a whole. In order to work cooperatively, in this system, the mediator agent will select a resource agent to perform the task according to its criteria and award a contract to it. Other resources are not selected to perform the task, but they are taken as alternatives, which may be contacted (negotiated with) in the future in the case of unforeseen situations. This will greatly reduce the rescheduling time when such unforeseen situations take place. The information about the alternative resources (raw materials, manpower, machines, transport and stocking facilities, etc.) will be sent to the LCA agents with the information about the selected resources and the LCA agents will save these two sorts of information in their knowledgebases. When the selected resources cannot perform the scheduled tasks due to unforeseen situations, the LCA agent may negotiate directly with alternative resource agents.

Table 1. Description of MAS.

Agents	**Perception**	**Knowledge/ Decisions/Actions**	**Environment**
Mediator Agent (MA)	Messages from LCA, RA, KMA, IA	• Send requests. • Receive proposals. • Evaluation proposals. • Negotiation. • Decision making. • Coordination.	All agents
Interface Agent (IA)	Messages from MA, KMA, LCA, User	• Decision making. • Receive or not proposals (requests). • Agree or not proposals (requests).	Users, all agents
Knowledge Management Agent (KMA)	Messages from MA, KMA, IA, LCA	• Retrieve information from knowledgebases and databases.	All agents
Local Coordinator Agent (LCA)	Messages from MA, RA, KMA, IA	• Initiate agent-to-agent interactions in their area. • Collect the requirement from resource agents. • Coordination/decision for various plants (factories).	All agents
Resource Agent (RA)	Messages from LCA	• Constructing requirements schedules.	All agents

4. Prototype Implementation

Agent-based systems are difficult to build and debug from scratch. Therefore, it is important to use agent-building tools, such as IBM Aglets Software Development Kit (also known as Aglets Workbench), Jave-based Agent Framework for Multi-Agent Systems (JAFMAS) [3], and Java Agent Template Lite (JATLite). Peng *et al.* [9] studied the use of agents for designing supply chain management systems and concluded that there are three requirements that are essential:

- *A common ACL and protocol:* This is the language the agents use to exchange messages. In this work, we used the KQML. KQML is both a message format and a message-handling protocol to support run-time knowledge sharing among agents. KQML can be used as a language for an application program to interact with an intelligent system or for two or more intelligent systems, to share knowledge in support of cooperative problem solving.
- A *common format* for the contents of communications is necessary. Agents should be able to access a set of neutral commands. Neutrality implies that the contents may be used for different purposes. The STandard Exchange of Product model data (STEP) and eXtensible Mark Language (XML) are investigated as the content languages.
- The agents also need to use *a shared ontology.* Ontology is a formal description of entities and their properties, relationships, constraints and behaviour. Ontology is a "model of the world". Agents must share the same conceptual model of the domain problem in order to understand the concepts and get effective communication, and the data models supported by the content languages represent this model of the environment.

Figure 2 shows the language implementation levels. The language chosen to implement the system is Java. The main reason for using Java is that it is Internet oriented and platform independent. The most important Java library for the development of this project is JATLite, a package of programs developed in Java by the Centre for Design Research of the University of Stanford. It allows users to quickly create new-layered software agents that communicate robustly over the Internet and exchange information with other agents on other computers where they are running. JATLite provides a set of Java templates and a Java agent infrastructure that make it easy to build systems in a common way. The template for building agents uses a common high-level language and protocol, and provides users with numerous pre-defined Java classes that facilitate agent construction. A useful feature of JATLite is that it can "wrap" existing and often stand alone programs by providing them with a front-end that allows them automatically to communicate with other programs, sending and receiving messages and files. The second level of implementation in our project uses KQML as the protocol language. KQML is the "shell" that provides information about the sender and receiver agents, the subject of the message and additional information. Finally, the innermost layer in the agent communication framework is represented by a neutral file exchange format, for which we use XML.

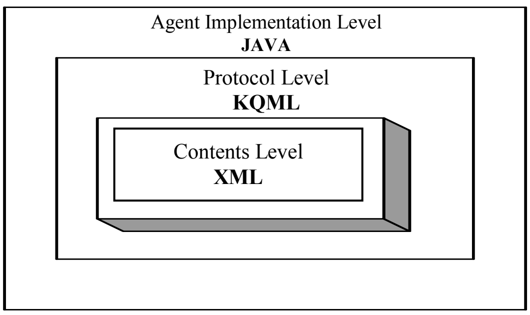

Figure 2. Level of implementation.

5. Case Study

A large automotive industry corporation composed of a PPC and 8 plants. The corporation produces vehicles (3 types), engines, transmissions and stampings. The customer places an order for purchase of 30 vehicles of type VHA6371C. Each plant can manufacture all the vehicle types and satisfy production demand from other plants. Due to the differences in technology, the production cost/time, holding cost, setup cost/time and so on at different plants, the production planning center must decide in which plant to produce the vehicles and in what quantity.

5.1. Scenario

- PPC receives a production order from some customer for 30 vehicles product VHA6371C with delivery due date.
- PPC sends production requests to competent plants for vehicle production; Plants receive a vehicle production request, finds some competent agents (through LCA) which can satisfy vehicle production request.
- LCA negotiates with suitable resource agents and then sends related information back to PPC. If the related plants are not able to produce the requested vehicle before the due date, they need to contact the customer for negotiating a new due date.
- PPC use MA for monitoring and making decisions.
- The PPC calculates

$$\sum_{p} \sum_{k} [\, M_{pk} \,.\, x_{pk} + C_{pk} \,.\, I_{pk} + S_{pk} \,.\, Y_{pk}]$$

6. Conclusion

In this paper, a model is developed to solve the multi-plant production planning coordination problem, so that the total production costs is minimized. The model approach is an ideal platform for information exchange that helps a decision-maker with useful information to production coordination among the plants by intelligent agents in order to produce the right product in the right place at the right time for the right price. Agent-based approaches offer many advantages for the development of distributed intelligent manufacturing systems: Modularity, reconfigurability, scalability, upgrade ability, and robustness. A multi-agent system prototype is being implemented. This shows the feasibility of the proposed architecture. More research is still underway, before it is applied on an industrial process.

References

[1] Bhatnagar R., Chandra P., and Goyal S. K., "Models for Multi-Plant Coordination," *European Journal of Operations Research,* vol. 67, no. 2, pp. 141-160, 1993.

[2] Bussmann S. and Müller J., "A Negotiation Framework for Cooperating Agents," *in Proceedings of CKBS-SIG*, Dake Centre, University of Keele, pp. 1-17, 1992.

[3] Chauhan D. and Baker A. D., "JAFMAS: A Multiagent Application Development System," *in Proceedings of the Second International Conference on Autonomous Agents*, Minneapolis, USA, pp. 100-107, 1998.

[4] Durfee E., Lesser V., and Corkill D., "Trends in Cooperative Distributed Problem Solving," *IEEE Transactions on Knowledge and Data Engineering*, vol. 1, no. 1, pp. 63-83, 1989.

[5] Jennings N. R., Sycara K., and Wooldridge M., "A Roadmap of Agent Research and Development," *International Journal of Autonomous Agents and Multi-Agent Systems*, vol. 1, no. 1, pp. 7-38, 1998.

[6] Kafoglis C., "Maximize Competitive Advantage with a Supply Chain Vision," *Hydrocarbon Processing*, pp. 47-50, February 1999.

[7] Maes P., "Agents that Reduce Work and Information Overload," *Communications of the ACM,* vol. 37, no. 7, pp. 31-40, 1994.

[8] Moon C., Kim j., and Hur S., "Integrated Process Planning and Scheduling with Minimizing Total Tardiness in Multi-Plants Supply Chain," *Computers and Industrial Engineering*, vol. 43, no. 1-2, pp. 331-349, July 2002.

[9] Peng Y., Finin T., Labrou Y., Cost S., Chu B., Long J., Tolone W. J., and Boughannam A., "An Agent-Based Approach for Manufacturing Integration - The CIIMPLEX Experience," *International Journal of Applied Artificial Intelligence*, vol. 13, pp. 39-63, 1999.

[10] Pinto J. M., Joly M., and Moro L. F. L., "Planning and Scheduling Models for Refinery Operations," *Computers and Chemical Engineering*, vol. 24, pp. 2259-2276, 2000.

[11] Shah N., "Single and Multisite Planning and Scheduling: Current Status and Future Challenges," *in Proceedings of AIChE Symposium Series*, Foundations Of Computer Aided Process Operations (FOCAPO'98), Snowbird, USA, pp. 75-90, 1998.

[12] Sikora R. and Shaw M. J., "A Multi-Agent Framework for the Coordination and Integration of Information Systems," *Management Science,* vol. 44, no. 11, pp. 65-78, 1998.

[13] Sycara K., Pannu K., Williamson M., Zeng D., and Decker K., "Distributed Intelligent Agents," *IEEE Expert*, vol. 11, no. 6, pp. 36-46, 1996.

[14] Wooldridge M. and Jennings N. R., "Intelligent Agents: Theory and Practice," *The Knowledge Engineering Review*, vol. 10, no. 2, pp. 115-152, 1995.

[15] Yan Y., Yen J., and Bui T., "A Multi-Agent Based Negotiation Support System for Distributed Transmission Cost Allocation," *in Proceedings of the Thirty-Third Annual Hawaii International Conference on Systems Sciences (HICSS-33)*, IEEE Computer Society Press, Los Alamitos, CA, January 2000.

Conrad Oboulhas has obtained his PhD degree in computer science and applications in 2005 from the School of Computer Science and Technology at Harbin Institute of Technology (HIT). His research interests include Computer Integrated Manufacturing System (CIMS), supply chain management, management, and decision support systems.

Xu Xiaofei is a professor and dean of the School of Computer Science and Technology at Harbin Institute of Technology (HIT). His research interests include computer integrated manufacturing system, database systems, supply chain management, agile virtual enterprises, management and decision information system, and knowledge engineering. He has published more than 200 academic papers. He is the standing member of the Council of China Computer Federation, member of the Expert Group for Discipline of Computer Science and Technology in the Academic Degree Committee of China State Council.

Zhan Dechen is a professor in the School of Computer Science and Technology at Harbin Institute of Technology (HIT). His research interests include computer integrated manufacturing system, enterprise management, decision support systems, and ERP.