Agents-based Interaction Protocols and Topologies in Manufacturing Task Allocation

Mohammad Owliya
School of Mechanical Engineering
University of Birmingham
Edgbaston, Birmingham
B15 2TT, United Kingdom
owliya@gmail.com

Mozafar Saadat
School of Mechanical Engineering
University of Birmingham
Edgbaston, Birmingham
B15 2TT, United Kingdom
m.saadat@bham.ac.uk

Mahbod Goharian
School of Computer Science
University of Birmingham,
Edgbaston, Birmingham
B15 2TT, United Kingdom
mahbod.goharian@gmail.com

Rachid Anane
Faculty of Engineering and Computing
Coventry University
Priory Street, Coventry
CV1 5FB, United Kingdom
r.anane@cs.bham.ac.uk

Abstract - *The focus of this paper is on interaction protocols and topologies of multi-agent systems for task allocation in manufacturing applications. Resource agents in manufacturing are members of a network whose possible logical topologies and governing interaction protocol influence the scheduling and control in the multi-agent system. Four models are identified in the paper, each having specific rules and characteristics for scheduling and task allocation. The models use either a standard interaction method such as Contract-Net Protocol (CNP), or a different method proposed in this research. A Java-based multi-agent system was developed to simulate different scenarios of task allocation and to compare the four models in terms of performance indicators. Data from an industrial case study involving a manufacturing shop floor was used to evaluate the performance of the models. The results indicate meaningful differences between the four models, and highlight the performance potential of a proposed task allocation model.*

Keywords: Task Allocation, Manufacturing Scheduling, Agent-based System, Topology, Contract-Net Protocol

1. Introduction

Scheduling is a major decision making process in many engineering systems. In general, its function is to determine the start and the end of activities in a system, and is essentially the process of allocation of tasks to the resources over a period of time. Many methodologies and techniques have been proposed in a variety of application areas during the recent decades. Through the emergence of a number of new manufacturing systems attention has been drawn to methods that can support dynamic and decentralized task allocation. Agent-based systems, a manifestation of Distributed Artificial Intelligence (DAI), have many applications in engineering fields and offer a framework that is able to meet such a demand. Agents are computational systems capable of acting autonomously and proactively in a dynamic environment in order to achieve a designated goal. They posses some form of social ability and interact with each other and with their environment. A community or network of interacting agents is referred to as Multi-Agent System (MAS). In a MAS various architectures or organizational patterns may exist, each leading to different roles for agents, different rules and relations between them, and hence different ways of achieving objectives [1].

The relations between agents are regulated by specific rules and protocols for interaction and are correlated to the topology of the network of agents [2]. Together, they may characterize solution models for the task allocation problem. More specifically, different basic topologies and corresponding protocols can be considered for task allocation in MAS. Given a specific protocol and topology, a fundamental issue is *how they compare in terms of manufacturing performance.* Little work has been reported that deals explicitly with the topology of multi-agent systems in manufacturing. This paper presents an attempt to address specifically this issue. In the process, it introduces new models for agent-based manufacturing job allocation, hence leading to the enhancement of relevant methods in shop floor automation. Four task allocation models have been considered in this research, each with a basic topology of MAS network and the corresponding rules and protocols. An agent-based simulation system has been developed to support experimental work and to facilitate comparisons of the performance of the models in an industrial case study involving a large turbine manufacturing shopfloor.

2. Literature review

Manufacturing scheduling is a task allocation problem, which seeks an optimised solution in order to satisfy specific criteria. An optimised allocation plan might however become obsolete by a simple change or disturbance in the real environment [3]. Thus, dynamic task allocation and the techniques that support it have been the subject of intensive research. In their survey on dynamic scheduling, Ouelhadj and Petrovic [4] name heuristics, meta-heuristics, knowledge-based systems, fuzzy logic, neural networks, hybrid techniques, and multi-agent systems as instances of such methods. On the other hand, a top-down centralised system for task allocation can cause rigidity and limit problem solving ability in real world [3], [5], although centralisation can provide a consistent global view of the state of the system [4].

978-1-4244-8196-5/10/$26.00 ©2010 IEEE
DOI 10.1109/SOSE.2010.373

A distributed approach to control and scheduling attempts to address the inflexibility of hierarchical systems.

Agent-based systems are able to accommodate both dynamic and distributed scheduling, and as such have been investigated in many research and development projects in manufacturing systems [1]. In a task allocation agent-based system, autonomous task and resource agents interact with one another in order to create a schedule, which aims at performance maximization by fulfilling the tasks as quickly as possible [6]. According to [7], in the development of an agent-based task allocation system, four main issues must be dealt with: 1) representation of the physical world entities by agents with an explicit relationship between the entity and relevant agent, 2) system architecture and topology of the agents' network 3) interaction protocols which are closely related to the topology, and 4) decision scheme for individual agents which is not independent of the interaction protocols. The focus of this paper is on the issues of topology and interaction protocol of agent-based systems.

Market mechanisms have offered resilient protocols for task allocation within agent-based systems that are dominant in this field [1], [8]. Variations of the contract net protocol (CNP) are the most common, although other methods such as auction-based, pricing-based, yellow-page-based, game-theory-based have been applied [9]. Monostori *et al.* [1] argue that market mechanisms have a number of drawbacks; for instance, it is hard to guarantee the avoidance of extreme situations. The interaction protocols are not independent of topology of the agents' network [2]. Together, they determine the performance of task allocation. In a MAS, as a network of agents with characteristic properties, the agents' interaction, collaboration, and sharing of data and knowledge depend on the system topology [10].

Zhu *et al.* [10], [11] have studied and compared three topologies in an application of multi-agent systems: Web-like, Star-like, and Grid-like. Relevant criteria included communication between the agents, dependency to complete tasks, and sharing knowledge/data. In another paper Zhu [2] assesses the advantages and drawbacks of the three topologies in terms of autonomy, adaptation, scalability, and efficiency of cooperation, and discusses their applicability to different environments. He argues that proper topology leads to better behaviour of the MAS, and has reported the results of evaluating many MAS systems to find out which topology is more popular in which category of application. As far as task allocation is concerned, the research shows that Star-like and Web-like topologies are prevalent.

Hsieh [12] proposes the formation of networks based on variations of the CNP in a holonic manufacturing research, where the order holon agent asks for proposal, and the resource holon agents bid for execution. A product holon has been added as an intermediary agent in the network. This configuration does not have a generic application in manufacturing although it creates a chain implementation of the CNP in a network of star-like clusters. The topology of the network is not fixed, and evolves over time in response to changes in the state of the tasks and resources. However, the basic logical topology, which depends on the negotiation and allocation protocol, remains the same.

The coordination and re-adjustment of the agents, following changes in the underlying network topology of MAS has also been a subject of study [6]. Here, the proposed algorithms for multi-agent task/resource negotiation and allocation concern the distribution of agents within the network, as well as the topology. The research is, however, related to geographical distribution and relocation of agents, which causes alteration to the network topology, rather than logical topologies and interaction protocols in multi-agent systems. In a another investigation [13], the effect of network topology in agent-based manufacturing has been reported and an infrastructure for co-ordination of agents in a network-based manufacturing system presented.

3. Agent interaction protocols and topologies in task allocation

A manufacturing system, modelled by agents, is a loosely coupled network of communicating and cooperating production entities [1]. In such a network, the connection method between these entities, together with their interaction rules, significantly affect the functionality of the system. As discussed in Section II, research efforts in dynamic distributed scheduling have widely used market mechanisms, particularly the standard Contract Net Protocol (CNP) or its variations for the allocation of tasks to resources. Smith [14] first proposed CNP as a simple and efficient tool, which has been later standardised by the Foundation for Intelligent Physical Agents [15]. The following steps are a summary of the CNP task allocation process among the contractor agents (known also as participants) by manager agent (also known as initiator) [14]:

1. Task Announcement by initiator
2. Task announcement processing by participant
3. Bidding by participant
4. Bid processing by initiator, and awarding the contract
5. Contract processing, reporting result, and termination

In its original form, the CNP is such that a participant awarded with a contract cannot bid for a new task until it has completed its current task. In a modified version suggested in this work, a participant can bid for a new task prior to completing the current task. This is deemed to be more compatible with real manufacturing environment. Regardless of these variations, it is evident that in a CNP type of interaction there is a central initiator, surrounded by participants. This provides the basis for a star-like topology in a network of agents. CNP in a star topology is a widely used model for task allocation.

In this paper the concept of star model in conjunction with CNP is first discussed. Then, a peer-to-peer (P2P) variation of the CNP-based star model is presented as the second model. The ring topology with a proposed protocol is presented as the third model. The ring model is finally modified and presented as the fourth model to incorporate features of peer-to-peer interaction, but with a similar protocol as the ring model. The structure and protocol description of these four models are given below.

In the development of the four models the following common manufacturing strategies are used regardless of the topology under consideration:

- In a series of tasks to be allocated, those in the 'critical path' have the highest priority, with the next priority given to the tasks with fewer margins left to due time.
- Task dependency is to be observed – i.e. all pre-requisite tasks are carried out prior to the original task.
- Resource agents cannot simultaneously operate on more than one task.

3.1. Star model

CNP in its simplest form, where contractors are only connected to the central manager, forms a Star network. Figure 1 illustrates CNP-based interaction in such a network, where logical topology and interactions of its members occur regardless of the physical arrangement of its resources. Messages are exchanged in this interaction protocol in the form of FIPA-ACL (Agent Communications Language) [15].

The Star model of task allocation should now be augmented with some specific decision-making rules. In this research, the task allocation model using CNP in a Star topology follows the laws of typical market-based scheduling mechanisms such as suggested in [16], [17]. Additionally, the following rules are suggested and used in this research when bids are received from the contractors and the manager begins to process them in order to arrive at a decision to award contracts:

1. IF more than one resource can complete a task before its due time, THEN the resource with the lowest cost will be chosen. In such a case, IF more than one resource has the same lowest cost, THEN the resource that can start earlier has priority.
2. IF only one resource can complete a task before its due time, THEN it is chosen without any cost consideration.
3. IF no resource can complete a task before its due time, THEN the resource that can start earlier has priority, without considering the resource cost.

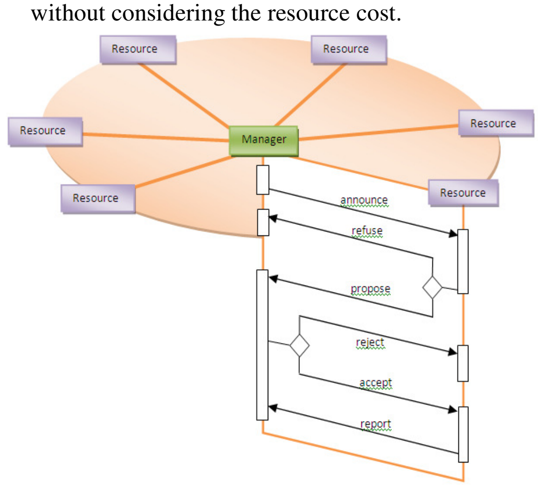


Figure 1. Star topology formed with CNP-based interactions simplified from FIPA [15]

Any alteration in the network architecture and interactions between agents, or even behaviour of the agents could result in a different task allocation outcome. For this reason, other types of topologies are suggested below with a particular focus on manufacturing applications.

3.2. Peer-to-Peer model

The CNP as seen in the Star model is flexible, but the model is still too centralized with only one manager. All resource agents in the Star model can be connected to one another to produce a Peer-to-Peer (P2P) model, as shown in Figure 2. Here no single central manager or broker exists. This means that any resource can itself be a manager as well. In contrast with the Star model, such a P2P model is more robust due to redundancy of autonomous Resource/Manager (R/M) agents [2]. When an agent plays the role of Manager (R/M) it interacts with all available Resources (R/M) similar to the Star model. To ensure central coordination among the managers and global knowledge in the system, a higher-level supervisory agent is added to this P2P model. This (a) makes it more centralized, and (b) adds stability. As a first step, the supervisor groups the tasks in terms of their dependencies and sends each group to an R/M agent. Depending on the number and dependency of tasks, each R/M agent could receive zero, one, or more than one task groups. Then each manager agent that possesses at least one task group uses CNP in conjunction with the relevant rules mentioned in the Star model for negotiation and task allocation to all other R/M agents. The final schedule is then produced.

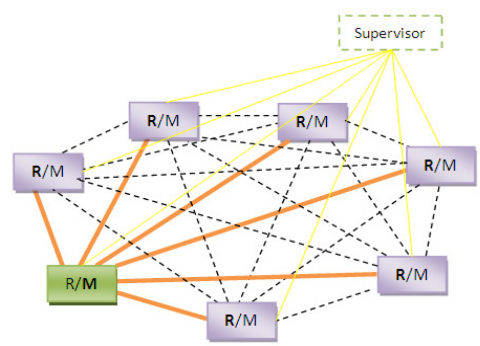


Figure 2. P2P topology using CNP

3.3. Ring model

Resource agents could be arranged to form a Ring as illustrated in Figure 3. There would be no manager agent as in the previous cases. A higher-level supervisory agent is in charge of the coordination among agents similar to the P2P model. The main issue with the classical ring topology is that the failure of one network member brings the entire network to a halt. In this model, however, the role of supervisor precludes such a situation, in addition to offering other benefits for P2P model, as already mentioned. Upon the arrival of a manufacturing order (set of tasks), a table of tasks to include all their specifications is created. The tasks are sorted in the table according to their priority, which is determined by pre-defined rules and user inputs. The supervisor agent successively circulates and monitors the task table among the resource agents. First the resource agent with lowest operating cost receives the table. In Figure 3, resource-1 is the cheapest and is the first agent to receive the table. The agent that holds the table reviews all the remaining tasks in it, which are already sorted by the highest priority, and identifies the ones that match its technical capability. From the identified tasks, it then picks those that can perform within their due time, and adds them to its local schedule (selfish and greedy behaviour). Then, it begins to execute the task with the highest priority.

The resource agent leaves a proposal for the tasks that match its technical capability, but is unable to meet their due time requirement. On receiving the table, the next agent, if also unable to satisfy the due time, compares the left proposal with its own, and decides which one should be kept in the task table for further circulation (the worse proposal will be omitted). Each resource agent has its local schedule in which the IDs together with all other attributes of the tasks undertaken, or the tasks it has offered a proposal for, are recorded. The table will be passed on to the next agents until all tasks are allocated. This is a new approach to the task allocation problem using ring topology which drastically differs from CNP, although it still uses bidding mechanism to a limited extent.

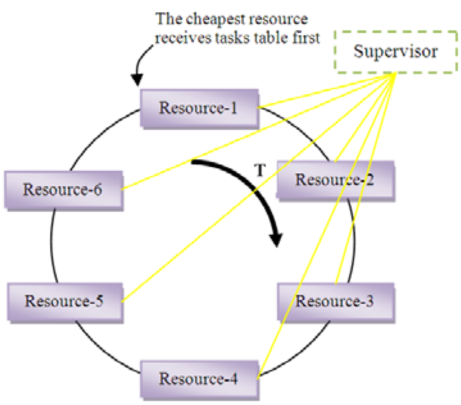


Figure 3. Ring topology

3.4. Modified Ring model

The Ring topology may be improved by employing the peer-to-peer mode of interaction to form a Modified Ring model. The structure and the basic protocol are similar to the Ring topology, but the agents can interact with one another through ACL messages in special situations, as shown in Figure 4. For instance, when an agent has replaced the previous proposal by its own, it will notify the agent, which had set the previous proposal, to update its local schedule.

4. Simulation system

A comparative evaluation of the conceptual models presented in the previous section needs an appropriate simulation tool. Agent-based simulation is considered as the most suitable means of MAS validation [18].

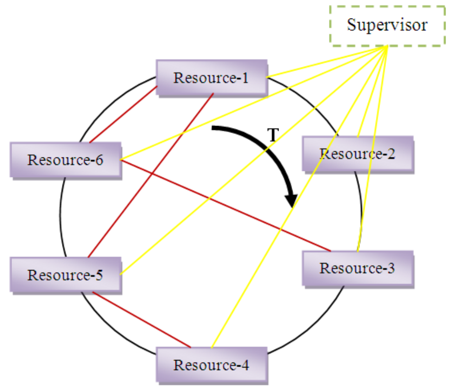


Figure 4. Modified Ring model

The simulation system is required to take a set of tasks, as well as available resources, simulate them visually and monitor the execution of tasks under the four different agent-based models of task. The system should facilitate the scheduling of the tasks; calculate several performance measures (time, cost, resource utilisation as discussed in the next section), and present the resulting task allocation schedule and performance parameters for each model. The user should be able to compare the four models after each simulation run. In the simulation environment, the resource agents have two components, one for decision-making and scheduling issues, and the other for operation (task execution). The two parts operate in parallel, allowing the resource agent to execute the tasks at the same time as it is negotiating or making decision. This makes it possible to implement the modified version of CNP as discussed in Section III. Another type of agent acts as supervisor or manager as required. Such agents have a rule-based inference and control function only, without any operational part.

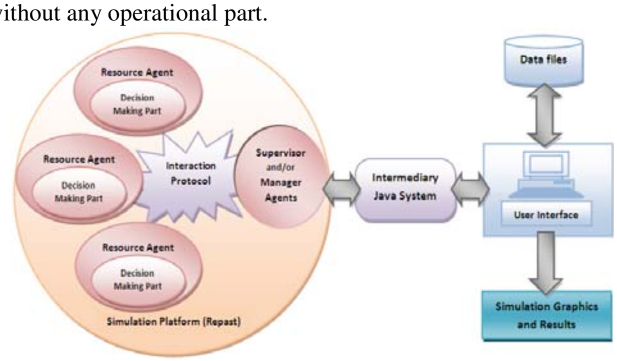


Figure 5. System architecture

The simulation tool was developed in Java and Repast. Java has many advantages; less effort is needed when Java-based agent development platforms are used [19]. Repast is one of the most complete Java-based simulation platforms with good execution speed, although it has some deficiencies in other aspects such as documentation [20]. Repast has a unit of time called ‘tick’ for simulating discrete events – an important feature that will be used in the evaluation of the proposed models. The overall system architecture is shown in Figure 5, in which there is a simulation platform consisting of agents in a Repast environment. Other components acting in conjunction with Repast are also shown in this figure.

Tasks and resources data are organised in data files, which are accessed through a user interface. The user has options in the interface data input, as shown in Figure 6. Monitoring the simulation process is done by the graphical facilities embedded in Repast. Figure 7 gives a snapshot of a sample run.

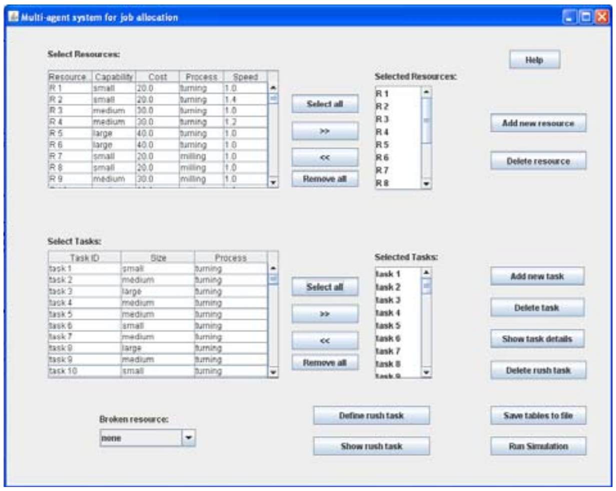


Figure 6. User interface for data input, showing a case of shop floor task allocation

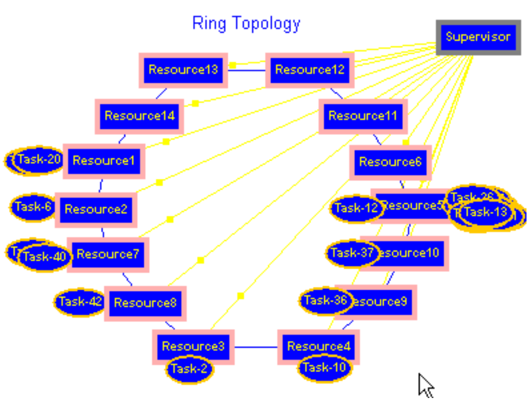


Figure 7. A snapshot of simulation

The experimentation and performance measurements of the four models are presented in the following section.

5. Evaluation of the models

In order to conduct the task allocation experiments and to compare performance of the models described in Section 4, the following quantitative parameters are calculated from simulations output data as performance indicators in this research:

Total time: This is the time elapsed to complete a manufacturing order (set of tasks), and contains any time spent on scheduling and operations until the last resource finishes the last task.

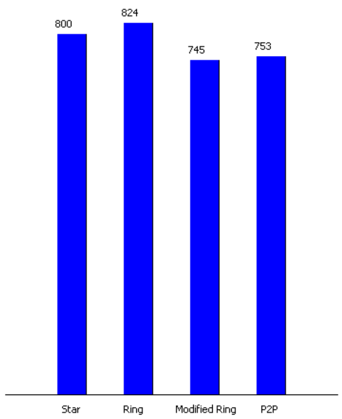
Costs: This consists of three major cost elements of the resources (machines). The first element is the cost when a machine is *busy* with a task. This is calculated by the rate of machine occupation, and depends on depreciation and running costs of each machine and the duration of operations including set-up times. The second element is *penalty* cost when a task passes its due time. The rate of penalty for each task is defined in the manufacturing order. Penalty cost is also an indication of tardiness. The third element is related to *idleness* (i.e. non-operating) status of machines.

Utilisation: Defined as the percentage of processing time against the total order execution time. It will be indicated by busy/idle percentage of the machines.

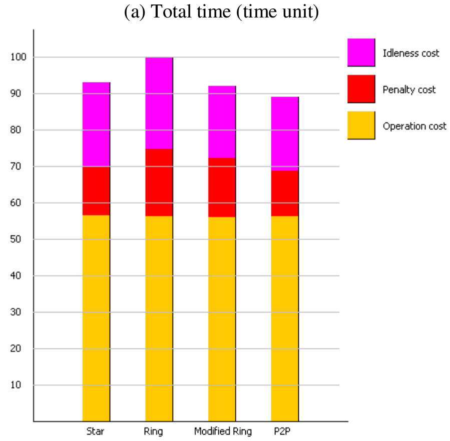
Manufacturing data from a turbine production company provides the basis for a case study used to support the industrial evaluation of this work. Incoming material to the shop floor are cast, forged, or welded parts that are to be machined with CNC machine tools, pre-assembled in some stages, and finally dispatched to assemble the final product. A manufacturing order is received at the workshop to make one or more product either in a make-to-order, or make-to-stock fashion. The order used in this case involves 47 tasks. These tasks, including machining processes of the turbine parts or parts assembly, are specified with their earliest start time, standard operation time, due time, penalty charge, operation type, and operation sequence. Resources include CNC machine tools for turning, milling and boring operations, as well as assembly stations. They are of different sizes, operating costs, and capabilities. The experiments included over 50 tests, covering four different task conditions - (1): pre-requisite tasks, penalty costs, and original due times are all in place, (2): as in (1) but with no penalty costs, (3): as in (1) but with extended due times for tasks, and (4): as in (1) but with no pre-requisite tasks.

An illustration of the results is presented in Figure 8, where the performance of each model is shown with all the three input variables in place as in task condition (1) above. Figure 8(a) shows the total time, and indicates that the Modified Ring model offers the shortest time, while the Ring model offers the longest time taken to complete the order. The three cost components are presented as percentages in Figure 8(b). In this case the P2P model has the total minimum cost. It is followed by the Modified Ring, since the latter is associated with higher penalties. In the case where there are no penalty charges for passing due times, the penalty part in the cost bar chart is removed, and the Modified Ring generates the lowest cost out of the four models. Utilisation percentages are given in Figure 8(c). The graphs show that the Modified Ring model gives the best utilization at 52.8% busy time, closely followed by the P2P model at 51% busy time.

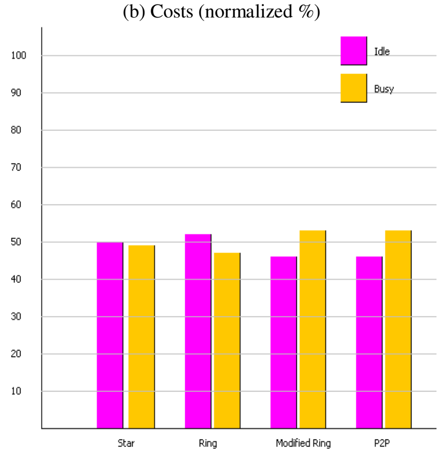
In the manufacturing case study the experimental results indicate that the Modified Ring and the P2P models offer more potential for optimal performance. In approximately 60% of the tests conducted, they display better performance compared to the Star and the Ring models. This is mainly due to the peer-to-peer interaction capability of the Modified Ring and the P2P, which allows them to fully exploit the decentralised architecture of agent-based systems. The results indicate that enhanced protocol and topology can have an impact on the performance of a system.



(a) Total time (time unit)



(b) Costs (normalized %)



(c) Utilisation (%)

Figure 8. Results of a real manufacturing case study

6. Conclusions

In this paper a number of agent-based models for task allocation in manufacturing shop floors were presented and compared through using a developed Java-based simulation software as the test platform. Experiments were conducted using real manufacturing data to test the performance of these models. Lead-time, cost, and resource utilisation were used as the performance criteria in this research. The results show that in most cases the proposed Modified Ring and CNP-based P2P models give superior performance output compared to the Star and Ring models.

References

[1] L. Monostori, J. Váncza, S.R.T. Kumara, "Agent-based systems for manufacturing", *CIRP Annals - Manufacturing Technology*, 55(2), 697-720, 2006.

[2] Q. Zhu, Q, "Topologies of agents interactions in knowledge intensive multi-agent systems for networked information services", *Advanced Engineering Informatics* 20(1), 31–45, 2006.

[3] W. Shen, L. Wang, Q. Hao, Agent-Based Distributed Manufacturing Process Planning and Scheduling: A State-of-the-Art Survey, *IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews*, 36(4), 563-577, 2006.

[4] D. Ouelhadj, S. Petrovic, "A survey of dynamic scheduling in manufacturing systems", *Journal of Scheduling*, 12(4), 417-431, 2009.

[5] A. Jones, L.C. Rabelo, "Survey of job shop scheduling techniques", NISTIR-National Institute of Standards and Technology, Gaithersburg, MD, USA, available online at: http://www.mel.nist.gov/msidlibrary/doc/luis.pdf [accessed 19/01/2010], 1998.

[6] Y.C. Jiang, J.C. Jiang, "A multi-agent coordination model for the variation of underlying network topology", *Expert systems with applications*, 29(2), 372-382, 2005.

[7] W. Shen, "Distributed Manufacturing Scheduling Using Intelligent Agents", *IEEE Intelligent Systems*, 17(1), 88-94, 2002.

[8] F. Versteegh, M.A. Salido, A. Giret, "A holonic architecture for the global road transportation system", *Journal of Intelligent Manufacturing*, 21(1), 133-144, 2010.

[9] D. Veeramani, K.J. Wang, "Bid construction scheme for job flow time reduction in auction-based fully-distributed manufacturing systems", *Int. J. Advanced Manufacturing Technology*, 28(5-6), 541-550, 2006.

[10] Q. Zhu, S.L. Aldridge, T.N. Resha, "Hierarchical Collective Agent Network (HCAN) for efficient fusion and management of multiple networked sensors", *Information Fusion*, 8(3), 266-280, 2007.

[11] Q. Zhu, J.D. Hicks, JP.V. Petrov, A.D. Stoyen, "The topologies of knowledge intensive multi-agents cooperation", Proceedings of the IEEE Knowledge Intensive Multi-Agent Systems Conference, Cambridge MA, USA, 741–746, 2003.

[12] F.S. Hsieh, "Deadlock free task distribution and resource allocation for holonic manufacturing systems based on multi-agent framework", IEEE International Conference on Systems, Man, and Cybernetics, Tucson, AZ, USA, Vol.4, 2463-2468, 2001.

[13] F.T.S. Chan, R. Swarnkar, M.K. Tiwari, M. K, "Infrastructure for co-ordination of multi-agents in a network–based manufacturing system", *Int. J. Advanced Manufacturing Technology*, 31, 1028–1033, 2007.

[14] R.G. Smith, "The Contract Net Protocol: High-level Communication and control in a distributed problem solver", *IEEE Transactions on computers*, Vol. C-29, No. 12, 1104-1113, 1980.

[15] Foundation for Intelligent Physical Agents, FIPA Contract Net Interaction Protocol Specification, available online at http://www.fipa.org [accessed on 19/12/2009], 2002.

[16] P. Leitao, F. Restivo, "A holonic approach to dynamic manufacturing scheduling", *Robotics and Computer-Integrated Manufacturing*, 24(5), 625-634, 2008.

[17] A. Markus, T. Kis Vancza, L. Monostori, "A Market Approach to Holonic Manufacturing", *Annals of the CIRP*, 45(1),433-436, 1996.

[18] I.J. Timm, T. Scholz, H. Fürstenau, H., "From Testing to Theorem Proving", *Multiagent Engineering - Theory and Application in Enterprises*, Springer, Berlin, part IV, chapter 8, 531-554, 2006.

[19] V.R. Komma, P.K. Jain, N.K. Mehta, "Agent-based Simulation of a Shop Floor Controller using Hybrid Communication Protocols", *Int J Simul Model*, 6(4), 206-217, 2007.

[20] S.F. Railsback, S.L. Lytinen, S.K. Jackson, "Agent-based Simulation Platforms: Review and Development Recommendations", *Simulation*, 82(9), 609-623, 2006.