- ORIGINAL ARTICLE -

Multi-agent Learning by Trial and Error for Resource Leveling during Multi-Project (Re)scheduling

Aprendizaje Multi-agente utilizando Trial and Error para la Nivelación de Recursos durante el (Re)scheduling de Múltiples Proyectos

Laura Tosselli[1], Verónica Bogado[1,2] and ErnestoMartínez[3]

[1]*Dpto. Ingeniería en Sistemas de Información – UTN FRVM, Villa María, X5900 HLR, Córdoba, Argentina*
{ltosselli, vbogado}@frvm.utn.edu.ar
[2] *CIT Villa María (CONICET-UNVM),Carlos Pellegrini 211, Villa María, Córdoba, Argentina*
[3]*INGAR (CONICET – UTN) – UTN FRSF, Santa Fe, S3002GJC, Santa Fe, Argentina*
ecmarti@santafe-conicet.gob.ar

Abstract

In a multi-project context within enterprise networks, reaching feasible solutions to the (re)scheduling problem represents a major challenge, mainly when scarce resources are shared among projects. Thus, the multi-project (re)scheduling must achieve the most efficient possible resource usage without increasing the prescribed project constraints, considering the Resource Leveling Problem (RLP), whose objective is to level the consumption of resources shared in order to minimize their idle times and to avoid over-allocation conflicts.

In this work, a multi-agent solution that allows solving the Resource Constrained Multi-project Scheduling Problem (RCMPSP) and the Resource Investment Problem (RIP) is extended to incorporate indicators on agents' payoff functions to address the Resource Leveling Problem in a decentralized and autonomous way, through decoupled rules based on Trial-and-Error approach. The proposed agent-based simulation model is tested through a set of project instances that vary in their structure, parameters, number of resources shared, etc. Results obtained are assessed through different scheduling goals, such as project total duration, project total cost and leveling resource usage. Our results are far better compared to the ones obtained with alternative approaches. This proposal shows that the interacting agents that implement decoupled learning rules find a solution which can be understood as a Nash equilibrium.

Citation: L. Tosselli, V. Bogado and E. Martínez. "*Multi-agent Learning by Trial and Error for Resource Leveling during Multi-Project (Re)scheduling",* Journal of Computer Science & Technology, vol. 18, no. 2, pp. 125-135, 2018.

DOI: 10.24215/16666038.18.e14

Received: February 8, 2018 **Revised:** April 28, 2018 **Accepted:** May 14, 2018

Copyright: This article is distributed under the terms of the Creative Commons License CC-BY-NC.

Keywords: Agent-based simulation, Multi-agent System, Multi-project (Re)scheduling, Project-oriented Fractal Organization, Resource Leveling.

Resumen

En un contexto de múltiples proyectos dentro de redes empresariales, alcanzar soluciones factibles al problema de (re)scheduling representa un gran desafío, principalmente al compartir recursos escasos entre proyectos. Así, el (re)scheduling de múltiples proyectos debe lograr el uso de recursos más eficiente posible sin incrementar las restricciones de proyecto planteadas, considerando el *Problema de Nivelación de Recursos*, cuyo objetivo es nivelar el consumo de recursos compartidos para minimizar tiempos ociosos y evitar conflictos de sobre-asignaciones.

En este trabajo, una solución multi-agente para resolver el Problema de Scheduling de Múltiples Proyectos con Restricción de Recursos y el Problema de Inversión de Recursos es extendida para incorporar indicadores en las funciones de recompensa de los agentes para abordar el Problema de Nivelación de Recursos de manera autónoma y descentralizada a través de reglas desacopladas basadas en el enfoque de Aprendizaje por prueba y error. El Modelo de Simulación basado en agentes propuesto es verificado mediante un conjunto de instancias de proyecto que varían en estructura, parámetros, número de recursos compartidos, etc. Los resultados obtenidos se evalúan mediante diferentes objetivos de scheduling, como duración total del proyecto, costo total del proyecto y nivelación en el uso de recursos. Nuestros resultados presentan mejoras en comparación a los obtenidos en enfoques alternativos. Esta propuesta muestra que los agentes interactuantes que implementan reglas de

aprendizaje desacopladas encuentran una solución que puede entenderse como un equilibrio de Nash.

Palabras claves: Nivelación de Recursos, Organización Fractal Orientada a Proyectos, (Re)scheduling de múltiples proyectos, Simulación basada en agentes, Sistema Multi-agente.

1. Introduction

Multi-Project (re)scheduling is considered as an NP-hard problem, thus becoming a difficult task for project leaders when many tasks and resources are involved which prevents the application of optimization-based methods to find a repaired schedule [1]. Thus, in a multi-project context within enterprise networks, there are conflicting constraints and interrelationships among projects that cause an increase of the complexity, making project (re)scheduling a difficult problem to be addressed under real-time pressure and selfish behavior of concerned agents [2]. Particularly, the unplanned events impact on due dates and achievement of milestones, premature budget consumption, and resources usage, which in turn affect timing and quality of delivered goals of different projects because of inefficient responses to such events.

In particular, the consumption and allocation of limited and shared resources in a multi-project context leads to conflicts in the joint schedule, affecting the related project constraints such as time, cost, quality, etc., and causing the need to project (re)schedule. The conflicts associated with consumption and resource allocations are related to different project scheduling problems such as the *Resource Constrained Project Scheduling Problem* (RCPSP), the *Resource Investment Problem* (RIP) and the *Resource Leveling Problem* (RLP), among others. The RCPSP aims to minimize the total project makespan; the RIP has as objective the minimization of the total project cost; and the RLP aims to achieve the most efficient resource consumption without increasing the prescribed makespan while avoiding over-allocation of resources [3, 4].

Thus, to achieve the multi-project (re)scheduling problem, considering the RCPSP, RIP and RLP problems simultaneously, requires techniques and tools that allow decision-making in an autonomous, efficient and decentralized way, providing metrics and indicators that support the decision-making process when unplanned events affect the initial schedule for multiple projects. Therefore, the Project-oriented Fractal Company Model developed in [5] was considered as a basis to implement a Multi-agent Simulation Model to aid solving the multi-project (re)scheduling problem. This Agent-based Model is composed of autonomous and selfish agents with different roles that incorporate in their decision-making behavior a decoupled learning rule that makes room for learning by trial-and-Error [6].

The objective of this work is to integrate new indicators to be considered by the interacting agents that composed the Agent-based Simulation Model published in [6]. As a result, a Multi-agent solution to that aims to level the use of resources by multiple projects, avoiding over-allocation problems is found. A standard and representative set of multi-project problem instances [7] that vary in their structure, parameters and number of shared resources is used to test the presented Multi-agent Model solution against other solutions to the multi-project (re)scheduling problem through performance measures that allow evaluating the emergent solution, including the resource leveling. Results obtained are encouraging and demonstrate the applicability of agent-based simulation to achieve a trade-off among the presented project scheduling problems (RCPSP, RIP and RLP) resulting from restrictions due to resource sharing by several projects, and strategic interactions among the interacting agents. Particularly, this work focalizes on the impact of considering resource leveling indicators in the solution found, i.e., in the process of (re)scheduling to obtain feasible schedules when the interacting agents lead to a Nash equilibrium.

2. Related Work

In recent years, the use of tools based on artificial intelligence techniques to deal with multi-project (re)scheduling problems considering multiple objectives simultaneously has increased. Mainly, these tools are needed to solve scheduling problems in contexts with the unrestricted amount of projects, tasks, and resources involved, and in which it is required to achieve a trade-off among conflicting objectives, such as cost, time, efficient use of resources, etc. The most used artificial intelligence techniques for this purpose are genetic algorithms and agent-based systems.

The proposals presented in [8, 9] use genetic algorithms to solve the multi-project (re)scheduling problem considering multiple objectives. Shahsavar et al. [8] consider three scheduling objectives, which are: (1) minimization of project duration; (2) minimization of project total cost; and (3) minimization of the variability of resource use in the different periods of project execution. In [9], distinct unplanned events, including the need to perform new tasks, the incorporation of new employees or the unavailability of resources, are considered. These events can affect the multiple objectives raised during the (re) scheduling process, defined as cost, duration, robustness, and stability of the

solution found. Both papers evaluate the performance of the genetic algorithms proposed using metrics based on project constraints such as cost, duration, and use of resources, among others. However, these approaches may have low efficiency rates because they seek a solution in a large combinatorial space of constraints, where the solution obtained may be a local optimum. Additionally, the difficulty of such approaches is found in the selection of parameters and evaluation functions for a particular application of the proposed approach.

Several works related to the multi-project scheduling problem have been presented using the agent-based approach [10, 11, 12, 13] where an initial schedule is obtained from the interaction among relevant entities in the problem domain. Adhau et. al. [10, 11] proposed a multi-agent system for the multi-project schedule, establishing the allocation of projects to resources through auction mechanisms. The authors consider a central and coordinator agent for the resolution of conflicts and communication among agents, which can cause a performance decrease in the algorithm execution when the number of resources that negotiate increases. Similarly, [12] presents an agent-based approach for project scheduling that solves the Resource-Constrained Project Scheduling Problem using a learning algorithm for modeling the strategy used to find a solution. In [13], the project planning process that carries out an organization at the strategic level is modeled using a *Multi-agent System* (MAS), where the interacting agents decide which projects to plan and which to reject, using metrics based on costs of such projects. These proposals do not present results that address explicitly the Resource Leveling Problem. Furthermore, many of existing proposals are based on the assumption that the project schedule will be implemented as initially defined. This situation is not representative of the project management environment where typically 80% of initial project schedules are affected by unplanned events during their execution, thus compromising deadlines, costs and estimated resource usage [14].

3. Problem definition

In this section, the multi-project (re)scheduling problem based on a project-oriented fractal model for enterprise networking is defined as follows:

- A set of *I* projects to be (re)scheduled within a project-oriented fractal organization, whose managers (agents) must respond to unplanned events and disturbances. These projects are interdependent and run to a certain extent in parallel. Additionally, each project has properties such as deadline, budget, estimated start and finish time, resource requirements, dependence and precedence relationships, among others. Some of these properties are considered as domain constraints. Each project ***i*** is considered as a *fractal unit* and it consists of a *recursive* structure, where a project can be composed of sub-projects, and these sub-projects can be composed of other sub-projects, and so on and so forth. For this reason, the recursive decomposition of a project continues until its minimum expression, i.e., a task that is also considered as a project. Each *fractal unit* or *project* is composed of a project manager and a managed object, where each managed object can be an *end* (e.g., a project, a sub-project, a phase) or a *mean* (a resource). Each project manager carries out functions such as: negotiation, (re)scheduling and has learning capabilities.
- A set *R* of links of *delegation* and *client-server* type. The delegation relationships are the result of recursive structure and client-server relationships are the result of negotiations, which are auction-based interactions among fractal units that manage projects or resources, where each manager of a project demands resources for fulfilling its scheduled work, whereas the managers of resources sell their specific capabilities and skilled workforce to different projects. The establishment of client-server and delegation relationships among *end*-managers (projects) and *mean*-managers (resources) provides the flexibility for rescheduling tasks at different abstraction levels, following the Project-oriented Fractal approach [5].
- A set of *K* renewable and available resources to processing the projects. Each resource has a list of allowed projects that can process.
- A set of *E* unplanned events that may affect the execution of the initial schedule. In this work, the types of events are related to variations in the availability of resources.

The scheduling of different goals considered in this work are related to three project scheduling problems that should be traded off in a multi-project context: *Resource Constrained Project Scheduling Problem, Resource Investment Problem* and *Resource leveling problem*, which are related to the project time, the project total cost and the leveling of project resources, respectively. These goals are:

(1) Minimize the total duration of the project set *I*.
(2) Minimize the total cost of the project set *I*.
(3) Balance the resource utilization in each period (solution stability) of project set *I*.

The relationship that exists between each of the stated objectives and scheduling problems mentioned is defined below.

Resource Constrained Project Scheduling Problem. The first goal in addressing the rescheduling of multiple projects is related to the *Resource Constrained Project Scheduling Problem* (RCPSP), which seeks to obtain, for each project, a schedule that have the shortest possible total project duration (TPD), subject to limited amounts of the resources available, precedence/synchronization relationships among projects/tasks, and project due date constraints [11, 12]. This objective is defined as follows for any project ***i***:

$$TPD = min\{max\{ft_i\}\} \quad \text{(Eq. 1)}$$

where, the term ft_i is defined as the finalization time of latest task in the project ***i*** schedule.

Resource Investment Problem. The second scheduling goal considered in this work aims to obtain solutions that present the lowest possible total cost (TC), giving rise to solutions to rescheduling of multiple projects corresponding to the *Resource Investment problem* (RIP) [8]. In this context, the objective is defined as follows:

$$TC = min \sum_{i=1,k=1}^{n,q} ral_{ik} \quad \text{(Eq. 2)}$$

where ral_{ik} is the cost of each resource allocation that forms part of the schedule for project ***i***.

Resource Leveling Problem. The third scheduling goal is related to the problem named *Resource leveling problem* (RLP) [8], and it aims to find solutions (schedules) that balance the use of each resource and reduces its idle time during all project execution (stability), since an inadequate leveling of resources usually leads to increasing execution time and cost to complete a given project. When the use of resources is leveled, the final schedule must have a resource profile where variations between periods are minimized and the use of resources is as balanced as possible [4]. The definition of RLP implies the consideration of a set of properties related to objects managed by concerned agents:

- Each *end*-managed object (project ***i***) requires req_{ik} units of *mean*-managed object (resource ***k***) per period.
- Each *mean*-managed object (resource ***k***) has a limited maximum availability, a processing capacity and a resource processing cost for each period.
- Each *mean*-managed object (resource ***k***) is defined as *local* or *global* resource.

Thus, for a given resource, the goal related to RLP, named as *Use Resource Variability* (URV) is framed as an optimization problem as follows:

$$URV = min \sum_{k=1}^{q} \sum_{t=1}^{T} \left(\frac{ur_{k,t} - ur_{k,t-1}}{ma_k} \right) \quad \text{(Eq. 3)}$$

Subject to.

$$ur_{k,t} \leq ma_{k,t} \quad \text{(Eq. 4)}$$

where $(ur_{k,t})$ and $(ma_{k,t})$ represent, respectively, the amounts of resource usage and maximum availability of resource for each period *t*. This goal allows measuring the adaptability of the solution found to the changes in the project environment and permits resource leveling while complying with the availability constraints to avoid over-allocations (according Eq. (4)), minimizing variations in resource usage between time periods [4, 8]. In this work, the goal related to RLP specified in [6] is redefined to consider the maximum availability of resources in each period of time *t*. Therefore, a goal that allows measuring the variation in the use of resources between periods is obtained, leading to a schedule without over-allocations.

4. Multi-agent based Methodologies

4.1. Agent-based Modeling and Simulation

In this proposal, the developed simulation model seeks to imitate the sequence of strategic interactions in the Project-oriented Fractal Organization when responding to events while addressing the multi-project (re)scheduling problem in a decentralized and distributed way. Thus, the multi-project (re)scheduling problem is modeled as a non-cooperative, repetitive and interdependent game [15], involving *N* players (agents) and *M* stages (repeated iterations), where each of the agents pursues its specific goals in a selfish manner. Furthermore, the simulation model is a key tool to evaluate emergent scheduling behaviors from established client-server and delegation relationships among strategically interacting agents.

The proposed agent-based model is composed of two kinds of agents, *Project* and *Resource* agents. Each agent class has different goals. A *Project* agent (PA) manages a project, a sub-project or a task, aims at minimizing its duration, total cost, and variations in the use of resources between periods. On the other hand, a *Resource* agent (RA) manages a resource aiming to maximize its profits while avoiding over-allocation conflicts. These individual goals (micro-level) lead to global and emergent behaviors (macro-level) which are aligned with the scheduling objectives defined in *Section 2*. These entities do not have complete knowledge about the other agents' strategies and payoffs, and there are also constraints related to the exchange of information of strategic

value. The defining properties for project and resource agents are described in Table 1 and Table 2 respectively.

The above defined agent types for interacting managers used in both defining an initial scheduling and during (re)scheduling, where the latter is called for when an unplanned event occurs. The interactions of these agents in the proposed game, through decentralized mechanisms, provide an easier way to obtain a schedule flexibly adapted to the unplanned events mentioned above, where their decisions depend only on the negotiations that they carry out. To respond to abnormal disturbances, the agents must incorporate learning capabilities for (re)scheduling when choosing alternative courses of action and gain experience from the situations that may arise. In the next section, *learning by trial-and-error* [16] using decoupled rules are incorporated to solve the multi-project (re)scheduling problem, selecting at each strategic interaction among agents those actions that present a greater benefit to achieve agent's goals, according to their payoff functions. More importantly, the resulting repaired schedule corresponds to near Nash equilibrium for all concerned agents.

Table 1. *Project* agent properties

Property	Description
pId	Project ID
pLevel	Project level
pGoal	Project goal
estST	Estimated project start time
estFT	Estimated project finish time
deadline	Project latest finish time
PR_i	Set of precedence relationships between other projects and project ***i***
DR_i	Set of dependence relationships between other projects and project ***i***
req_{ik}	Requirements of resource ***k*** for project ***i***
P_u_i	Payoff function for project agent ***i*** defined as: $minTPD_i + TC_i + URV_i$

Table 2. *Resource* agent properties

Property	Description
rId	Resource ID
ma_k	Maximum availability
$ur_{k,t}$	Resource usage for each period
$pr_{k,t}$	Processing Capacity (%) for each period
rType	Resource type: *local* or *global*
R_u_k	Payoff function for resource agent ***k*** defined as: $max \sum_{i=1}^{n} ral_{ik}$

4.2. Learning by Trial-and-Error

One of the distributed learning techniques having as its main characteristic resorting to uncoupled learning rules for each individual agents is the proposal of Young [16]. In learning by trial-and-error, interacting agents respond to changes in their own rewards, which are affected, indirectly, by other agents' actions. Uncoupled learning rules can be implemented in environments where the agents cannot observe what the other agents might be doing. That is why this learning method has great potential for distributed optimization problems and complex adaptive systems involving many autonomous agents interacting strategically.

In this work, the trial-and-error learning rule is incorporated in each agent to obtain the best course of action to be followed through a simple implementation at each decision stage in the repeated game (presented in the previous sub-section) to solve the multi-project (re)scheduling problem and to obtain a feasible schedule. Each interacting agent's state (z_i) is made up of a type of *mood* m_i (the four possible moods are *content*, *discontent*, *hopeful* and *watchful*), a reference action ($\bar{a}_i$) and a reference reward ($\bar{u}_i$). The mood is a characteristic that defines the agents' reaction to its experience with the environment. According to his mood and the obtained reward, each agent decides which is the next action to be applied. The parameter ε defines the rate of exploration to be used in the next game stage, where $\varepsilon \in (0,1)$. Thus, an agent will explore with probability ε and will not explore with probability $1 - \varepsilon$. [16]. In Table 3, state variables for the implementation of decoupled learning rules in each agent are shown.

Table 3. Agents' state variables related to *Learning by trial-and-error* rules.

Agents' State variables	Description
$z_i = (m_i, \bar{a}_i, \bar{u}_i)$	Reference state of agent.
$z_i = (m_i, a_i, u_i)$	Current state of agent.
ε	The exploration/exploitation rate of each agent

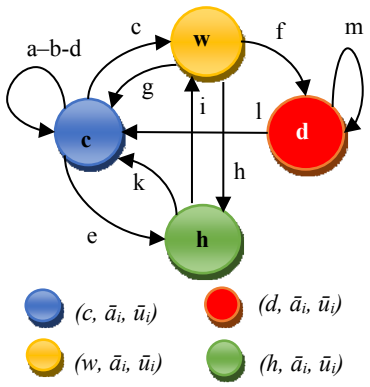
At the end of each game stage, once all the agents define their actions to follow, they simultaneously collect the stage payoffs according to the selected actions and agents' state transitions occur. *Project* agents select the actions that minimize the time, cost, and variations in the use of resources for the processing of its managed object, whereas *resource* agents select the actions that maximize their profits without exceeding their maximum availability for each period, according to their payoff functions, *sub-section 4.1*. The transitions between the different states that an agent can experiment during the simulation of the agent-based model are depicted in Fig. 1. The transitions "a" to "k" depends on agents' actions and payoffs only, while transitions "l" and "m" depends on a probability function called *response function*, which is monotonically increasing in u_i and monotonically decreasing in $\bar{u}_i$ for project agents, and conversely for resource agents. These transitions in every possible mood are depicted as follows:

- *Content*: If at stage *m* agent *i* is "content" at the next stage ($m + 1$) it may choose its reference action $\bar{a}_i$ with probability ($1 - \varepsilon$) and experiments a new action a_i with probability

Ɛ. If the agent i decides to experiment, it evaluates the utility u_i associated with a_i as follows: if $u_i < \bar{u}_i$, the agent's state remains $(c, \bar{a}_i, \bar{u}_i)$; otherwise, with probability Ɛ its state change to $(c, a_i, u_i(a_i))$, where a new action and utility benchmark are set out.

- *Hopeful-Watchful*: If the agent i perceives an increment or a decrement in its utility following its reference action ($\bar{a}_i$), then the mood become "hopeful" or "watchful" according to rules f to k defined in Fig. 1.

- *Discontent*: If player i is "discontent" it experiments a new action a_i. Thus, considering the probability *response function*, the mood turns out "content", and its new state is $(c, a_i, u_i(a_i))$. Otherwise, the state of the *ith* agent remains $(d, \bar{a}_i, \bar{u}_i)$.



Id	Initial State	Transition conditions	Resulting state
a	$(c, \bar{a}_i, \bar{u}_i)$	$a_i \neq \bar{a}_i$, $u_i(a) \leq \bar{u}_i$	$(c, \bar{a}_i, \bar{u}_i)$
b		$a_i \neq \bar{a}_i$, $u_i(a) > \bar{u}_i$	$(c, a_i, u_i(a))$
c		$a_i = \bar{a}_i$, $u_i(a) < \bar{u}_i$	$(w, \bar{a}_i, \bar{u}_i)$
d		$a_i = \bar{a}_i$, $u_i(a) = \bar{u}_i$	$(c, \bar{a}_i, \bar{u}_i)$
e		$a_i = \bar{a}_i$, $u_i(a) > \bar{u}_i$	$(h, \bar{a}_i, \bar{u}_i)$
f	$(w, \bar{a}_i, \bar{u}_i)$	$a_i = \bar{a}_i$, $u_i(a) < \bar{u}_i$	$(d, \bar{a}_i, \bar{u}_i)$
g		$a_i = \bar{a}_i$, $u_i(a) = \bar{u}_i$	$(c, \bar{a}_i, \bar{u}_i)$
h		$a_i = \bar{a}_i$, $u_i(a) > \bar{u}_i$	$(h, \bar{a}_i, \bar{u}_i)$
i	$(h, \bar{a}_i, \bar{u}_i)$	$a_i = \bar{a}_i$, $u_i(a) < \bar{u}_i$	$(w, \bar{a}_i, \bar{u}_i)$
j		$a_i = \bar{a}_i$, $u_i(a) = \bar{u}_i$	$(c, \bar{a}_i, \bar{u}_i)$
k		$a_i = \bar{a}_i$, $u_i(a) > \bar{u}_i$	$(c, \bar{a}_i, u_i(a))$
l	$(d, \bar{a}_i, \bar{u}_i)$	prob ϕ $(u_i(a), \bar{u}_i)$	$(c, \bar{a}_i, u_i(a))$
m		prob 1-ϕ $(u_i(a), \bar{u}_i)$	$(d, \bar{a}_i, \bar{u}_i)$

Fig. 1.State transitions experienced by agents using decoupled learning rules [16].

Fig. 2 shows illustrative project agents' state transition examples at different stages (***s*** and ***s*+1**) of the proposed non-cooperative game applying the learning trial-and-Error rules depicted in Fig.1. The example shows three project agents and three resource agents that participate in the game, where the state of each project agent is shown below it, denoted by $z_i = (m_i, \bar{a}_i, \bar{u}_i)$. The different *moods* of project agents are represented by different colors: blue, red, yellow and green for *content*, *discontent*, *watchful* and *hopeful* mood, respectively.

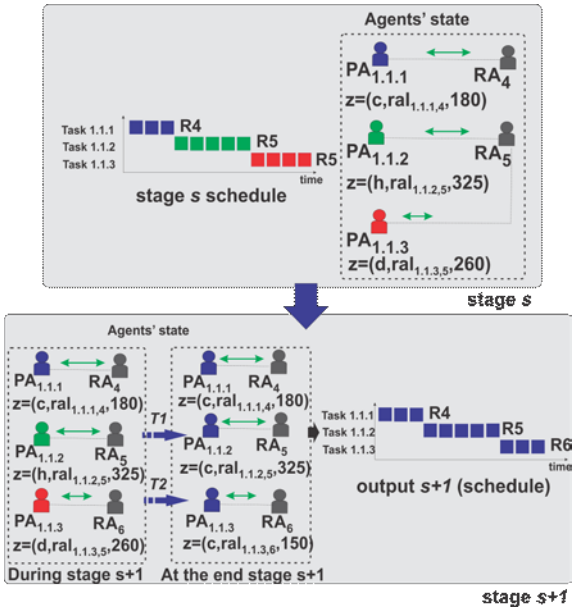


Fig. 2. State transitions experienced by agents using decoupled learning rules.

For simplicity of Fig. 2, the resource agents' state transitions are not shown. The result of the actions taken by the agents are the assignments *project-resource*, which are designated as $ral_{i,k}$, where the sub-index ***i*** and ***k*** represents the project agent and the resource agent, respectively.

In Fig. 2, two state transitions are depicted. The state transition *T1* is described for the project agent $PA_{1.1.2}$, which has the state $z_{1.1.2}$= (h, $ral_{1.1.2,5}$, 325) in the stage ***s***. During the stage ***s*** + 1, the agent selects its reference action $ral_{1.1.2,5}$ (*exploitation* strategy). This action implementation causes the state transition *T1*, where the new state of $PA_{1.1.2}$ switch to $z_{1.1.2}$= (c, $ral_{1.1.2,5}$, 325), with content *mood* and payoff equal to the reward obtained in the previous stage (state transition "j", Fig. 1). The second state transition rule considered is named *T2,* which is specified for the project agent $PA_{1.1.3}$ that selects one action that differ from its benchmark action during stage ***s*** + 1. The agent's state changing from $z_{1.1.3}$= (d, $ral_{1.1.3,5}$, 260) to $z_{1.1.3}$= (c, $ral_{1.1.3,6}$, 150), after the evaluation of response function that is calculated in terms of the obtained and the reference payoff (state transition "l", Fig. 1).

5. Implementation

The Multi-agent model was implemented in Netlogo, a multi-agent simulation environment for agent-based modeling that allows generating emergent behaviors resulting from on-going interactions among autonomous, learning agents [17]. As a consequence of the initial concurrent

projects scheduling received as input to the multi-agent model, the global (re)schedule of such projects is obtained in an autonomic and decentralized manner, so as to accomplish the scheduling goals defined in *Section 3*. The multi-agent model presented in this work, named as *Multi-agent model for a Project-oriented Fractal Organization* (MAS-MPR), is based on the prototype presented in [18]. This prototype has been extended to incorporate new goals based on additional project constraints such cost and resource leveling (since only the project duration was considered in [18]), and to eliminate assumptions defined during the initial implementation, such as that a resource can only process one task at a time.

In the virtual simulated world implemented using Netlogo, each estimated project schedule is graphically depicted by means of a Gantt diagram, on which changes will be made according to each agent action during interactions (Fig. 3). At each stage, agent decisions define the multi-project schedule, which is recorded as the output of such stage.

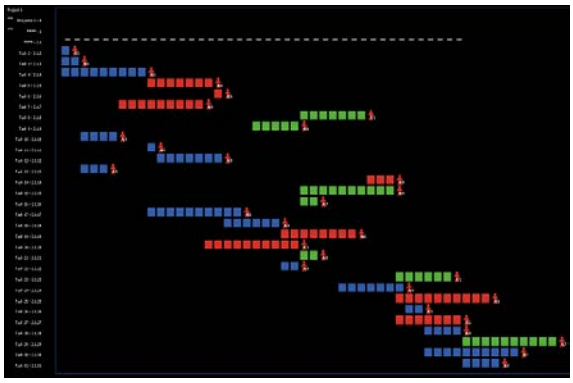


Fig. 3. Example of a project schedule obtained of MAS-MPR simulation as result of one stage.

5.1. Evaluation metrics

The simulated response and performance using MAS-MPR can be assessed through domain indicators (project total duration and cost, leveling resources) based on obtained agent's payoffs after simulation execution, which can be used by project leaders to assess the multi-project (re)scheduling problem generated by unplanned events, thus providing a tool to find an optimal decentralized solution in which all agents have no incentives to deviate, hence is a Nash equilibrium [19].

The scheduling efficiency for each project in the solution generated by the MAS-MPR is evaluated using a performance measure defined as the *Average Project Delay* (APD) [10], which in this work is calculated as follows:

$$APD = \frac{1}{M}\sum_{i=1}^{M}(ft_i - edd_i) \quad \text{(Eq. 5)}$$

where ft_i is the finalization time of the latest task in the generated schedule for project ***i***, edd_i is the estimated due date for project ***i*** (considered as the critical path for the project), and M represents the number of game stages played.

In addition, resource leveling is accounted for the solution generated by the MAS-MPR through a metric that determines the average usage of the different shared or global resources during the execution of the multiple projects. This indicator is defined as *Average Usage Global Resource Variability* (AUGRV) and calculated as follows:

$$AUGRV = \frac{1}{M}\sum_{k=1}^{L}\left(\frac{\sum_{t=1}^{T}\left(\frac{ur_{k,t}-ur_{k,t-1}}{ma_k}\right)}{T}\right) \quad \text{(Eq. 6)}$$

where $(ur_{k,t} - ur_{k,t-1})$ and $(ma_{k,t})$ represent, respectively, the variation of resource usage at two consecutive periods (t and t-1) and maximum resource availability for each period; T represents the project total duration and M represents the number of game stages played.

Finally, a specific metric is proposed to evaluate the social welfare of the agents that make up the MAS-MPR in a given stage ***s*** (equilibrium of the Model), considering that a *state of equilibrium* is one in which all the players are *content* and select their benchmark actions, bringing the system to a Nash equilibrium [19]. This metric is defined as:

$$ES_s = \frac{1}{PG}\sum_{pg=1}^{PG}\left[\frac{nContentAgents_{s,pg}}{nTotalAgents_{s,pg}}\right] \quad \text{(Eq. 7)}$$

where PG represents the set of all games played, *nContentAgents* is the number of content agents at the end of stage ***s*** in the game ***pg***, and *nTotalAgents* represents the total number of agents in the MAS-MPR during the stage ***s*** of game ***pg***.

The metrics defined in Eq. (5) and Eq. (6) are calculated at the end of the simulation of each game, while the metric defined in Eq. (7) is obtained considering all games played.

6. Computational Results

This section presents and discusses results obtained through different simulations of the *Multi-agent model for a Project-oriented Fractal Organization* (MAS-MPR) developed to solve the multi-project (re)scheduling problem in enterprise networking, considering the RLP. These results allow assessing the response and performance of the MAS-MPR.

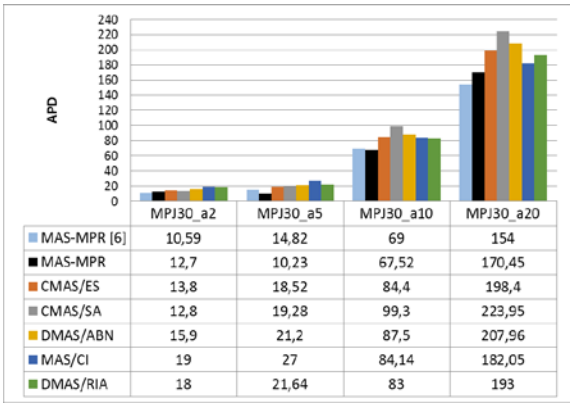
Instances of multi-project problems with different features are used to test the proposed multi-agent system (Table 4). These problem instances are

available in http://www.mpsplib.com. To adapt the problem instances to the simulation problem, different complexity levels (L1, L2, L3) are defined for each problem, describing the number of project agents (managing either a project, sub-project or task) that interact at each level of the fractal hierarchy. Then, four resource agents are available for each project, divided into shared (G) and local (L). The number of global resources can vary from 1 to 4. The maximum availability and processing capacity of resources remains unchanged throughout the simulation run of each problem instance. In the scheduling phase of the MAS-MPR simulation, 10 games per problem instance were considered, and 10 stages for each game are defined. Thus, the overall number of experiments performed in this work is 400. The simulations were executed on a computer Intel Core I5 (2.5GHz, 8GB RAM).

The APD values (Eq. (5)) obtained through the MAS-MPR simulation are shown in Fig. 4, together with the values for the different proposals presented in [6, 10, 11]. The project leader can analyze the solutions provided by MAS-MPR, and assess each problem instance to obtain the best solution, i.e., the schedule with the minimum APD value.

Table 4.Multi-project problem instances

Set of Instances	L1	L2	L3	N. resources	
				G	L
MPJ30_a2	1	2	30	1;2;3	1;2;3
MPJ30_a5	1	5	30	1;2;3	1;2;3
MPJ30_a10	1	5	30	2;3	1;2
MPJ30_a20	1	5	30	2	2



	MPJ30_a2	MPJ30_a5	MPJ30_a10	MPJ30_a20
MAS-MPR [6]	10,59	14,82	69	154
MAS-MPR	12,7	10,23	67,52	170,45
CMAS/ES	13,8	18,52	84,4	198,4
CMAS/SA	12,8	19,28	99,3	223,95
DMAS/ABN	15,9	21,2	87,5	207,96
MAS/CI	19	27	84,14	182,05
DMAS/RIA	18	21,64	83	193

Fig. 4. Results of APD obtained by different proposals.

The obtained results in Fig. 4 vividly demonstrate the advantages of incorporating decoupled learning to the interacting agents in the resolution of the multi-project (re)scheduling problems, comparing against the results obtained in other proposals. From Fig. 4 it can be seen that the MAS-MPR obtains better results for the four subsets of presented problems. These results correspond to the objective evaluation defined as minimization of the project total duration, considering the most efficient resource consumption without increasing the prescribed makespan, and avoiding resource over-allocation conflicts. Particularly, some results obtained in this work show higher APD values than those presented in the previous version of MAS-MPR [6]. This is because the analyzed sub-sets of problems present a high consumption of resources shared by period and when redefining the goal related to the RLP problem (Eq. (3)), considering the maximum availability of global resources per period to avoid over-allocations, schedules with greater total duration of the project are obtained.

The results obtained after the evaluation of the metric defined as AUGRV for the four problem sub-sets are given in Fig. 5. As shown in this figure, the variability in the use of resources shared (global) during the multi-project execution between periods (t and $t + 1$) is less than 12 % in all the problems considered, which highlights that the solutions obtained by MAS-MPR present an efficient use of shared resources, considering that the average use of global resources is over 60% for all simulated examples, avoiding resource idle time.

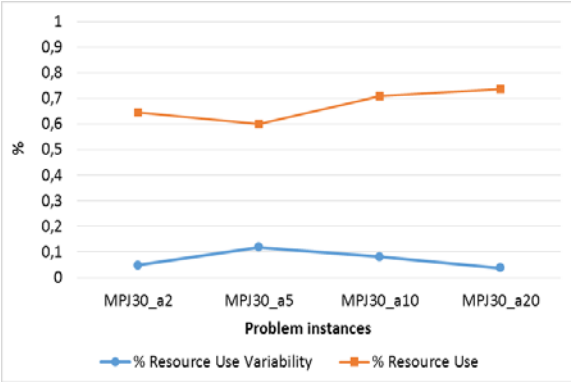


Fig. 5. Results of AUGRV metric for the simulated problem instances.

The last proposed metric calculates the average number of agents having a "*content*" mood in each stage for different executions of MAS-MPR for each problem subset (Fig. 6). This reflects that most of the agents that make up the MAS-MPR are *content* during the different game stages and select their benchmark actions, bringing the strategic interaction game to a near Nash equilibrium.

Additionally, the MAS-MPR provides other results to evaluate the different solutions obtained. In Fig. 7, the project total duration and costs after a repeated game simulation, as well as the average utilization of global resources by game stages are shown. To facilitate calculating the costs of the resulting project schedule, different processing costs were assigned to the shared resources. Thus, the project leader can choose the solution that is closest to the estimated duration and cost, and one that presents an efficient use of resources shared. For example, the global schedule obtained after a

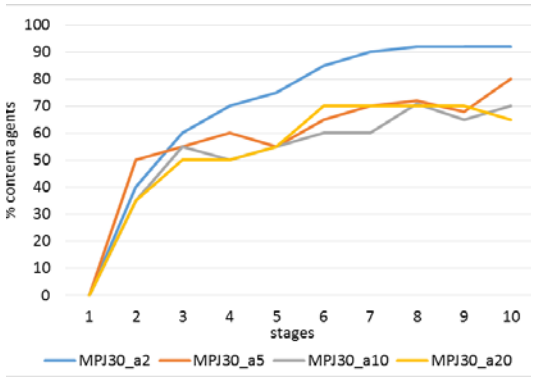
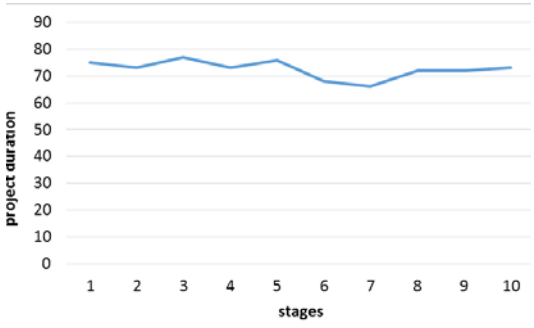
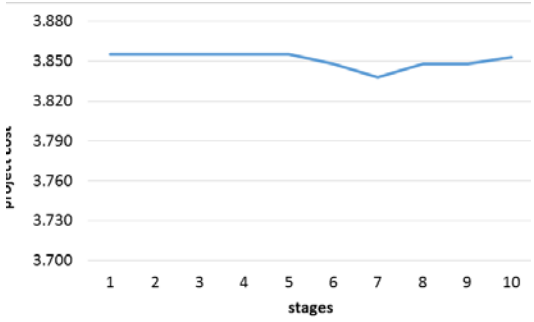


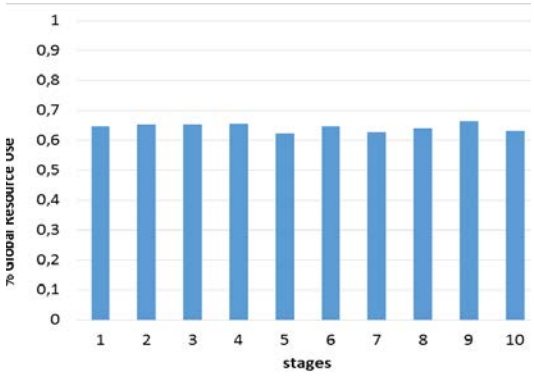
Fig. 6. Results of ES metric for the simulated problem instances.



(a) Project Total duration



(b) Project Total Cost



(c) Global Resources Average Use

Fig. 7. Project Scheduling goals in a simulated game of MAS-MPR

simulation of one problem instance with two projects, in the stage 7 presents the minimum project duration (Fig. 7(a)), whereas in the stage 6 presents the lowest project cost (Fig. 7(b)).

In addition, it can be observed in Fig. 7(c) that the average usage of global resources is almost constant in each game stage, not presenting great variations, which means that the MAS-MPR is able to obtain schedules with an efficient resource usage, avoiding over-allocation conflicts. Thus, the project leader can analyze these three variables calculated in each stage, namely time, cost, and global resources average usage and then choose the generated schedule that fulfills better project goals while complying with all related constraints.

7. Conclude Remarks

Multi-project (re)scheduling is considered a critical problem for organizations, mainly in those that share resources and execute inter-dependent projects. In ideal situations, these resources are unlimited and available as required. In contrast, resources are generally available in limited quantity and the project leader needs to level out the use and consumption of shared resources, since an efficient use of them is essential to achieve the success of achieving multiple goals. To this aim, it is necessary to consider the Resource Leveling Problem (RLP) whose specific concern is to achieve the most efficient resource consumption possible, without compromising the other project constraints and objectives. In this work, a new objective to be considered by the interacting agents that allow to level the use of shared resources by multiple projects and to avoid over-allocation problems has been incorporated into the Agent-Based Model presented in [6]. Thus, the proposed MAS-MPR, which is defined as a non-cooperative and multi-stage game, allows achieving a trade-off among the presented project scheduling problems caused by the restrictions of resources shared, obtaining emergent and feasible schedules through learning by trial-and-error, leading to a Nash equilibrium. This equilibrium is considered as the solution to the proposed game, where the interacting agents do not need to know the structure of the game, nor the actions or strategies of the other agents, or even that they are participating in such game.

The fractal feature of the MAS-MPR gives more flexibility and adaptability to response to unplanned events, allowing the emergence of solutions to the multi-project (re)scheduling problem in a distributed and decentralized way without limiting the number of resources or levels in the project hierarchy (projects, sub-projects, tasks) of the considered problem instances. Thus, the MAS-MPR allows learning to be incorporated in all agents in the same way, through simple and decoupled rules, regardless of the fractal level in which they interact.

The results obtained through Agent-based simulation in the Netlogo language show indicators that permit the analysis of the project total duration, cost, leveling resources, etc., and provided a quality evaluation of the generated schedule. These results highlight that the proposal, even incorporating new restrictions related to the RLP, obtains better solutions than other competing approaches.

Current research work is about incorporating the MAS-MPR in project management tools with a user-friendly interface supporting the decision-making process and allowing training activities for project leaders that favor seeking a solution to solve the project (re)scheduling problem in the framework of game theory and multi-agent learning. Furthermore, the agents' payoff functions will be defined as weighting functions, so that the project leader will define the importance of the project constraints that are considered in these functions, such as time, cost, resource leveling.

Acknowledgements

Financial support for this research study was granted by "Planificación automática en sistemas cognitivos de producción integrando aprendizaje por refuerzos con abstracciones lógicas y relacionales" PID 25/OR01 and "Evaluación dinámica de la calidad de procesos y productos para la toma de decisiones tempranas en industrias de base tecnológica" PID UTN3581 of Universidad Tecnológica Nacional.

Competing interests

The authors have declared that no competing interests exist.

References

[1] M. Garey, D. Johnson, *Computers and Intractability: A Guide to the Theory of NP Completeness*, New York, USA: W. H. Freeman, 1979.

[2] Project Management Institute, *Guía de los Fundamentos para la Dirección de Proyectos (Guía del PMBOK)*, 5ª ed., Project Management Institute, Pensilvania, USA, 2013.

[3] J.L. Ponz-Tienda, V. Yepes, E. Pellicer and J. Moreno-Flores, "The Resource Leveling Problem with multiple resources using an adaptive genetic algorithm", *Automation in Construction*, vol. 29, pp. 161–172, 2013.

[4] M. Vanhoucke, *Integrated Project Management Sourcebook – A Technical Guide to Project Scheduling, Risk and Control*, Springer, 2016.

[5] M.Canavesio and E. Martinez, "Enterprise modeling of a project-oriented fractal company for SMEs networking", *Computers in Industry*, vol. 54, pp. 794–813, 2007.

[6] L. Tosselli, V. Bogado and E. Martínez, "An agent-based simulation model using decoupled learning rules to (re)schedule multiple projects", in *XXIII Congreso Argentino de Ciencias de la Computación*, pp. 33-42, 2017.

[7] J. Homberger, "A (μ, λ)-coordination mechanism for agent-based multi-project scheduling", *OR Spectrum*, vol. 34, pp. 107-132, 2009.

[8] A. Shahsavar, A. Najafi and S.T.A. Niaki, "Three self-adaptive multi-objective evolutionary algorithms for a triple-objective project scheduling problem", *Computers & Industrial Engineering*, Elsevier, vol. 87, pp. 4-15, 2015.

[9] X. Shen, L. Minku, R. Bahsoon and X. Yao. "Dynamic Software Project Scheduling through a Proactive-Rescheduling Method", *IEEE Transactions on Software Engineering*, vol. 42,no. 7, pp. 658–686, 2016.

[10] S. Adhau, M. Mittal and A. Mittal, "A multi-agent system for distributed multi-project scheduling: An auction-based negotiation approach", *Eng. Applications of Artificial Intelligence*, vol. 25, pp. 1738–1751, 2012.

[11] S. Adhau, M. Mittal and A. Mittal, "A multi-agent system for decentralized multi-project scheduling with resource transfers", *Int. J. Prod. Economics*, vol. 146, pp. 646–661, 2013.

[12] T. Wauters, K. Verbeeck, G. Vanden Berghe and P. De Causmaecker "A Multi-Agent Learning for the Multi-Mode Resource-Constrained Project Scheduling Problem", *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems*, pp. 1-8, 2009.

[13] J. Araúzo, J. Pajares and A. López-Paredes, "Simulating the dynamic scheduling of Project portfolios", *Simulation Modelling Practice and Theory*, vol. 18, pp. 1428-1441, 2010.

[14] Chaos Report. Standish Group. Available at https://www.infoq.com/articles/standish-chaos-2015.Accessed on 2017-07-07.

[15] Y. Shoham and K. Leyton-Brown, *Multiagent Systems-Algorithmic, Game-Theoretic, and Logical Foundations*, Cambridge University Press, 2009.

[16] H.P. Young, "Learning by trial and error", *Games and Economic Behavior*, vol. 65, pp. 626-643, 2009.

[17] S. Railsback and V. Grimm, *Agent-Based and Individual-Based Modeling: A practical Introduction*, Princeton University Press, 2012.

[18] L. Tosselli, V. Bogado and E. Martínez, "Un Enfoque de Sistemas Multiagente para la Gestión Ágil de Riesgos en la Compañía Fractal Mediante la (Re) Planificación de Proyectos", in *Conaiisi 2015*, Bs. As., Argentina, 2015.

[19] S. Bary, R. Pradelski and H.P. Young, "Learning efficient Nash equilibria in distributed systems", *Games and Economic Behavior*, vol. 75, pp. 882-897, 2012.