Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial
ISSN: 1137-3601
revista@aepia.org
Asociación Española para la Inteligencia Artificial
España

Arauzo, José Alberto; Galán, José Manuel; Pajares, Javier; López-Paredes, Adolfo

Multi-agent technology for scheduling and control projects in multi-project environments. An Auction based approach

Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial, vol. 13, núm. 42, 2009, pp. 12-20
Asociación Española para la Inteligencia Artificial
Valencia, España

Inteligencia Artificial 42 (2009) 12-20
doi: 10.4114/ia.v13i42.1042

INTELIGENCIA ARTIFICIAL

http://erevista.aepia.org/

Multi-agent technology for scheduling and control projects in multi-project environments. An Auction based approach

José Alberto Arauzo[1], José Manuel Galán[2], Javier Pajares[1], Adolfo López-Paredes[1]
1 Social Systems Engineering Centre (INSISOC), University of Valladolid.
ETSII, Pº del Cauce s/n, 47011 Valladolid, Spain
{arauzo, pajares, adolfo}insisoc.org
2 INSISOC. University of Burgos,
Escuela Politécnica Superior, Edificio La Milanera, C/ Villadiego s/n, 09001, Burgos, Spain
jmgalan@ubu.es

Abstract Most existing methods for scheduling projects are based on centralized or hierarchical decision making using global models. So far, these methods have not produced the desired results, especially in highly dynamic multi-projects environments. In this study, we investigate a new method based on multi-agent systems and a combinatorial auction mechanism to allocate resources for the projects tasks. In combinatorial auction the bidders demand a combination of dependent goods with a single bid. We consider basically two types of agents, projects and resources. Projects are the bidders and resources are the goods. An auction mechanism based on Lagrangian-based decomposition is designed to achieve efficient allocation resources. Our approach allows manage some traditional aspects of multi-project environments: dynamic addition of resources or projects, changes of the resources capabilities, allocation flexibility, etc.

Keywords: multi-agent systems, auction based allocation resources, project scheduling and control.

Introduction

A multi-project environment can be described as a set of resources, which allows executing the portfolio projects. In order to execute the projects, the resources of the system need to perform a series of tasks that need to be perfectly coordinated. This coordination is achieved by means of a set of scheduling and control activities. These activities include aspects such as management program, data capture, resource synchronization, transport coordination, the supply of raw materials, execution of schedule, or supervision.

The traditional scheduling and control systems, based on hierarchical and centralized architectures, are not flexible enough to adapt themselves to the dynamism and complexity needed multi-project environments. For this reason successive proposals to improve the scheduling and control in a multi-project environment are appearing continually. The recent paradigm of Multi-agent Systems, which offers new techniques to face complex unsolved problems, can help to find promising solutions. As a matter of fact, it is currently a very active field of investigation

In a multi-project environment, managers have to prioritise and select the projects that will be added to the portfolio taking into account their expected profits, returns and their alignment with the firm strategy.

Priority ranking changes over time because of the addition of new interesting projects, changes in corporate strategy or simply, because of feedback information about individual project overruns affecting their expected returns. As priority changes, resources have to be reallocated to meet the requirements of the most interesting projects. At the same time, individual projects belonging to the portfolio have disruptions, delays and over-costs in their activities.

ISSN: 1988-3064(on-line)

Overruns and priority changes take place in parallel, and as a consequence, conflicts between projects emerge since individual projects compete for the same scarce resources. Multi-project management requires the development of *dynamic* resource constrained project scheduling systems.

Unfortunately, the research in this area has not converged to one solution or scheduling rule robust enough to hold in the general case [1]. Hans et al [7] review existing literature in hierarchical approaches and propose a generic project planning and control framework for helping management to choose between planning methods, depending on organisational issues. Kao et al [9] suggest an event-driven approach to develop a trade-off decision framework for scheduling and re-scheduling. Cohen et al. [4] analyse the performance of critical chain in multi-project environments.

In this paper, we propose a novel approach for online dynamic scheduling in multi-project environments. We use a multi-agent system where the agents are the resources and the projects; the price of the work done emerges from an auction.

Projects have scheduled work to be done by different resources. Resources are endowed with some capabilities (knowledge, work force, etc.) that are needed to do the work. Projects demand resources over time and resources offer their capabilities and time availability. There is an auction process, and the price of resource-time slots emerges endogenously as a result of supply and demand.

Our research differs from other previous works because we are not so concerned with the auction mechanism but with helping project managers to take their portfolio decisions about resources and about portfolio composition.

Under this powerful framework, we are able to analyse some hard problems in portfolio project management: under and over usage of resources; key resources and resources that should be added to the firm; the role of flexibility in human resources, that is, the role of people exhibiting several capabilities at the same time; or the addition or deletion of individual projects; priorities in project execution, etc.

The rest of this paper is organised as follows. First, we will discuss the role of agent based methodology in project scheduling. Then, in section 3, we will formally define the problem we want to face with, that is, the scheduling problem with project rejection. In section 4, we will explain the main features of our multi-agent approach to portfolio project management and we will see some of preliminary results in section 5. We will end with the main conclusions of our work.

Multi-agent systems and multi-project environments

Projects are characterized by complexity (they include many components and dependencies), uncertainty (about the availability of resources, task durations), dynamic behaviour (changes in the scope of the project, adding or removing unexpected tasks, re-scheduling processes) and are inherently distributed (each task may be completed by different resources or in different geographical locations).

In the case of a multi-project environment, each one of these features is severely intensified. More projects mean more complexity, more uncertainty sources, more complex dynamics, and problems more dispersed since often not only the tasks are distributed but the management of each project too.

Multiagent systems are precisely designed to deal with problems of these characteristics. As Jennings and Wooldridge [8] [20] point out, multiagent systems are suitable for problems having the following properties: *complexity*, *openness* (components of the system are not known in advance, can change over time, and are highly heterogeneous, dynamic in project management terms), with *dynamical and unknown environments changing over time* (uncertainty) and *ubiquity* (the activity is distributed over the complete structure).

Depending on the physical or organizational structures, many problems can be abstracted and managed using the multiagent paradigm as metaphor, e.g. [6] [16] [18]. In the particular case of multi-project systems, the agents can be abstracted as tasks, as resources, as project managers, etc. This design enables us to distribute the management system in elemental components directly identifiable in the target system, and hence giving the opportunity to create systems easier to design, to adapt and to maintain. Moreover, since the system is distributed according to its structure, any change in the structure can be easily translated to the management system.

This decentralized approach in project management has been used since the beginning of this decade[22], but it is in the last years when market based approaches [3] [23] are receiving a growing interest. Recently, Lee, Kumara and Chatterjee [14] [15] have proposed a multiagent based dynamic resource scheduling for distributed multiple projects using market mechanisms. Following the same research line Confessore et al [11] [5] propose an iterative combinatorial auction mechanism as coordination mechanism to resolve the same problem.

Other examples of agent based approaches in project management field can be found in the work by Kim and colleagues [10] [11] [12] [13] where software agents evaluate the impact of changes, simulate decisions, and give advice on behalf of the human subcontractors. In Wu and Kotak [21] and Cabac [2] a multiagente system enables project managers to accommodate effectively frequent engineering changes and other uncertainties while making an effective use of available distributed resources.

As underlined in the introduction, our work makes use of a market metaphor and an auction mechanism to help project managers to take portfolio decisions about resources and about portfolio composition. Our proposal integrates dynamically portfolio selection and operational issues as resource allocation. Therefore, it helps to fill the gap between project portfolio selection and operational issues.

The scheduling problem with project rejection

We define a multi-project scheduling problem with opportunity of project rejection as follows:

At any instant t there are I projects in the system, each one denoted by i. Each project is characterized by a value V_i, that can be interpreted as the revenue obtained for the project, a weight w_i representing the strategic importance given to the specific project, a desirable delivery date D_i, a limit delivery date D_i^* that cannot be exceeded, an arrival date of the project to the system, B_i , and a limit answer date R_i that represents the latest date to decide to reject the project.

The system is considered dynamic: while some projects are being developed other projects can be included or rejected in real time.

Each project i consists of J_i activities, each one denoted by ij where $i \in \{1, 2,\ldots, I\}$ and $j \in \{1, 2,\ldots, J_i\}$. They have associated a workload d_{ij}. A set of M resources is given. Each resource $m \in \{1, 2, 3\ldots M\}$ can just be assigned simultaneously to one activity.

A set Γ of K competences $\Gamma=\{k_1, k_2, \ldots k_K\}$ are necessary to complete the projects. Each resource is endowed with a given cost rate per unit of time C_m and a subset $\Gamma_m=\{k_{1m}, k_{2m}, \ldots k_{fm}\}$ of competences, each k_{lm} in a certain grade or ability to perform an activity e_{lm}, where $e_{lm} \in (0,1]$. Thus, each resource is defined by the set $\{C_m, H_m\}$ where $H_m=\{(k_l, e_{lm}),\ldots,(k_f, e_{fm})\}$ with $k_l,\ldots, k_f \in \Gamma$.

Every activity is associated with a competence c_{ij}. Any activity ij with a given c_{ij} can be performed by a resource m if $c_{ij} \in \Gamma_m$. The duration of the activity ij depends on the resource assigned to perform it according to d_{ij}/e_{ijm}, where e_{ijm} is the ability of the resource m in the competence c_{ij}.

In this first simplified model we assume that the activities of any project should be performed sequentially in the order defined by j and only one resource can be assigned to an activity. We also assume that once some resource has begun a task, the activity cannot be interrupted; the resource needs to finish it to be assigned to any other activity. The reassignment mechanism in real time takes place only for those tasks which have not begun.

We include explicitly the option of reassigning resources in real time when a new project arrives to the system. The activities can be assigned to any resource that has the specific competences to perform it.

In order to evaluate the overall efficiency of the system we calculate the average benefit obtained in a certain time interval T according to:

$$Efficiency = \frac{B_T}{T} = \frac{\sum_i (V_i - Cost(i))}{T} \qquad \textbf{(1)}$$

Where i are each one of the projects finished in T and *Cost(i)* is the cost to complete the i project. This cost has two components, the direct resource cost and the delay cost. The delay cost is calculated by means of a squared expression to penalize high delays. Also, this improves the auction performance.

$$Cost(i) = \sum_j C_{\bar{m}(j)} \cdot \frac{d_{ij}}{e_{ij\bar{m}}} + w_i \cdot (D_i - F_i)^2 \qquad \textbf{(2)}$$

The first addend corresponds to the direct resource cost to finish each activity j. $\bar{m}(j)$ denotes the resource selected to comply with activity j. The second addend is the delay cost associated to the project, where F_i is the real delivery date.

- The problem considers the decision to reject projects. This could happen in any of the following cases:
- The revenue obtained from the project does not compensate the costs.
- Our scheduling exceeds the D_i^* of the project.
- The impact on the scheduling of the rest of the projects is not acceptable. This may happen for two causes. First, if the new project obliges to delay a committed project beyond D_i^*, it will be rejected. If not, but the

inclusion of the new project increases the delay costs of the other projects more than the direct benefit obtained for the project, it will also be rejected. When a project is started, it can not be rejected. This implies that a new project with a high value can be rejected to avoid that the ongoing projects finish after D*.

A multi-agent system for online multi-project scheduling

Traditionally, scheduling problems have been solved offline and simultaneously by a centralized decision-maker that use a global optimisation model. Instead of the previous approach, we propose in our systems several decision-makers modelled as agents. We consider two kinds of agents: project agents and resource agents. At any moment, the system has as many project agents as projects are considered. Each agent represents a determined project characterized by its tasks, precedence relationships, due date, value, local programs and their execution state. Their goal is to look for contracts with resources that can perform the required activities and hence completing successfully the project. The system includes as many resource agents as resources are considered. They are defined by their competences, availability, local programs, cost rate, and pending task queue. Their objective is to execute tasks from project agents. Each agent will try to achieve its particular goal separately but overall the system work efficiently. The implementation has been carried out using JADE.

Every agent in the system can communicate with others by sending messages. The interaction mechanism is ruled by means of a combinatorial auction. The mechanism offers a theoretical basis for structuring message sequencing, bid evaluation, and price updating. This auction based multi-project scheduling approach is founded on Lagrangian Relaxation [17] [24]), a decomposition technique for mathematical programming problems. The Lagrangian Relaxation technique provides a mathematical model of the auction process and it guarantees us to obtain the equilibrium price.

In our distributed multi-project system, the decision-making is decentralized since each project creates its own schedule (local schedule). The auction mechanism ensures that local schedules are nearly compatible (several projects do not use the same resource simultaneously) and globally efficient.

Each unit of time (time slot) on each resource is modelled as a 'good' that can be sold in an auction where each resource acts as a seller. The number of sellers is equal to the number of resources in the system and each resource proposes a price for the time slots from the current time to the end of the scheduling horizon. The scheduling horizon changes dynamically by coinciding with the latest time slot that some project has asked at any moment.

Each project agent is a 'bidder' that participates in auctions by asking the resource agents for the set of time slots that are required to execute its pending tasks at the current time. The goal of each project agent is to find time slots of the resources to complete all the needed tasks while incurring in the minimum possible cost. This cost has two components, the sum of the price of the selected time slots and the delay cost (expression 3).

$$TC_i = \sum_{mt \in Z_i} p_{mt} + w_i \cdot (D_i - F_i)^2 \qquad \textbf{(3)}$$

Where: TC_i is the total cost of project, p_{mt} is the price of the time slot (t) of the resource (m), and Z_i the set of selected time slots for project (i)

To minimize the cost, the project agents use a dynamical programming algorithm where all possible combination of time slots and resources are considered[19]. In their decision, they take into account that only those resources endowed with the necessary competences can carry out a given activity. Moreover, the number of time slots necessary to complete the task is determined by the ability of the resource in the competence. Each project agent consider as scheduling horizon the time slot which goes from the current time to the limit delivery date (D_i^*). If some project agent cannot find a set of time slots in such a manner that it is possible to schedule tasks before D_i^* and with a smaller cost than its value (V_i), then it will not ask for any set of time slots. This implies that the project is unprofitable at the correspondent round of bidding and should be rejected. When the number of project tasks and the number of time slots are high, the resolution time of this algorithm can be high, however this algorithm is polinomical and the computational experiences show that it works properly in normal problems. Even if the size of the problem is very high, we can reduce the number of time slots. This would reduce the accuracy of the method, but would allow us to deal with big problems.

Each project agent determines the price charged for the time slots to reduce resource conflicts and maximize its revenue. For this purpose, a subgradient optimization algorithm is used to adjust prices at each round of bidding [19] [24]. By means of this algorithm the resource agents increase the price of the time slots where there is conflict (more than a project have asked for this time slot) and reduce the price of the time slots that have not been demanded. The process of price adjustment and bid calculation continues indefinitely. At each round of bidding the resource conflicts will be lower and lower.

At the first round of bidding, the time slots prices for the resource (m) are equal to the resource cost rate (C_m). At the rest of bidding round, the prices will be updated through the expression 4.

$$p_{mt}{}^{n+1} = \max\left\{ C_m \, , \, p_{mt}{}^{n} + \alpha^n \cdot g_{mt}^{n} \right\} \qquad \textbf{(4)}$$

Where:

$p_{mt}{}^{n+1}$: price of the time slot (t) of resource (m) at the round (n+1)

$p_{mt}{}^{n}$: price of the time slot (t) of resource (m) at the round (n)

α^n : step at the round (n). The calculation of this parameter can be seen in [18].

$g_{mt}^{n} = a_{mt}^{n} - 1$: subgradient, where a_{mt}^{n} is the number projects that ask for the slot (t) of resource (m).

As result of the described auction mechanism, project agents build compatible and globally efficient local schedules for their pending tasks. Additionally, at the same time agents interact through a complementary process to make solid agreements. These agreements determine fixed programs for earliest scheduled tasks. When these agreements are obtained, project agents will not consider those tasks included as pending.

Case study. The role of resource capabilities

In this section, we show the results of some preliminary simulations, focusing on the role of resource capabilities in multi-project environments. In the final version of the paper, we will extend the results, and we will show a wider set of applications and its implications on managerial decisions.

We consider three different resources (R1, R2 and R3), endowed with the competences C1, C2 and C3 respectively. In table 1, we show a portfolio of six projects, and the tasks needed to complete the project. Each task is defined by the competence needed to be performed and the expected time to be completed. The arrival date is the date when the project is included in the system. Projects can start-up in the starting date; otherwise, they should have been rejected before this date. Due Date 1 (DD1) is the most desirable duration whereas Due Date 2 (DD2) is the maximum allowed. All the projects have a weight of 1.

Table 1: Dynamic portfolio of projects.

Project	Tasks			Arrival date	Starting Date	DD 1	DD 2	Value
	Task 1	Task 2	Task 3					
P1	C1 50	C2 10	C3 20	-30	0	120	180	2500
P2	C3 10	C1 60		-30	0	180	240	3500
P3	C2 15	C1 50		-30	0	120	180	7000
P4	C3 20	C1 45	C2 10	30	120	240	270	4000
P5	C2 15	C3 10	C1 60	30	120	240	270	3000
P6	C3 10	C2 20	C1 50	-30	0	120	180	5000

In figure 1, we show the evolution of the tasks performed by each resource and the prices of the time slots. We also show the evolution of the Duality Gap (upper side of figure 1). The prices of time slots are the solution of the dual problem and the duality gap is a measure of the difference between the primal (expression 1) and dual objective function, so it gives a measure of the quality of the solution [19]. The smaller the duality gap, the more representative the prices, and better solution is obtained.

When the first projects are included in the system, the duality gap is high. Nevertheless, the price formation mechanism makes the prices to stabilise and the gap becomes smaller.

As new projects are required in time 30, the gap increases because the prices of the resources are not stable yet. After some time, the prices change to adapt themselves to the new system conditions and the gap decreases again. Finally, the value of the objective function is 19402.

We have to remark that project P5 has been rejected although it has a high value. The rationale of this fact is based on the project was not available at time 0, when projects P1, P2, P3 and P6 (of lower value) were not compromised.

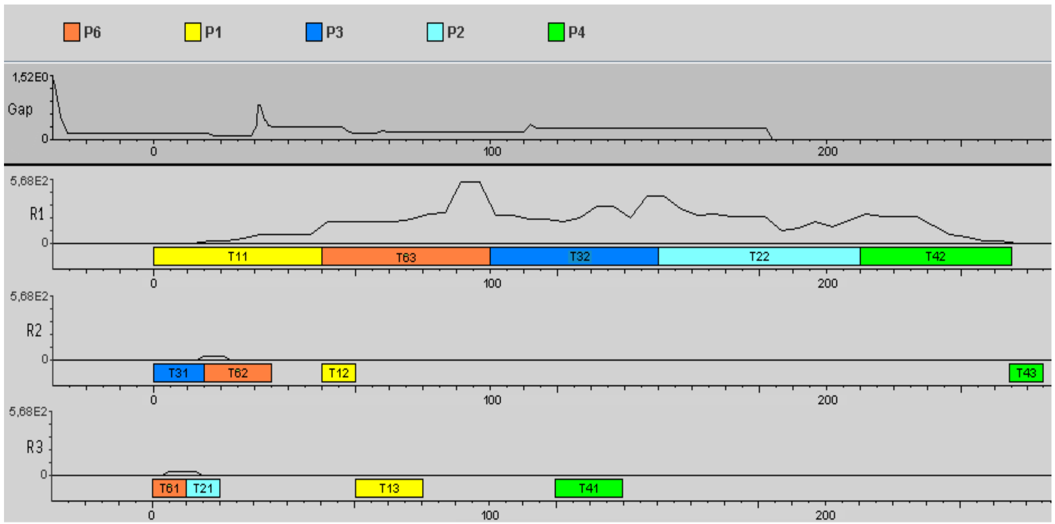

Figure 1. Tasks performed by resources. T_{ij} denotes Task j of project P_i.

The simulation not only gives us the dynamic schedule and the refused projects, but the value of each resource as well. For instance, in figure 1, the prices of the resource R1 are very high during all time slots. This means that the resource competence is very valuable (bottle neck), so if the firm is going to be engaged in similar projects in the nearby future, it would be useful to include more resources with the same competences. On the other hand, prices of resources R2 and R3 are small, although they are working in different tasks during the simulation. So we should inquiry about the possibility of enhancing the range of capabilities of resources R2 and/or R3; for instance, in the case of human resources, this can be done by means of training.

In figure 2, we show the evolution of the system when the resource R2 is also endowed with the competence C1. Compared with the previous case, now the price range is lower for resource R1 and higher for R2. Some tasks exclusively performed by R1 are now made by R2. This shows that the system is capable to use the flexibility of resource R2 to improve the global performance. Now, the project P5 is accepted and executed and the objective function has been increased from 19402 to 24039.

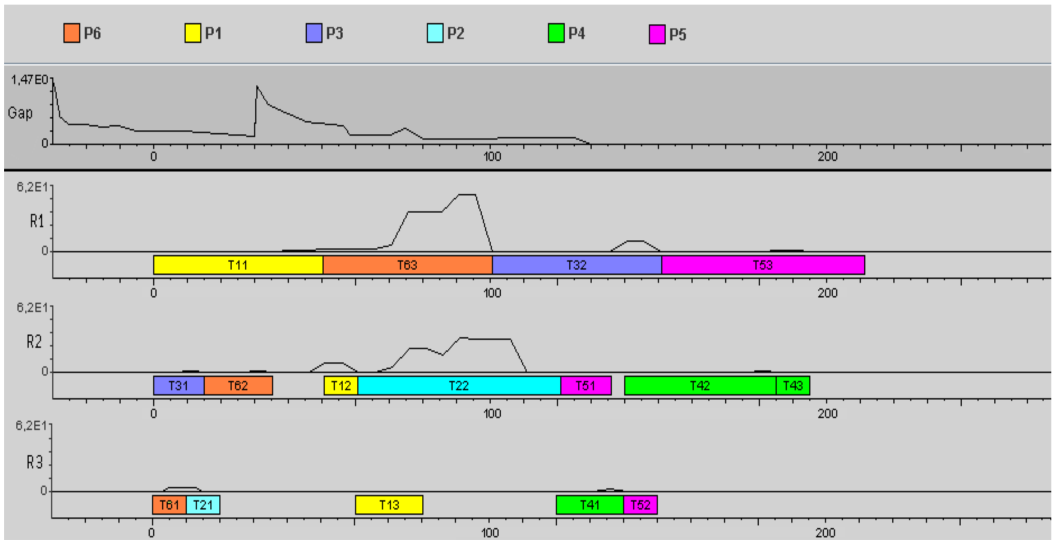

Figure 2. Tasks performed by resources (competences of resource R2 increased).

Conclusions

After studying the literature, we can establish that there are numerous proposals for implementing multi-agent system for multi-project scheduling and control. The most widely accepted idea, which is the one we have followed in our study, consists in building an agent society. In this society the physical agents that represent the real resources interact with other software agents that represent the existing projects. These agents negotiate according to certain schemes, and the outcome of this interaction provides an acceptable performance of the system.

According to these ideas we have developed a small prototype where we intend to verify the adaptation of these techniques for our problem. We proposed a multi-agent system with two types of agents: projects and resources, and we propose an auction mechanism to distribute scheduling and control activities in the set of agents. By means of this auction mechanism, time slots of resources are valued and these prices are used as scheduling and control decision hints.

The proposed case study shows some of the capabilities of our system to deal with some of the decisions manager necessities in the context of multi-project environments. The system allocates dynamically resources to projects, and decides what projects to accept or reject taking into account project values, profitability and (feedback) operational information. We also show how it is possible to discover which resources are the most valuable for a given portfolio and hence they might be strength. Our approach has some advantages over traditional methods: the system makes easier to capture the complexity of multi-project environments, it allows using resource flexibility to improve the system efficiency, and also it gives the option of distributing the calculation over several agents. The evolution of duality gap in the case study shows that the price formation mechanism achieves prices which remain close to equilibrium, and therefore, good solutions are found.

Preliminary results show efficient performance, but there are still many issues to investigate. Future work will be devoted to test the proposed approach in more case studies including more complex structure of the projects, analyzing aspects as convergence and stability depending on the auction mechanism or studying the performance of the system in real cases.

Acknowledgements

The authors have benefited from the financial support of the Spanish Ministry of Education and Science (projects DPI2005-05676 and TIN2008-06464-C03-02) and of the JCyL (project BU034A08).

References

[1] S. Anavi-Isakow, B. Golany. Managing multi-project environments through constant work-in-process. *International Journal of Project Management* 21:9-18, 2003.

[2] L. Cabac. Multi-agent system: A guiding metaphor for the organization of software development projects. In: J. G. Carbonell and J. Siekman (Eds.), *Multiagent System Technologies. Lecture Notes in Artificial Intelligence Vol. 4687*, Springer, Berlin / Heidelberg, pages 1-12, 2007.

[3] S. Clearwater. *Market-Based Control: A Paradigm for Distributed Resource Allocation*. World Scientific, 1996.

[4] I. Cohen, A. Mandelbaum and A. Shtub. Multi-Project Scheduling and Control: A Process-Based Comparative Study of the Critical Chain Methodology and Some Alternatives. *Project Management Journal* 35:39-49, 2004.

[5] G. Confessore, S. Giordani and S. Rismondo. A market-based multi-agent system model for decentralized multi-project scheduling. *Annals of Operations Research* 150:115-135, 2007. DOI: http://dx.doi.org/10.1007/s10479-006-0158-9

[6] J.M. Galán, R. Olmo and A. López-Paredes. Diffusion of Domestic Water Conservation Technologies in an ABM-GIS Integrated Model. In: E. Corchado, A. Abraham, and W. Pedrycz (Eds.), *HAIS '08: Proceedings of the 3rd international workshop on Hybrid Artificial Intelligence Systems. Lecture Notes in Artificial Intelligence 5271*, Springer-Verlag, Berlin, Heidelberg, pages 567-574, 2008.

[7] E.W. Hans, W. Herroelen, R. Leus and G. Wullink. A hierarchical approach to multi-project planning under uncertainty. *Omega* 35:563-577, 2007. DOI: http://dx.doi.org/10.1016/j.omega.2005.10.004

[8] N.R. Jennings, M.J. Wooldridge. Applying agent technology. *Applied Artificial Intelligence* 9:357-369, 1995. DOI: http://dx.doi.org/10.1080/08839519508945480

[9] H.P. Kao, B. Wang, J. Dong and K.C. Ku. An event-driven approach with makespan/cost tradeoff analysis for project portfolio scheduling. *Computers in Industry* 57:379-397, 2006. DOI: http://dx.doi.org/10.1016/j.compind.2005.11.004

[10] K. Kim, J. Paulson. Agent-based compensatory negotiation methodology to facilitate distributed coordination of project schedule changes. *Journal of Computing in Civil Engineering* 17:10-18, 2003. DOI: 10.1061/(ASCE)0887-3801(2003)17:1(10)

[11] K. Kim, J. Paulson. Multi-agent distributed coordination of project schedule changes. *Computer-Aided Civil and Infrastructure Engineering* 18:412-425, 2003.

[12] K. Kim, J. Paulson, R.E. Levitt, M.A. Fischer and J. Petrie. Distributed coordination of project schedule changes using agent-based compensatory negotiation methodology. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM* 17:115-131, 2003.

[13] K. Kim, J. Paulson and J. Petrie. Agent-based project schedule coordination. *Computing in Civil and Building Engineering* 1:732-739, 2000. DOI: http://dx.doi.org/10.1061/40513(279)96.

[14] S.R.T. Kumara, Y.H. Lee and K. Chatterjee. Distributed multiproject resource control: A market-based approach. *CIRP Annals - Manufacturing Technology* 51:367-370, 2002.

[15] Y.H. Lee, S.R.T. Kumara and K. Chatterjee. Multiagent based dynamic resource scheduling for distributed multiple projects using a market mechanism. *Journal of Intelligent Manufacturing* 14:471-484, 2003.

[16] A. López-Paredes. *Ingeniería de Sistemas Sociales. Diseño, Modelado y Simulación de Sociedades Artificiales de Agentes*. Servicio de Publicaciones de la Universidad de Valladolid, Valladolid, 2004.

[17] P.B. Luh, D.J. Hoitomt. Scheduling of manufacturing systems using the Lagrangian relaxation technique. *IEEE Transactions on Automatic Control* 38:1066-1079, 1993. DOI: http://dx.doi.org/10.1109/9.231461

[18] J. Pavón, J. Gómez-Sanz. Agent Oriented Software Engineering with INGENIAS. In: V. Marik, J. Müller, and M. Pechoucek (Eds.), *Multi-Agent Systems and Applications III, 3rd International Central and Eastern European Conference on Multi-Agent Systems, CEEMAS 2003. Lecture Notes in Artificial Intelligence 2691*, Springer-Verlag, Berlin Heidelberg, pages 394-403, 2003.

[19] J. Wang, P.B. Luh, X. Zhao and J. Wang. An optimization-based algorithm for job shop scheduling. *Sadhana - Academy Proceedings in Engineering Sciences* 22:241-256, 1997.

[20] M. Wooldridge. *An Introduction to Multiagent Systems*. John Wiley & Sons, 2002.

[21] S. Wu, D. Kotak. Agent-based collaborative project management system for distributed manufacturing. *IEEE International Conference on Systems, Man and Cybernetics, 2003*, pages 1223-1228, 2003.

[22] Y. Yan, T. Kuphal and J. Bode. Application of multiagent systems in project management. *International Journal of Production Economics* 68:185-197, 2000. DOI: 10.1016/S0925-5273(00)00082-7

[23] F. Ygge, H. Akkermans. Decentralized Markets versus Central Control: A Comparative Study. Journal of Articial Intelligence Research 11:301-333, 1999.

[24] X. Zhao, P.B. Luh and J. Wang. Surrogate Gradient Algorithm for Lagrangian Relaxation. *Journal of Optimization Theory and Applications* 100:699-712, 1999. http://dx.doi.org/10.1023/A:1022646725208