10th WSEAS Int. Conf. on MATHEMATICAL METHODS AND COMPUTATIONAL TECHNIQUES IN ELECTRICAL ENGINEERING (MMACTEE'08), Sofia, Bulgaria, May 2-4, 2008

Cooperative Intelligent System for Manufacturing Scheduling

ANA MADUREIRA JOAQUIM SANTOS IVO PEREIRA

GECAD – Knowledge Engineering and Decision Support Group
Institute of Engineering – Polytechnic of Porto
Porto, Portugal

Abstract – Hybridization of intelligent systems is a promising research field of computational intelligence focusing on combinations of multiple approaches to develop the next generation of intelligent systems.

In this paper we will model a Manufacturing System by means of Multi-Agent Systems and Meta-Heuristics technologies, where each agent may represent a processing entity (machine). The objective of the system is to deal with the complex problem of Dynamic Scheduling in Manufacturing Systems.

Keywords: *Dynamic Scheduling, Multi-Agent Systems, Meta-Heuristics.*

I. Introduction

A manufacturing system has a natural dynamic nature observed through several kinds of random occurrences and perturbations on working conditions and requirements over time. For this kind of environment it is important the ability to efficient and effectively adapt, on a continuous basis, existing schedules according to the referred disturbances, keeping performance levels. The application of Meta-Heuristics to the resolution of this class of real world scheduling problems seems really promising.

In order to innovate production, new organizational and technological paradigms are needed to reply to the modern manufacturing systems challenges. The traditional structure of manufacturing industries is constructed upon the three pillars of land, labour and capital. The challenge is to move towards a new structure, which can be described as 'innovating production', founded on knowledge and capital. Future manufacturing solutions must identify multiple perspectives and linkages between novel approaches to customisation, customer response, logistics and maintenance. The current typically linear approach to research, development, design, construction and assembly will be replaced by simultaneous activity in all areas to satisfying global demand and shorten time-to-market [12].

Scheduling resolution requires the intervention of highly skilled human problem-solvers. This is a very hard and challenging domain because current systems are becoming more and more complex, distributed, interconnected and subject to rapidly changing. For these dynamic optimization problems environments, that are often impossible to avoid in practice, the objective of the optimization algorithm is no longer to simply locate the global optimal solution, but to continuously track the optimum, or to find a robust solution that operates optimally in the presence of perturbations [3][5][7][10][15].

Multi-agent paradigm is emerging for the development of solutions to very hard distributed computational problems. This paradigm is based either on the activity of "intelligent" agents which perform complex functionalities or on the exploitation of a large number of simple agents that can produce an overall intelligent behavior leading to the solution of alleged almost intractable problems. The multi-agent paradigm is often inspired by biological systems.

Meta-Heuristics form a class of powerful and practical solution techniques for tackling complex, large-scale combinatorial problems producing efficiently high-quality solutions. From the literature we can conclude that they are adequate for static problems. However, real scheduling problems are quite dynamic, considering the arrival of new orders, orders being cancelled, machine delays or faults, etc. Scheduling problem in dynamic environments have been investigated by a number of authors, see for example [3][5][10][15].

In this paper we will model a Manufacturing System by means of Multi-Agent Systems and Meta-Heuristics technologies, where each agent may represent a processing entity (machine). The objective of the system is to deal with the complex problem of Dynamic Scheduling in Manufacturing Systems. Our approach shows that a good global solution for a scheduling problem may emerge from a community of machine agents solving locally their schedules while cooperating with other machine agents that share some relations between the operations/jobs. Meta-Heuristics (Tabu Search or Genetic Algorithms) can be adapted to deal with dynamic problems, reusing and changing solutions/populations in accordance with the dynamism of the Manufacturing System. Self-parameterization of the Meta-Heuristics allows for a better adaptation to the situation being considered. The idea is that each agent adopts and sets the self-parameterization in accordance with the problem being solved (the method and parameters can change in run-time).

Coordination Mechanisms are used to guarantee the feasibility of schedules. Notice that joining problems that were locally solved will not guarantee the feasibility of schedules (e.g. precedence relations could not be guaranteed). The cooperation mechanism will be established between machine agents involved in the execution of operations (jobs) with precedence relations in order to deal with the feasibility of the generated schedules in run-time.

Considering that the inherent nature of current manufacturing systems is distributed we will address the complex dynamic scheduling problems in a distributed way using the Multi-Agent paradigm. The proposed architecture is based on Team-Work characteristics due to its philosophy of cooperation.

Team-oriented programming suggests a number of different approaches to the definition of agent teams and their coordination in order to achieve common goals. Some MAS organizational [7][20] are evaluated in order to define the proposed cooperation mechanism.

The remaining sections are organized as follows: in section 2 the scheduling problem under consideration is presented. Section 3 summarizes some important MAS aspects like possible coordination approaches that must be considered. Section 4 presents the MASDScheGATS Systems and describes implemented mechanisms. Finally, the paper presents some conclusions that were obtained from our model and puts forward some ideas for future work.

II. Problem Definition

Real world scheduling problems have received a lot of attention in recent years. In this work we consider the resolution of realistic problems. Most real-world multi-operation scheduling problems can be described as dynamic and extended versions of the classic Job-Shop scheduling combinatorial optimization problem.

In practice, many scheduling problems include further restrictions and relaxation of others. Thus, for example, precedence constraints among operations of the different jobs are common because, often, mainly in discrete manufacturing, products are made of several components that can be seen as different jobs whose manufacture must be coordinated. Additionally, since a job can be the result of manufacturing and assembly of parts at several stages, different parts of the same job may be processed simultaneously on different machines (concurrent or simultaneous processing).

Moreover, in practice, scheduling environment tends to be dynamic, i.e. new jobs arrive at unpredictable intervals, machines breakdown, jobs can be cancelled and due dates and processing times can change frequently.

The problem, focused in our work, which we call Extended Job-Shop Scheduling Problem (EJSSP) [10],[11], has major extensions and differences in relation to the classic Job-Shop Scheduling Problem (JSSP). In this work, we define a job as a manufacturing order for a final item, that could be Simple or Complex. It may be Simple, like a part, requiring a set of operations to be processed. We define it as Simple Product or Simple Final Item. Complex Final Items, requiring processing of several operations on a number of parts followed by assembly operations at several stages, are also dealt with. The main elements of the EJSSP problem could be modelled as shown in the following subsections.

A. Jobs

- A set of multi-operation jobs $J_1,\ldots,J_n$ has to be scheduled. d_j is the due date of job J_j. t_j is the Initial processing time of job J_j . r_j is the release time of job J_j.
- The existence of operations on the same job, on different parts and components, processed simultaneously on different machines, followed by components assembly operations (multi-level jobs).
- The existence of different job release dates r_j and due dates d_j.
- The possibility of job priorities definition, reflecting the importance of satisfying their due dates, being similar to the weight assigned to jobs in scheduling theory.
- Precedence constraints among operations of the different jobs.
- The existence of operations on the same job, with different parts and components, processed simultaneously on different machines.
- New jobs can arrive at unpredictable intervals.
- Jobs can be cancelled.
- Changes in task attributes can occur: Processing times, date of deliver, priorities.

B. Operations

- Each operation O_{ijkl} is characterized by the index (*i, j, k, l*), where *i* defines the machine where the operation *k* of job *j* is processed and *l* the graph precedence operation level (level *1* correspond to initial operations, without precedents).
- Precedence constraints among operations of the different jobs.
- Each job J_j consists of one or more operations O_{ijkl}., where:
- IO_{ijkl} is the time interval for starting operation O_{ijkl}
- r_{ijkl} is the release time of operation O_{ijkl}
- t_{ijkl} is the earliest time at which O_{ijkl} can start
- T_{ijkl} is the latest time at which O_{ijkl} can start
- p_{ijkl} is the processing time of the operation O_{ijkl}
- C_{ijkl} is the *k* operation completion time from job *j*, level *l* on the machine *i*
- Each operation O_{ijkl} must be processed on one machine of the set M_i, where p_{ijkl} is the processing time of operation O_{ijkl} on machine M_i.
- The existence of operations on the same job, on different parts and components, processed simultaneously on

different machines, followed by components assembly operations (multi-level jobs).

C. Machines

- The shop consists of a set of machines $M_1,\ldots,M_n$.
- A machine can process more than one operation of the same job (recirculation).
- The existence of alternative machines, identical or not.

III. Multi-Agent Systems

Supply chains are evolving to more coupled organizations like virtual enterprises, though maintaining the single entities autonomy, adaptability and dynamism properties. Such organizations imply organizational and technological developments through agility, distribution, decentralization, reactivity and flexibility. New organizational and technological paradigms are needed in order to reply to the modern manufacturing systems challenges [12]. Some recent trends in manufacturing in particular and business in general, lead to new approaches regarding the organization and software architecture, mainly adopting distributed solutions. Multi-Agent Systems (MAS) is a promising approach for developing applications in complex domains.

A. MAS for Scheduling

Considering the complexity inherent to the manufacturing systems, dynamic scheduling is considered an excellent candidate for the application of agent-based technology. In many implementations of MAS systems for manufacturing scheduling, the agents model the resources of the system and the tasks scheduling is done in a distributed way by means of cooperation and coordination amongst agents [16][17]. There are also approaches that use a single agent for scheduling that defines the schedules that the resource agents will execute [10], [8]. When responding to disturbances, the distributed nature of multi-agent systems can also be a benefit to the rescheduling algorithm by involving only the agents directly affected, without disturbing the rest of the community that can continue with their work.

This fact incites researchers to explore new directions and Multi-Agent technology has been considered an important approach for developing industrial distributed systems[1-2]5[6][9][15][18][22-23].

B. Terms and Definitions

The concept of an agent has found application in a diverse range of sub-disciplines of information technology, including computer networks, software engineering, artificial intelligence, human-computer interaction, distributed and concurrency systems, mobile systems, computer-supported cooperative work, control systems, decision support, information retrieval and management, and electronic commerce. In practical developments, web services, for example, now offer fundamentally new wais of doing and independent software components interacting to provide valuable functionality.

The main term of Multi-Agent based computing is an Agent. However the definition of the term Agent has not common consent. In the last few years most authors agreed that this definition depends on the domain where agents are used. However there is a general consensus about its two main abstractions:

- An agent is a computational system that is situated in a dynamic environment and is capable of exhibiting autonomous and intelligent behaviour.
- An agent may have an environment that includes other agents. The community of interacting agents, as a whole, operates as a multi-agent system.

The most important common properties of computational agents are as follows [20][25]:

- act on behalf of their designer or the user they represent in order to meet a particular purpose.
- are autonomous in the sense that they control both their internal state and behaviour in the environment.
- exhibit some kind of intelligence, from applying fixed rules to reasoning, planning and learning capabilities.
- interact with their environment, and in a community, with other agents.
- are ideally adaptive, i.e., capable of tailoring their behaviour to the changes of the environment without the intervention of their designer.

Additional agent properties, characteristic in particular domains and applications are mobility (when an agent can transport itself to another environment to access remote resources or to meet other agents), genuineness (when it does not falsify its identity), credibility or trustworthiness (when it does not communicate false information wilfully) and sociality (when agents work in open operational environments hosting the execution of a multiplicity of agents, possibly belonging to different stakeholders (think, e.g., of agent-mediated marketplaces).

C. Coordination Aspects

In this section we will focus on the coordination, cooperation and negotiation terms, which are extensively mentioned in scientific research but whose definitions are often unclear and overlapping.

Most recent publications agree that the definition of MAS depends on the domain where agents are used and because of that MAS definitions abound. We believe that a solid and generic definition is offered by [17] that define a MAS as a system capable to "*solve complex problems in a distributed fashion without the need for each agent to know about the whole problem being* *solved*". This definition and many others imply the existence of an organizational model, or management mechanism, that allows for interaction, communication and achievement of objectives within the group of agents, regardless whether the system is comprised of self-interested, group oriented or both types of agents. A proper definition for such mechanism is given by the definition of the word coordination, particularly if one considers coordination applied to MAS. We can find several reasons on why agents need to be coordinated:

- **Chaos and anarchy prevention** – coordination is desirable because in distributed system anarchy is easy to implement,
- **Global restrictions** – agents must obey to a set of restrictions in order to be well succeeded,
- **Knowledge, resources and information is distributed** – agent can have different capacities and specializations, Alternatively agents can have different information fonts, resources, responsibilities and limitations.
- **Dependency between agents actions** – agents objectives are normally inter-dependent, needing that their activities be coordinated,
- **Efficiency** – autonomous agents can work in an independent way, but at the same time they have the necessity of coordinate their actions with others. The information of an agent can be sufficient for in group with another agent solve a problem quickly.

We think therefore that coordination in MAS is the protocol that ensures a consistent and efficient sociological functioning among any group of agents.

Cooperation and **negotiation** are the two main types of coordination protocols developed in the literature and implemented systems.

Cooperation is generally defined in the literature as the act of combining efforts, in order to accomplish a common objective that one autonomous agent alone cannot reach by itself. In other words, "*cooperation refers to a coordination protocol among no disputant agents*". Such protocol is suited when agents need to share tasks or results as a way to reach the system's objective.

On the other hand, "*negotiation is the coordination among competitive or simply self-interested agents*". Negotiation is then the process in which at least two agents negotiate through a protocol in order to accomplish an agreement. This protocol is suitable for systems made of agents that must pursuit their own individual goals but need interaction with other agents to achieve a satisfactory outcome. The buyer-seller situation is the best analogy to demonstrate this protocol.

D. Learning Aspects for MAS

A capable learning is essential in multi-agent systems because it allows saving lots of time and computer processing costs. A system like this helps the MAS increasing results, efficiency and flexibility because it will recognize patterns in scheduling systems. If a pattern is discovered the system will use the scheduling plan made in the first time that this pattern was generated. In Multi-Agent Systems a learning mechanism can be applied in each autonomous agent, and if necessary passed to the other agents present in the system.

Learning in Multi-agent systems is a challenging but relatively unexplored problem, so does Optimization. Optimization in such environments must deal with dynamism. Agents could change its behavior because they are learning or because they are optimizing.

Learning strategies could improve system performance, giving agents the capability to learn which BIT is best for solving particular problem, or which parameterization is more adequate for.

We will consider two different approaches to support agents learning: Learning from the history, once a request is sent to a resource community, is first filtered by the system to decide whether the request can be recognized as associated with a previous similar situation. In the learning from the future approach, the overall behavior of the system is constantly monitored and future emergent behaviors can be identified through simulation forecasting mechanisms.

A more detailed discussion on learning in agent-based manufacturing systems can be found in [14][21]

IV. Multi-Agent System for Distributed Manufacturing Scheduling with Genetic Algorithms and Tabu Search

A. MASDScheGATS Architecture

Distributed environment approaches are important in order to improve scheduling systems flexibility and capacity to react to unpredictable events. It is accepted that new generations of manufacturing facilities, with increasing specialization and integration, add more problematic challenges to scheduling systems. For that reason, issues like robustness, regeneration capacities and efficiency are currently critical elements in the design of manufacturing scheduling system and encouraged the development of new architectures and solutions, leveraging the MAS research results.

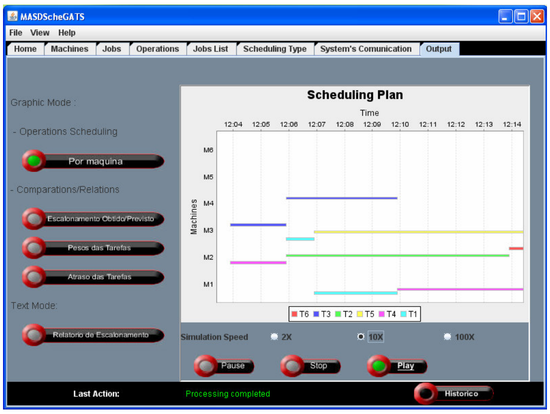

Figure 1 - MASDScheGATS System

The work described in this paper is a system where a community of distributed, autonomous, cooperating and asynchronously communicating machines tries to solve scheduling problems.

The main purpose of MASDScheGATS (Multi-Agent System for Distributed Manufacturing Scheduling with Genetic Algorithms and Tabu Search) is to decompose the scheduling problem into a series of Single Machine Scheduling Problems (SMSP)[10],[11] and create a Multi-Agent system where each agent represents a resource (Machine Agents) in a Manufacturing System.

Each Machine Agent must be able:

- to find an optimal or near optimal local solution trough Genetic Algorithms or Tabu Search meta-heuristics.
- to deal with system dynamism (new jobs arriving, cancelled jobs, changing jobs attributes, etc).
- to change/adapt the parameters of the basic algorithm according to the current situation.
- to switch from one Meta-Heuristic algorithm to another.
- to cooperate with other agents.

The proposed architecture is based on three different types of agents. In order to allow a seamless communication with the user, a User Interface Agent is implemented. This agent, apart from being responsible for the user interface, will generate the necessary Task Agents dynamically according to the number of tasks that comprise the scheduling problem and assign each task to the respective Task Agent.

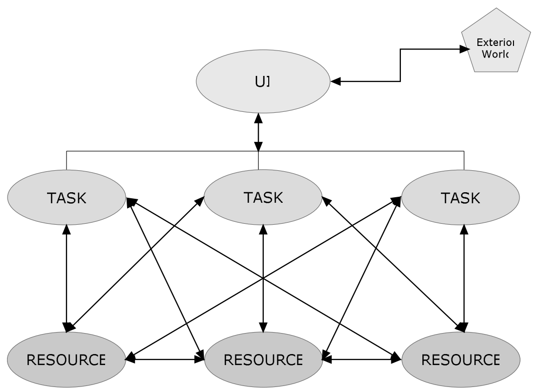

Figure 2. MASDScheGATS System Architecture

The Task Agent will process the necessary information regarding the task. That is to say that this agent will be responsible for the generation of the earliest and latest processing times, the verification of feasible schedules and identification of constraint conflicts on each task and the decision on which Machine Agent is responsible for solving a specific conflict. Finally, the Machine Agent is responsible for the scheduling of the operations that require processing in the machine supervised by the agent. This agent will implement meta-heuristic and local search procedures in order to find best possible operation schedules and will communicate those solutions to the Task Agent for later feasibility check (Figure 2).

B. Coordination Mechanism

Once the Machine Agents find their respective best local solution to the set of assigned operations, it is likely that the assembly of such solutions in a final plan will not establish a feasible schedule. The reason for this situation derives from the fact that each Machine Agent does not take into account, due to the concurrent procedure of local searching, the plans of other agents with which it has inter-agent constraints. It is therefore necessary a subsequent coordination mechanism so that a global feasible schedule is attained whilst minimizing the adjustments to the initial local solutions [11].

In a real manufacturing system a product is produced, step by step, passing on several machines. In each machine it will be performed at least one operation (job) of the process plan. In our approach we have one agent for each machine. However, if we join solutions obtained by our machine agents we will observe that, some times, they will not be feasible. In fact, if operation op1 (in machine m1) precedes operation Op2 (in machine m2) and Op2 precedes Op3 (in machine m3) in a manufacturing process, it is not guaranteed that the initial time for Op2 in m2 will be after the end of Op1 in m1 nor that the end of Op2 in m2 will be before the start of Op3 in m3.

Two possible approaches, to deal with this problem, could be used. In the first, the MASDScheGATS system waits for the solutions obtained by the machine agents and then apply a repair mechanism to shift some operations in the generated schedules till a feasible solution is obtained (Repair Approach). In the second, a coordination mechanism is established between related agents in the process, in order to interact with each other to pursuit common objective through cooperation. These coordination mechanisms are prepared to accept agents subjected to dynamism (new jobs arriving, cancelled jobs, changing jobs attributes). The latter approach is the one implemented in the proposed system.

C. Meta-Heuristics Self-Configuration Properties

Generally, self-organization can be defined as the process by which systems tend to reach a particular objective with no external interference. All the mechanisms dictating its behavior is internal to the system e.g. are autonomous. This field of research has received much attention through Autonomic Computing paradigm.

In this paper we consider that Meta-Heuristics self-parameterization could permit a better adaptation to the dynamic situation being considered. The idea is that each agent adopts and provides self-parameterization in accordance with the problem being solved: the method and/or parameters can change in run-time, the agents can use different MH according with problem characteristics, etc.

Meta-Heuristics can be adapted to deal with dynamic problems, reusing and changing solutions/populations in accordance with the dynamism. We will use the Dynamic Adaptation Mechanisms defined in [10][11], which includes a method for neighborhood regeneration under dynamic environments, increasing or decreasing it according to new job arrivals or cancellations.

The considered initial parameters for Meta-Heuristics follow the parameterization study described in [10].

V. Conclusions and Future Work

We believe that coordination mechanisms like they are described in this paper are the cornerstone to obtain a Multi-Agent System capable of solving dynamic and distributed scheduling problems with better results and less mathematical and computational costs. This mechanism must assume an even more relevant position in the whole MAS, because in our opinion does not make much sense try to solve inconsistency of generated plans using complex mathematical algorithms in a system like this, that have like biggest advantage in the distributed resolution of problems.

Work still to be done includes the testing of the proposed system and negotiation mechanisms under dynamic environments subject to several random perturbations. We realize, however, that this is not an easy task because it is difficult to find test problems and computational results for the considered dynamic environment where the jobs to be processed have release dates, due dates and different job assembly levels (parallel/concurrent operations) as defined in section II.

In future MASDScheGATS will provide a learning mechanism to allow the recognition of manufacturing scheduling in order to diminish computer processing costs and to turn the system more affordable in overall.

ACKNOWLEDGEMENTS

The authors would like to acknowledge FCT, FEDER, POCTI, POCI for their support to R&D Projects and GECAD Unit.

References

[1] Babiceanu, R.F., Chen, F.F., 2006, Development and Applications of Holonic Manufacturing Systems: A Survey, Journal of Intelligent Manufacturing, 17: 111-131.

[2] Caridi, M., Cavalieri, S., 2004, Multiagent Systems in Production Planning and Control: An Overview, Production Planning and Control, 15/2: 106-118.

[3] Cowling, P. and Johansson, M. Real time information for effective dynamic scheduling. European J. of Operat. Research,139 (2), 2002, 230-244.

[4] Horling, Brian, Lesser, Victor, 2005. A Survey of Multi-Agent Organizational Paradigms, University of Massachusets.

[5] Kádár, B.; Monostori, L.; Csáji, B., 2003, Adaptive Approaches to Increase the Performance of Production Control Systems, Proc. of 36th CIRP ISMS, June 3-5, Saarbrücken, Germany:305-312.

[6] Kim, B.I., Heragu, S.S., Graves, R.J., St Onge, A., 2003, A Hybrid Scheduling and Control System Architecture for Warehouse Management, IEEE Trans. on Robotics and Autom., 19/6: 991-100

[7] Lee, Y., Kumara, S., Chatterjee, K., 2003, Multiagent based Dynamic Resource Scheduling for Distributed Multiple Projects Using a Market Mechanism, J. of Intell. Manufact., 14/5: 471-484.

[8] Logie, S., Sabaz, D., and Gruver, W.A., Sliding Window Distributed Combinatorial Scheduling unsing JADE, IEEE International Conference on Systems, Man and Cybernetics, 2004.

[9] Lu, T., Yih, Y., 2001, An Agent-Based Production Control Framework for Multiple-Line Collaborative Manufacturing, International Journal of Production Research, 39/10: 2155-2176.

[10] Madureira, A. Meta-Heuristics Application to Scheduling in Dynamic Environments of Discrete Manufacturing. Ph.D. Thesis, University of Minho, Braga, Portugal, 2003 (in portuguese).

[11] Madureira, A., Ramos, C. and Silva, S. Toward Dynamic Scheduling Through Evolutionary Computing. In WSEAS Transactions on Systems. Issue 4. Volume 3, 2004, pp.1596-1604.

[12] MANUFUTURE High level group, MANUFUTURE A vision for 2020, Report of High level group, European Commission, November 2004.

[13] Márkus, A., Kis, T., Váncza, J., Monostori, L., 1996, A Market Approach to Holonic Manufacturing, Annals of the CIRP, 45/1: 433-436.

[14] Monostori, L., Márkus, A., Van Brussel, H., Westkämper, E., 1996, Machine Learning Approaches to Manufacturing, Annals of the CIRP, 45/2: 675-712.

[15] Monostori, L., Váncza, J., Kumara, S.R.T,(2006) Agent Based Systems for Manufacturing, CIRP Annals-Manufacturing Technology, Volume 55, 697-720.

[16] Nof, S.Y., Weill, R., 1992, Collaborative Coordination Control (CCC) of Distributed Multimachine Manufacturing, Annals of the CIRP, 41/1: 441-444.

[17] Nwana, H., Lee, L. and Jennings, N. Coordination in software agent systems, BT Technol J Vol 14 No 4 October, 1996.

[18] Papakostas, N., Mourtzis, D., Bechrakis, K., Chryssolouris, G., Doukas, D., Doyle, R., 1999, A Flexible Agent Based Framework for Manufacturing Decision-Making, Proc. of the 9th Flexible Autom. And Intell. Manufact. Conf.: 789-800.

[19] Russel, S., and Norvig ,P., Artificial Intelligence: A Modern Approach, Prentice Hall/Pearson Education International: Englewood Cliffs (NJ), (2nd Ed), 2003.

[20] Shen, W., Hao, Q., Yoon, J. and Norrie, D.H., 2006, Applications of Agent Systems in Intelligent Manufacturing: An Updated Review, Advanced Engineering Informatics, Volume 20, Issue 4, October 2006, Pages 415-431.

[21] Shen, W., Maturana, F. and Norrie, D.H. (1998b). Learning in Agent-Based Manufacturing Systems. In Proceedings of AI & Manufacturing Research Planning Workshop, Albuquerque, NM, The AAAI Press, pp. 177-183.

[22] Valckenaers, P., Van Brussel, H., 2005, Holonic Manufacturing Execution Systems, Annals of the CIRP, 54/1: 427-432.

[23] Váncza, J., Márkus, A., 2000, An Agent Model for Incentive-based Production Scheduling, Computers in Industry, 43/2: 173–187.

[24] Weiss, G., 1999, Multi Agent Systems – A Modern Approach to Distributed Artificial Intelligence. The MIT Press.