Agent-based modeling and simulation of multi-project scheduling

José Alberto Araúzo, Javier Pajares, Adolfo Lopez-Paredes
Social Systems Engineering Centre (INSISOC)
University of Valladolid
Valladolid (Spain)
{arauzo,pajares,adolfo}insisoc.es

Juan Pavón
Facultad de Informática
Universidad Complutense de Madrid
Madrid (Spain)
jpavon@fdi.ucm.es

***Abstract*—There are no analytical solutions for the problem of dynamic scheduling of resources for multiple projects in real-time. Mathematical approaches, like integer programming or network based techniques, cannot describe complexity of real problems (multi-projects environments have many interrelated elements), and have difficulties to adapt the analysis to dynamics changes. However, this complex problem can be modeled as a multi-agent system, where agents negotiate resources through an auction inspired mechanism. Agents can be used to represent projects and resources. Projects demand resources for fulfilling their scheduled planned work, whereas resources offer their capabilities and workforce. An auction inspired mechanism is used to allocate resources to projects and the price of resources emerges and changes over time depending on supply and demand levels in each time slot. By means of this multi-agent system, it is possible to overcome most of the problems faced in multi-project scheduling such as changes in resources capabilities, allocation flexibility, changes in project strategic importance, etc.**

Keywords—agent-based modelling; agent-based simulation; multi-project environments; auction based resources allocation; project scheduling.

I. Introduction

The problem of allocating resources for multiple concurrent projects appears in large cases of service and manufacturing organizations. A paradigmatic example can be an engineering projects office. This organization makes different kinds of projects that are proposed at any time, which must be handled in a given time frame. Each project consists of a number of activities (calculations, design, checks, budgeting, etc.) that are performed by workers and with some precedence relationships. The workers can perform one or several activities according to their skills. Decision makers have to reject inadvisable projects and decide which resources will be allocated to which projects and when.

Previous decisions have high impact in the office's profit. In order to achieve strategic goals it is important to give priority to projects, and to allocate activities to the most efficient workers at the appropriate time. Because of this, before executing projects it is advisable to make a schedule that optimizes the allocation of resources.

Classical methods are based on mathematical programming and can solve this problem when the complexity is low. And there are some heuristics and meta-heuristics that are able to provide good schedules for more complex problems [9]. The traditional scheduling and control systems propose hierarchical and centralized architectures, where a classical scheduler system that has a global model of the multi-project environment makes schedules according to the current state of the system. Hans et al. [4] 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.

But these techniques are not flexible or robust enough, and have difficulties to consider many real factors. In addition, real environments undergo frequent changes (new resources, new technologies) that force to modify the scheduling system. The traditional scheduling and control systems, which are based on hierarchical and centralized architectures, have not enough flexibility to adapt themselves to the dynamism and complexity of multi-project environments.

These issues have motivated, in last years, successive proposals are appearing to improve the scheduling and control in a multi-project environment. The paradigm of Multi-agent Systems (MAS) can help to find solutions, especially in cases where some social behaviour emerges. This paper shows an agent-based approach for online dynamic scheduling and control in multi-project environments that takes advantage of the ability of agents to negotiate and adapt to changing conditions. The MAS has basically two types of agents: projects managers and resources managers.

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. The design of the auction process uses a technique that has been proposed for distributed scheduling in the literature [8], [14], [11].

This agent-based approach has two distinctive aspects with respect to other works: the integration of strategic decisions (accept or reject new projects) and operative aspects (resource allocation), and the ability to manage resource flexibility. This allows mangers to study the advisability of increasing the flexibility of resources.

The next section introduces the role of agent-based modeling and simulation in project scheduling. Section 3 presents the MAS for the real-time scheduling problem, which has been specified with an agent-oriented modeling language, INGENIAS [10]. This has been the basis for implementing a simulation, which is described in section 4, and whose results are discussed in section 5. Finally, section 6 presents main conclusions of using this agent-based modeling and simulation approach.

II. AGENT ORIENTED MODELING AND SIMULATION FOR REAL-TIME SCHEDULING OF MULTIPLE PROJECTS

Multi-projects environments are complex and dynamic systems. They include many components and dependencies, and many changes may occur in the execution of projects. Moreover, projects are inherently distributed; each task may be completed by different resources or in different geographical locations and each project manager may be in different places.

MAS have been shown to deal with problems of *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) [5] [12].

In the particular case of multi-project systems, the agents can be abstracted as tasks, resources, project managers, etc. This design enables 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 facilitates the design of market mechanisms to solve the scheduling problem by means of distributed approximations [2]. Recently, Lee, Kumara and Chatterjee [7] have proposed an agent-based dynamic resource scheduling for multiple distributed projects using market mechanisms. Following the same research line, Confessore et al. propose in [3] another iterative combinatorial auction mechanism. Other examples of agent-based approaches in project management can be found in the works of Kim and colleagues [6], Wu and Kotak [13], and Cabac [1].

III. A MAS MODEL FOR MULTIPLE PROJECT SCHEDULING

The system can be modeled with two types of agents representing project and resource managers. Agents have the ability to interact with each other. In this case, it is important to define an auction protocol for project agents to compete for the use of resources. Resource Manager Agents interact with project agents to inform on the status, capabilities and cost at each specific time. A third type of agent is included in the system to create new agents and monitoring the global behavior.

A. Project Manager Agents

Each project is associated to a *Project Manager Agent*. The system is considered dynamic: while some projects are being developed other projects can be included or rejected in real-time, which implies the creation and deletion of the corresponding agents.

At any instant t there are I projects in the system, each one denoted by i. Each one 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^*, which 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 whether to accept or reject the project.

Each project i consists of J_i activities, each one denoted by ij, where $i \in \{1, 2,\dots, I\}$ and $j \in \{1, 2,\dots, J_i\}$. Every activity j of a project i is associated with a competence $h(i,j)$. Any activity ij with a given competence $h(i,j)$ can be performed by a resource m just if m is endowed with the competence $h(i,j)$. The duration of the activity ij depends on the resource assigned to perform it. The duration of activity ij in resource m is denoted as d_{ijm}. It is calculated according to $d_{ijm}=d_{ij}/e_{m,h(ij)}$, where d_{ij} is the standard duration of activity j of project i and $e_{m,h(ij)}$ is the efficiency of resource m to perform the competence $h(i,j)$.

This first simplified model assumes 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. There is also the assumption 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.

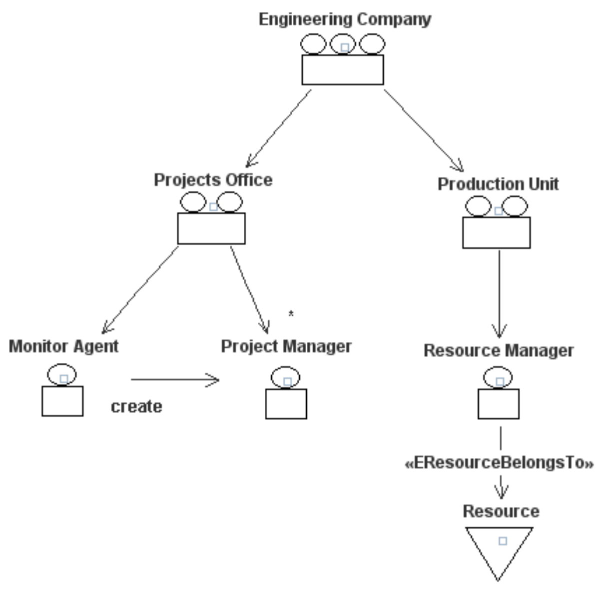

Figure 1. MAS organization model (with INGENIAS notation [10]). The diagram shows an organization (Engineering Company), which has several departments (Projects Office and Production Unit). The Projects Office has one Monitor Agent and several Project Manager Agents. The Production Unit has Resource Managers that take care of the use of Resources.

B. Resource Manager Agents

A resource is modelled as a *Resource Manager Agent*. There are M resources, which can be assigned simultaneously to one activity. Each resource is endowed with a given cost rate per unit of time, c_m ($m \in \{1, 2, 3\ldots M\}$), and a subset H_m of competences that can be performed ($H=\{h_1, h_2, \ldots h_K\}$ is the set of competences that are necessary to complete the projects).

Each resource has a certain grade or ability to perform a competence. Therefore, the work capacity of resources can be symbolized by means of a vector of abilities per resource $e_m=(e_{m1}, e_{m2},\ldots,e_{mk})$, where $e_{mf} \geq 0$ shows the ability degree of resource m to perform the competence h_f. If $e_{mf} = 0$ then the resource m has not the competence h_f, if $0 < e_{mf} < 1$ the resource is able to perform inefficiently the competence h_f, if $e_{mf} = 1$ it has standard efficiency to perform the competence, and if $e_{mf} > 1$ it will do it efficiently.

C. Monitoring Agent

A Monitoring Agent has the responsibility to visualize the current state of the system to the user. Moreover, this agent allows the user to create new *Project Manager Agentsm*, as shown in Figure 1.

IV. Agent workflows and Interactions

The agent workflows and interactions must be designed in order to maximize the global efficiency of the system, which will be evaluated by the average benefit obtained in a certain time interval *T* according to:

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

for all projects *i* that are finished in *T*, *Cost(i)* is the cost to complete the project *i*. This cost has two components, the direct resource cost and the delay cost:

$$Cost(i) = \sum_j \mathrm{C}_{\bar{m}(j)} \cdot \frac{d_{ij}}{\mathrm{e}_{ij\bar{m}}} + w_i \cdot (D_i - F_i)^2 \qquad (2)$$

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 with 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.
- The 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.

A. Auction Interactions

At any time, the system has as many Project Manager Agents as projects are ordered. Each one represents a particular 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. In order to achieve their goal, Project Manager Agents make plans that take into account only their own activities (local schedule).

The decision-making process is decentralized as it emerges from interactions among the agents in an auction process. Each project manager creates its own schedule (local schedule) by taking into account its own project goals and its own knowledge. This procedure can bring incompatible local schedules (several projects try to use the same resource at the same moment). Moreover, the local schedules can be globally inefficient (profitable projects are rejected; most important projects have delays; etc). These difficulties that arise from the autonomy of each agent are solved with a market mechanism that ensures that local schedules are nearly compatible and globally efficient according to the expression (1). This auction based multi-project scheduling approach is founded on Lagrangian Relaxation [8][11][14], a decomposition technique for mathematical programming problems.

In order to apply the market metaphor, the periods when resources are available are subdivided in a set of small time intervals or time slots. Each time slot on each resource is modelled as a *good* that can be sold in an auction, where each resource acts as a seller. Thus, a local schedule will be a bundle of time slots that has been allocated to a project.

The number of sellers is equal to the number of resources in the system. 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 plays the role of a *bidder* that participates in auctions by asking the Resource Manager Agents for the set time slots that it requires to execute its pending tasks at the current time. It will try to find a set of time slots (Z_i) through the resource pool while incurring the minimum possible local cost (LC_i). This cost has two components, the sum of the price of the selected time slots and the delay cost (expression 3):

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

where p_{mt} is the price of the time slot (t) of the resource (m).

To select the set of time slots (Z_i) that minimizes their local cost, Project Manager Agents use a dynamical programming algorithm where all possible combinations of time slots and resources are considered [13]. In their decision, they take into account that only those resources endowed with the necessary competences can carry out a certain activity. Moreover, the number of time slots necessary to complete a task (duration) are determined according to the ability degree of the resource in the competence. Each project agent will regard as scheduling horizon the time slot that 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 allows to schedule tasks before D_i^*, 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 must be rejected.

Each Resource Manager Agent determines the price charged for the time slots with the purpose of reducing resource conflicts and maximizing their revenue. In order to get this goal a subgradient optimization algorithm is used to adjust prices at each round of bidding. By means of this algorithm the Resource Manager Agents increase the price of the time slots where there is conflict (more than one project manager has 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 reduced.

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 by means of the expression 4. α^n is calculated according to [8].

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

Where:

- $p_{mt}^{\ n+1}$ is the price of the time slot (t) of resource (m) at the round (n+1)
- $p_{mt}^{\ n}$ is the price of the time slot (t) of resource (m) at the round (n)
- α^n is the step at the round (n). It decreases when (n) increases.
- And ($g_{mt}^n = a_{mt}^n - 1$) is the subgradient, where a_{mt}^n is the demand of slot (t) of resource (m)

B. *Contract Interactions*

By means of the auction mechanism described above, project agents build compatible and globally efficient local schedules for their pending activities. Moreover, at the same time, agents interact through a complementary process to make firm agreements based on the local schedules that have been created by means of the auction process. These agreements determine fixed programs for earliest scheduled tasks. When these agreements are obtained, project agents will never consider the tasks included as firm contracts as pending.

The global efficiency and the compatibility of local schedules depend on the degree of convergence of market prices to the equilibrium prices. If the prices get closer to the equilibrium price, they will be representative of the system state; they will have information about any system feature and local schedules will be compatible and globally efficient. If agents are making firm contracts when prices are not representative of the system state, then incompatibilities could take part. In these cases, the agents resolve incompatibilities by means of local schedule based heuristics rules. More exactly, when several activities use the same resource at the same moment, the activity that has been earliest programmed in local schedule will have priority to be contracted in firm agreements. Although this heuristic does not ensure global efficiency, it will achieve perfect compatibility in final decisions.

V. SIMULATION AND RESULTS

The system has been implemented and simulated with different scenarios. Here the analysis focuses on the role of resource capabilities and the option of project rejection. The first scenario shows a simple case to illustrate the main features of the system, in the next subsection. This is followed by a dynamic scenario in order to evaluate the system performance in evolving complex environments.

A. *Simple Case Study*

Consider three different resources (R1, R2 and R3), endowed with the competences C1, C2 and C3 respectively. TABLE I. shows a portfolio of five projects, and the tasks needed to complete each project. Each task is defined by means of the pertaining competence and expected standard time to be completed.

TABLE I. SIMPLE CASE STUDY

Proj.	**Tasks**			**Arrival date**	**Starting Date**	**DD1**	**DD2**	**Value**
	Task 1	***Task 2***	***Task 3***					
P1	C1 50	C2 25	C3 30	0	0	120	180	10000
P2	C3 40	C1 45	C3 10	0	0	180	240	12000
P3	C2 35	C1 40	C2 25	0	0	120	180	30000
P4	C3 30	C1 50	C2 10	50	90	150	270	15000
P5	C1 45	C3 20	C1 50	50	90	150	270	30000

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.

Figures 2 and 3 show the system state at a given time (current time). In the upper area of the figures the relative duality gap evolution is presented. 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 and dual objective function, so it quantifies the quality of the solution [8]. The relative duality gap is calculated as the duality gap divided by the dual solution. A small relative duality gap means that the prices are representative of the system state, thus, a good solution is achieved. The lower part of the figures present charts of resources. These charts show the tasks that each resource has performed until the current time (lower area of the resource charts) and the time slot prices (upper area of the resource chart). The time slots prices previous to current time are the prices when agents were doing firm agreements for those time slots. The prices later than current time are the estimated prices in the current round of the auction.

Figure 2. shows the system state at the moment 45, just before projects P4 and P5 arrive at the system. When the first projects (P1, P2 and P3) are included in the system, the duality gap is high, indication of a bad solution. But then, the price formation mechanism makes the prices to stabilise, and the gap becomes smaller. This means that prices are close to equilibrium.

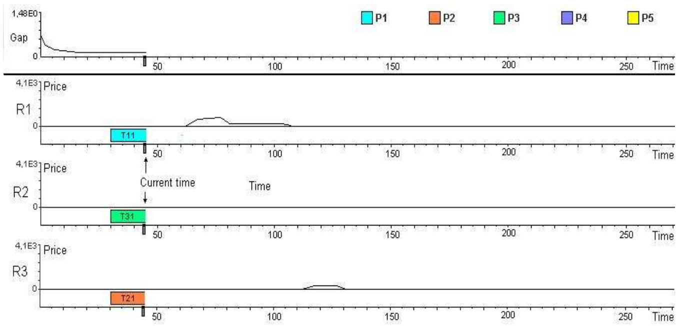

Figure 2. System state at the moment 45

Figure 3. shows the evolution of the tasks performed by each resource and the prices of the time slots after finishing the simulation. This figure shows how the duality gap increases when the projects P4 and P5 arrive to the system. At this moment previous prices did not reflect the new system state (new projects are in the system). After some time, the prices change to adapt themselves to the new system conditions, and the gap decreases again. It can be observed that the new prices are very different from previous prices. This happens especially in resource R1 where prices are very high.

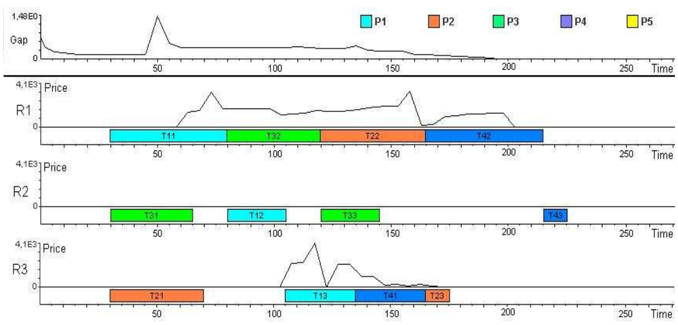

Figure 3. Tasks performed by resources. Tij denotes Task j of project Pi.

Note that project P5 has been rejected although it has a high value, because its value was not available at time 0, when projects P1, P2, P3 were waiting to start-up. The calculus of the payment that projects have done for time slots (TABLE II.) shows that the same projects do a payment higher than their values. When projects P4 and P5 arrive at the system the prices of time slots of the resource R1 grow because P5 is able to pay higher prices to be performed. Although P5 accepts higher prices than other projects, P1, P2 and P3 cannot be rejected and finally they must pay the market prices. The final total value (B_T=total values of performed projects minus total delay cost) is 55700.

TABLE II. PAYMENT FOR TIME SLOTS PER PROJECT

Project	Value	Total payment
P1	10000	12719
P2	12000	11414
P3	30000	8298
P4	15000	6887
P5	0	0

The simulation not only gives the dynamic schedule and the refused projects, but the value of each resource as well. For instance, in Figure 3. the prices of resource R1 are very high during all time slots. This means that the resource competence is very valuable (bottleneck), 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 on different tasks during the simulation.

So, the possibility of enhancing the range of capabilities of resources R2 and R3 should be considered; for instance, in the case of human resources, this can be done by means of training.

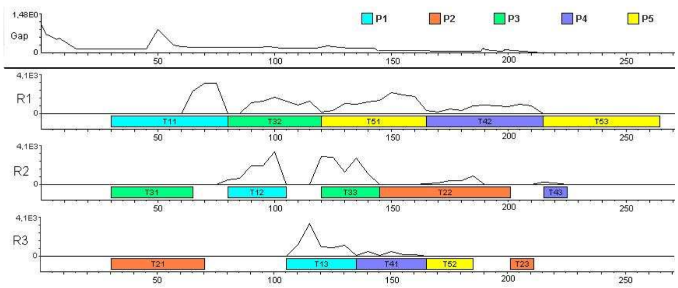

Figure 4. Tasks performed by resources (competences of resource R2 increased).

Figure 4. shows the evolution of the system when the resource R2 is also endowed with the competence C1 with efficiency 0.8. Compared with the previous case, now the price range is lower for resource R1 and higher for R2. Although the duration of task T22 of project P2 is smaller in resource R1 than in R2, now the system have to reallocate in real-time this activity to R2. So, R1 can perform in time the task T51 of the project P5. In this experiment, the project P5 is accepted and executed, and the total value has been increased from 55700 to

69369. This shows that the system is capable to use the flexibility of resource R2 to improve in real-time the global performance.

B. Complex Dynamic Scenario

In order to check the system performance in very dynamic environments, consider 12 projects (table 3) that arrive at the system every 20 units of time (first P1, second P2, …, and finally P12). In TABLE III. DD1 and DD2 are relating to starting date. Resources and competences are similar to the previous case study.

TABLE III. DYNAMIC PORTFOLIO OF PROJECTS (COMPLEX SCENARIO)

Proj.	Tasks			DD1	DD2	Value
	Task 1	*Task 2*	*Task 3*			
P1	C1 50	C2 25	C3 30	60	150	10000
P2	C3 40	C1 45	C3 10	60	120	15000
P3	C2 35	C1 40	C2 25	60	120	6000
P4	C3 30	C1 50	C2 10	90	120	7000
P5	C1 45	C3 20	C1 50	60	120	8000
P6	C3 10	C2 45	C1 20	120	150	7000
P7	C1 20	C2 10	C3 30	60	150	15000
P8	C3 40	C1 45	C3 50	90	120	10000
P9	C2 35	C1 10	C2 45	60	120	15000
P10	C3 30	C1 15	C2 10	60	120	10000
P11	C1 15	C3 50	C1 10	90	150	7000
P12	C3 35	C2 50	C1 20	60	120	12000

We have done several simulations by changing two types of parameters: the response period and the set of competences of resources. The response period is the time interval between the arrival date and the starting date. During this period, projects wait in the system for rejection or acceptance decision. If this period is long, more projects are waiting for decision simultaneously, so decisions will be more efficient.

We have simulated three competence distribution cases: case A (R1 has the competence C1, R2 the C2, R3 the C3), case B (R1 C1, R2 C1 and C2, R3 C3), and case C (R1 C1, R2 C1 and C2, R3 C2 and C3).

Figure 5. shows the total values obtained in different experiments. Each curve represents the value variation for cases (A, B and C) when the response period increases. Note that the system efficiency is higher when the response period increases and when the resources are more flexible (they have more competences). This shows that in this scenario the system performance is suitable; the software is able to manage complexity to improve the global efficiency.

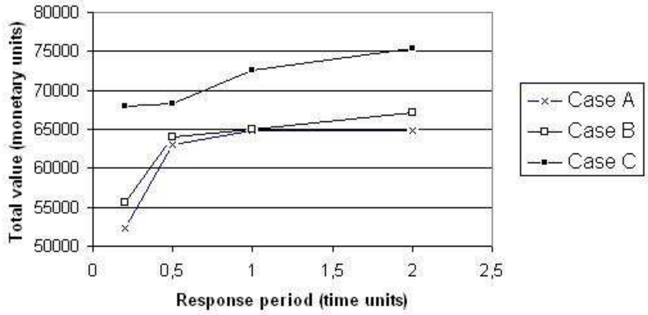

Figure 5. Total value in complex experiments

VI. CONCLUSIONS

Although project management literature has been mainly concerned with managing individual projects, in practice firms usually work in dynamic and complex multi-project environments.

We propose a multi-agent system and an auction mechanism for online dynamic scheduling in multi-project environments. Projects have tasks to be completed, so they compete for the resources endowed with the capabilities required to do some pieces of work. The prices of resources emerge endogenously by means of an auction process.

We show some of the possibilities of this multi-agent approach to deal with some of the decisions that managers need to take within multi-project environments. The system allocates dynamically resources to projects, and decides what projects to accept or reject taking into account project value, profitability and (feedback) operational information. We also show how it is possible to discover which resources are the most valuable, so they should be added to the firm.

This approach contributes to fill the gap between the literature in portfolio project management (usually focused on corporate strategy and finance) with the work in multi-project management (mainly concerned with operational issues, scheduling and resource allocation).

ACKNOWLEDGMENTS

This work has been done in the context of the following projects: (1) “Agent-based Modelling and Simulation of Complex Social Systems (SiCoSSys)”, supported by Spanish Council for Science and Innovation, with grants TIN2008-06464-C03-01 and TIN2008-06464-C03-02; (2) “ABACO VA006A09”, (3) the Programa de Creación y Consolidación de Grupos de Investigación UCM-BSCH GR58-08, and (4) GR251/09 supported by the “Junta de Castilla y Leon”.

REFERENCES

[1] Cabac, L. “Multi-agent system: A guiding metaphor for the organization of software development projects”. In: Carbonell, J.G. & Siekman, J., Multiagent System Technologies. Lecture Notes in Artificial Intelligence 4687. Springer: Berlin / Heidelberg, pp 1-12, 2007

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

[3] Confessore, G., Giordani, S. & Rismondo, S. “A market-based multi-agent system model for decentralized multi-project scheduling”. Annals of Operations Research, 150, pp 115-135, 2007

[4] Hans EW, Herroelen W, Leus R and Wullink G. “A hierarchical approach to multi-project planning under uncertainty”. Omega 35, pp :563-577, 2007.

[5] Jennings, N.R. & Wooldridge, M.J. Applying agent technology. Applied Artificial Intelligence, 9, pp 357-369, 1995.

[6] Kim, K. & Paulson, J. “Multi-agent distributed coordination of project schedule changes”. Computer-Aided Civil and Infrastructure Engineering, 18, pp 412-425, 2003.

[7] Lee, Y.H., Kumara, S.R.T. & Chatterjee, K. “Multiagent based dynamic resource scheduling for distributed multiple projects using a market mechanism”. Journal of Intelligent Manufacturing, 14, pp 471-484, 2003

[8] Luh, P.B. & D. J. Hoitomt. “Scheduling of Manufacturing Systems Using the Lagrangian Relaxation Technique”. IFAC Work Shop on Discrete Event System Theory and Applications in Manufacturing and Social Phenomena, Shenyang, China, 1991.

[9] Newman K, Schwindt C., Zemmermann J. Book review of Project Scheduling with Time Windows and Scarce Resources: Temporal and Resource-Constrained Project Scheduling with Regular and Non-regular Objective Functions. 2nd Edition, Springer-Verlag, 2003

[10] Pavón, J., Gómez-Sanz, J.:“Agent Oriented Software Engineering with INGENIAS”. In: Marik, V., Müller, J., Pechoucek, M. (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, pp 394-403, 2003.

[11] Wang, J., Luh, P.B., Zhao, X. and Jinlin Wang. “An Optimization-Based Algorithm for Job Shop Scheduling”. Sadhana, a Journal of Indian Academy of Sciences, Vol. 22, Part 2, pp. 241-256, 1997

[12] Wooldridge, M.J. An Introduction to Multiagent Systems. John Wiley & Sons Ltd: New York, 2002.

[13] Wu, S. & Kotak, D. “Agent-based collaborative project management system for distributed manufacturing”. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, pp 1223-1228, 2003

[14] Zhao, P. B. Luh, J Wang. “Surrogate Gradient Algorithm for Lagrangian Relaxation”. Journal of Optimization Theory and Applications, Vol. 100, Nº 3, pp 699-712, 1999