View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Agora University Editing House: Journals

INT J COMPUT COMMUN, ISSN 1841-9836
8(2):334-345, April, 2013.

Disaster Response Project Scheduling Problem: A Resolution Method based on a Game-Theoretical Model

O.C. Vásquez, J.M. Sepúlveda, M.D. Alfaro, L. Valenzuela-Osorio

Óscar C. Vásquez
1. Department of Industrial Engineering, University of Santiago of Chile
3769 Ecuador Avenue. Santiago, Chile.
2. Computer Science Laboratory (LIP6), Pierre and Marie Curie University
4, Place Jussieu, Paris, France.
E-mail: oscar.vasquez@usach.cl

Juan M. Sepúlveda, Miguel D. Alfaro
Department of Industrial Engineering, University of Santiago of Chile
3769 Ecuador Avenue, Santiago, Chile.
E-mail: juan.sepulveda@usach.cl, miguel.alfaro@usach.cl

Luis Osorio-Valenzuela
Department of Electrical Engineering, University of Santiago of Chile
3519, Ecuador Avenue, Santiago, Chile.
E-mail: luis.osoriov@usach.cl

Abstract: We present a particular disaster response project scheduling problem (DRPSP) motivated by Fukushima's nuclear accident of Japan in 2011. We describe the problem as $MPS; R, N|prec, d_n|\sum c_k f(r_k(\mathsf{S}))$ by using Hartmann and Briskorn scheme and formulate a mixed integer linear programming (MILP) model. Due to the NP-hardness of the problem, we propose a resolution method based on game theory. This method associates the DRPSP to a non-cooperative game model, such that game solution is a feasible solution of the problem. In order to explore the potential of the proposed model and the performance of the resolution method, computational experiments are carried out. The results of resolution method show on average, that the method finds a feasible solution with a difference of 15.44% with respect to optimal solution within one percent of the time required by the MILP over GAMS 22.7.2/CPLEX 11.0.
Keywords: Disaster response, project scheduling problem, resolution method, game-theoretical model.

1 Introduction

Worldwide, disasters still continue to cause human casualties, damage and destruction of infrastructure, instability in the economy and the order of the community. Nowadays, the problems of disaster operation management (DOM) are considered an important factor into policies of national security [1]; thus, it is relevant to understand how one can face them effectively and efficiently. In particular, there is increasing recognition of the need of applying of operations research (OR) methods in DOM, because it develops a scientific approach to help decision making before, during and after a disaster [2]. Indeed, DOM problems are considered as the opportunity to re-visit and re-define OR, being a likely and worthwhile growing area in the next 50 years [3].

DOM problems arise in four phases: mitigation, preparedness, response and recovery [4]. An issue of particular interest in the disaster response phase is the project scheduling problem, because it addresses a double dilemma: on the one hand, the urgent need for project completion in the minimum possible time in order to prevent casualties and to reduce the impact on the social, economic, and political order; on the other hand, the scarcity in the number of resources with limited availability to carry it out. Some recent works of this area developed in the literature

Copyright © 2006-2013 by CCC Publications

have been proposed by De Angelis et al. [5], Fukura et al. [6], Yan and Shih [7], Yan et al. [7], Zhang et al. [9], Córdova and Yanine [10].

In general, a disaster response project scheduling problem (DRPSP) is considered as a complex variation of the multi-mode resource-constrained project scheduling problem (MRCPSP), simultaneously addressing both scheduling and assignment [11]. In this work, we study a particular DRPSP motivated by Fukushima's nuclear accident of Japan in 2011, where the shortest completion time was needed in order to avoid fatalities due to radiation exposition, besides of minimizing the number of rescue workers exposed to radiation.

The paper is broken down as follows: in section two, the statement of the problem is exposed; in section three, a mixed integer linear programming (MILP) model is formulated; in section four, the resolution method is presented based on a game-theoretical model for the problem; in section five, the performance analysis of resolution method is made by computational experiments; and finally, in section six, the conclusions and directions for further research are given.

2 Statement of the problem

Our problem concerns a project scheduling with a completion time equal to the critical path length in order to prevent loss of lives and to reduce the impact on the social, economic, and political order. The processing time of activities is a function of the allocated resources, which have an available workload as stipulated by a technical constraint or some other condition. The goal is to determine a schedule of the project, where the number of resources is minimized. Formally, the problem can be understood as a resource-constrained project scheduling problem (RCPSP) and denoted as $MPS; R, N|prec, d_n|\sum c_k f(r_k(\mathsf{S}))$ by using Hartmann and Briskorn scheme [12].

$MPS; R, N$ denotes the case of multi-mode problem with doubly constrained resources [13], which means that the "consumed resource" will be at most to its available workload and the "used resource " will be one or zero in each time unit. $prec$ and d_n denote precedence relations among the activities and the project deadline constraint, respectively. $\sum c_k f(r_k(\mathsf{S}))$ denotes the problem objective as cost minimization of the resources in order to meet the project deadline, where c_k is the cost per unit of resource k and $f(r_k(\mathsf{S}))$ is a function of the consumption r_k of resource k in scheduling S. Note that minimization of resource costs can be understood as a general goal, since it may consider others objectives such as the the minimization of the resources or the minimization of any non-monetary characteristic associated to them (e.g., utilized energy, generated pollution, etc.).

Our problem is closely related to the *Multi-mode double resource-constrained time/cost trade-offs problem* in the deadline version [14]. However, our problem adds a new set of constraints on each double-constrained resource due to the use of dedicated resources [15]. In next section, we formulate a MILP model for the DRPSP described above.

3 MILP Model

The MILP model is defined as follows. A disaster response project processes a set of n activities with a corresponding precedence directed acyclic graph $G(V, A)$, where V is the set of nodes, and $A \subset V \times V$ is the set of immediate precedence constraints on the activities. It is noteworthy that G also includes two dummy "start" and "end" nodes, indexed by 0 and $n+1$. Each identical resource $k \in R = \{1, ..., m\}$ has an available workload W, a monetary or non-monetary cost rate $c_k = c$; and it can only perform one activity at a time. Without loss of generality, we adopt $c = 1$ and define $f(r_k(\mathsf{S}))$ equal to 1 if the resource k is used and 0 otherwise. Each project activity

$i \in V$ has a processing time p_i, without pre-emption. By definition p_0 and p_{n+1} are zero. The objective is to find a scheduling S of the project, which minimizes the resources and finishes it in a given completion time equal to the critical path length d, thus preventing casualties and reducing the social, economic, and political impacts.

Decision variables:

$$X_k = \begin{cases} 1 & \text{if resource } k \text{ is used} \\ 0 & \text{otherwise} \end{cases}$$

$$Y_{ik} = \begin{cases} 1 & \text{if activity } i \text{ is processed by resource } k \\ 0 & \text{otherwise} \end{cases}$$

$$U_{ij} = \begin{cases} 1 & \text{if activity } i \text{ can be processed before activity } j \neq i, \text{ by the same resource} \\ 0 & \text{otherwise} \end{cases}$$

CT_i = Completion time of activity i

Objective function:

$$\min \sum_{k=1}^{m} X_k \tag{1}$$

Subject to:

$$CT_i \leq CT_j - p_j \qquad \forall (i,j) \in A \tag{2}$$

$$CT_0 = 0 \tag{3}$$

$$CT_{n+1} = d \tag{4}$$

$$\sum_{i=1}^{n} Y_{ik} p_i \leq W X_k \qquad \forall k \tag{5}$$

$$\sum_{k=1}^{m} Y_{ik} = 1 \qquad \forall i \tag{6}$$

$$[(CT_j - p_j) - CT_i] + \varepsilon \leq N(U_{ij}) \quad \forall i, j \neq i, (i,j), (j,i) \notin A \tag{7}$$

$$[CT_i - (CT_j - p_j)] \leq N(1 - U_{ij}) \quad \forall i, j \neq i, (i,j), (j,i) \notin A, \tag{8}$$

$$Y_{ik} + Y_{jk} \leq 1 + (U_{ij} + U_{ji}) \quad \forall i, j < i, \forall k \tag{9}$$

where ε is a small number and N is a large number. The objective function (1) minimizes the number of resources used for the project. Constraint (2) defines the finish-start type of precedence relations between the activities. Thus, if for starting activity j it is required that activity i have been completed, then the processing time of activity j minus the completion time of activity j should be greater than or equal to the completion time of activity i. Constraints (3) and (4) allow the fictitious activities of the project to be incorporated so that they generate the scheduling of the real activities. Thus, constraint (3) determines the project start time, while constraint (4) imposes a duration deadline for the project. Constraint (5) indicates that if a

machine is used, then the machine usage should be less than or equal to its workload available. Constraint (6) establishes that an activity can be processed by only one machine. Constraints (7), (8) and (9) are of the on-off type and ensure that if a resource is processing a given activity, then it does not process another one at the same time. In the case of constraint (7), an ε is incorporated that allows approaching the case in which an activity completion time is equal to the start time of another one. It is important to point out that for all U_{ij} it is true that $U_{ij} + U_{ji} \leq 1$. This inequality avoids duplicity in the processing of the real activities, because if activities i and j can only be processed by machine k, they can be only processed in one sequence, i.e. i and j or j and i.

It is easy to see that the existence conditions for a feasible solution are: the available workload of resources W is greater than or equal to the maximum processing time of activities and the number of resources available m is equal to the number of activities n. In what follows, we assume these feasibility conditions.

4 Resolution method

The development of a resolution method is based on the computational complexity of the problem. In fact, our problem is NP-hard by using a trivial reduction from the multi-mode resource-constrained discrete time-cost tradeoff problem [16]. Unfortunately, this NP-hard property implies that even modest instances of the problem become intractable in practice, which is specially expensive in disaster situations, where the decisions must be made quickly for preventing casualties and reducing the social, economic, and political impacts.

Our resolution method is based on a game-theoretical model. This method associates the proposed DRPSP to a non-cooperative game model, such that game solution is a feasible solution of DRPSP. In the game, the activity or a set of them are the players with a utility, which minimizes the remaining workload of the resource where it is performed; and the strategies are to associate or not with other player in order to execute them in a specified schedule on the same resource. In order to guarantee the precedence and workload constraints, a price mechanism is defined. The solution concept will be a Nash equilibrium in pure strategies (PNE), which always exists by definition of the game.

The resolution method generates new players, that is, a new set of activities from the PNE. These new players are integrated to other new game. This process is repeated until new players can not be generated. The description of the game, the algorithms and convergence of the method are presented next.

4.1 A game-theoretical model.

Players

Players are activities or a set of them. Each player is characterized by α/Ω, where $\alpha = \{1, 2, 3\}$ corresponds to the "type of player" and Ω is the set of activities. By definition a player type $\alpha = 1$ represents a project activity (e.g., activity 1 is defined by $1/\{1\}$). Players of type 2 and 3 are formed by two players, a "predecessor" of features $\alpha\prime/\Omega\prime$ and "successor" of features $\alpha\prime\prime/\Omega\prime\prime$; and a set of activities $\Omega = \Omega\prime \bigcup^* \Omega\prime\prime$, where $\bigcup^*$ is the ordered union, i.e., if $\Omega\prime = \{i, .., l\}$ and $\Omega\prime\prime = \{j, ..., s\}$, then $\Omega = \Omega\prime \bigcup^* \Omega\prime\prime = \{i, .., l, j, ..., s\}$. Both players type 1 and type 2 are differentiated by the time between the completion time of all activities of the "predecessor" player and the start time of all activities of the "successor" player. Figure 1 illustrates the difference between a player of type 2 and type 3. In the case of type 2 player, it shows that the time of completion of all activities of the predecessor player with $1/\{1\}$ is equal to the start time of

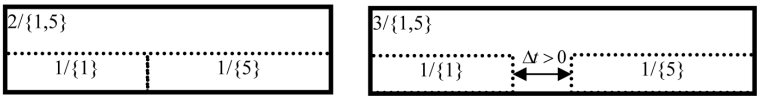


Figure 1: Difference between players of type 2 and 3.

all activities of the successor player with $1/\{5\}$. Whereas in the case of the player type 3, it has a time equal to $\Delta t > 0$ between each activity. Notice that the players of type 1 and 2 are determined only by the completion time of all activities of the predecessor player and the start time of all the activities of the successor player. Therefore, a player of type 2 could have $\Delta t > 0$ between some of its activities, because the predecessor and/or successor could be a player of type 3. Figure 2 shows an example of the situation.

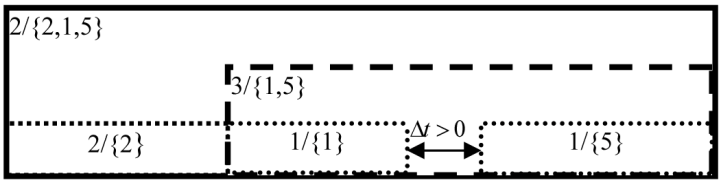


Figure 2: Player of type 2 conformed by a player of type 3.

Strategies

Consider a game with M players. Each player $i \in \{1, ..., M\}$ will have a set of strategies $S_i = \{1, ..., i-1, i, i+1, ...M, S\}$. The strategy $s_i = \{1, ..., i-1, i+1, ...M\}$ means that the player i want to be after player s_i in a schedule on a resource, shortly we say that the player i "buys" to player s_i. The strategy $s_i = i$ indicates that player i does not want to bind with any player in a schedule on a resource, shortly we say that the player i does not want to "negotiate" with any player. The strategy $s_i = S$ means that player i is available to be before any player in a schedule on a resource, shortly we say that the player is "placed on sale". If the player i decides to "buy"‘, then a "buying price" equal to maximum time to start all the activities of player i is calculated. On the other hand, if the player i decides to "sell", a "selling price" equal to the time of completion of all activities of the player i is calculated. For the calculation of prices, the free slack (FS_i) and the earliest start time (ES_i) of each activity i is needed, being easily obtainable through the Critical Path Method (CPM). Table 1 presents the definition of the buying and selling price for each of type of players.

Player	BP (buying price)	SP (selling price)
$1/\{i\}$	$ES_i + FS_i$	$ES_i + p_i$
$2/\{\Omega = \Omega\prime \bigcup^* \Omega\prime\prime\}$	$ES_i + \min(BP_{\alpha\prime/\Omega\prime} - ES_i, BP_{\alpha\prime\prime/\Omega\prime\prime} - SP_{\alpha\prime/\Omega\prime})$	$SP_{\alpha\prime/\Omega\prime} + (SP_{\alpha\prime\prime/\Omega\prime\prime} - ES_j)$
$3/\{\Omega = \Omega\prime \bigcup^* \Omega\prime\prime\}$	$ES_i + \min(BP_{\alpha\prime/\Omega\prime} - ES_i, BP_{\alpha\prime\prime/\Omega\prime\prime} - ES_j)$	$SP_{\alpha\prime\prime/\Omega\prime\prime}$
For the player type 2 and 3, i and j are the index of the first activity in set $\Omega\prime$ and $\Omega\prime\prime$, respectively.		

Table 1: Prices definition for types of players.

Note that if the “buying price" of player A is greater than or equal to the “selling price" of player B, then the set of activities of player A can start after the set of activities of player B. In practice, these prices allow to bind a set of activities, respecting the precedence constraints and

deadline of the project. Figure 3 presents a illustration of the buying and selling price for each of type of players.

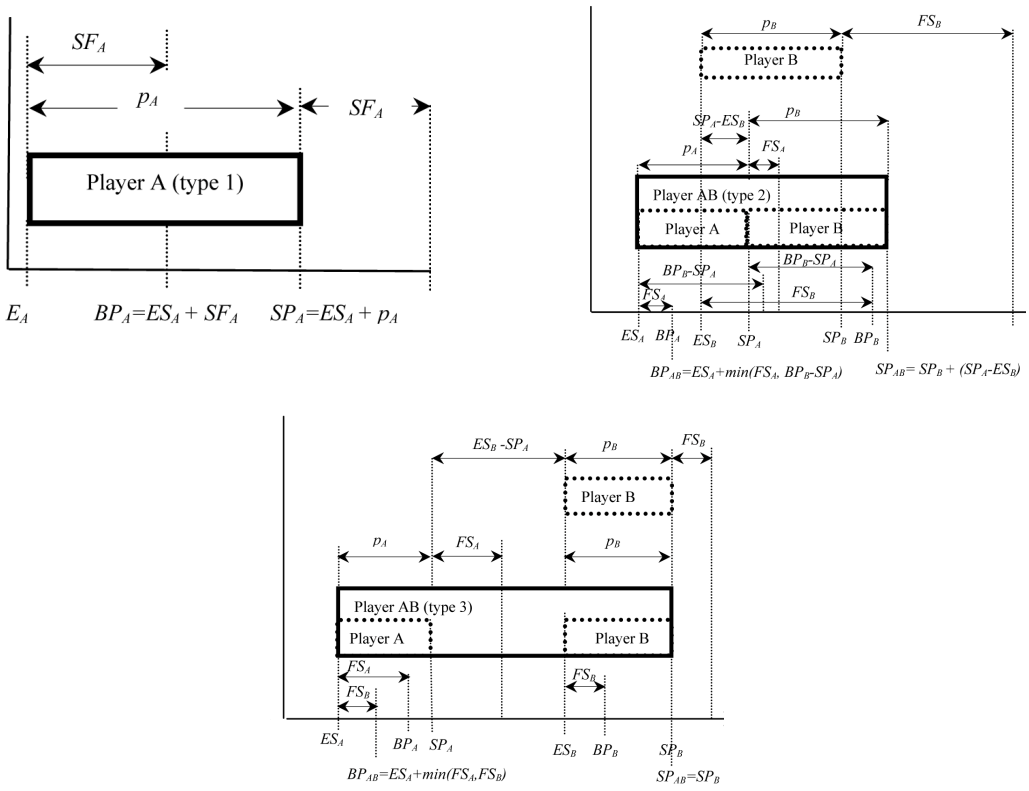


Figure 3: Price illustration of types of players.

Payoffs

The payoff of each player $i \in \{1, ..., M\}$ depends on his strategy and the strategies of other players. The notation is $r_i(s_i, s_{-i})$ where $s_i \in S_i$ and $s_{-i} \in S_{-i} = \bigotimes_{j \neq i} S_j$. The definition of payoff for player i of feature α/Ω is presented in (10)

$$r_i(s_i, s_{-i}) = \begin{cases} \sum_{w \in \Omega} p_w & \text{if } \forall s_i = i,\ s_{-i} \in S_{-i} \\ \sum_{w \in \Omega \bigcup^* \Omega\prime} p_w & \text{if } \sum_{w \in \Omega \bigcup^* \Omega\prime} p_w \leq W, \Omega \cap \Omega\prime = \varnothing \text{ and} \\ & \text{a)} s_i = j, s_j = S, BP_i \geq SP_j, \forall s_k \neq j (k \neq i, j) \text{ or} \\ & \text{b)} s_i = S, s_j = i, BP_j \geq SP_i, \forall s_k \neq i (k \neq i, j) \\ & \text{where } \Omega\prime \text{ is the set of activities of the player } j \\ -\infty & \text{otherwise} \end{cases} \tag{10}$$

Note that the payoff will be different to minus infinity if and only if the precedence constraints, critical path time as deadline project and the available workload of the resource are satisfied.

In fact, the relation of prices guarantee the precedence and deadline constraints; the relation of the sum of processing time guarantees satisfaction of workload availability of resources, and the relations between the activity set of player avoids the duplicity of activities. In the case $s_i = i$, the payoff value will always be different to minus infinity by definition.

Solution concept

In this game, the interaction among players is modeled as non-cooperative and the solution concept is a Nash equilibrium in pure strategies (PNE). The set of PNE of any game is always nonempty, because the strategy profile where all players use the strategy "to negotiate" with no player is a PNE by definition. Thus, any defined game has at least a PNE.

4.2 Algorithms

We define three algorithms, called ALGO 1, 2 and 3 respectively, which define our resolution method. ALGO 1 finds a pure strategies profile for any game. Here, we prove that this strategies profile is always a PNE. ALGO 2 generates new players as a new set of activities from PNE generated by ALGO 1. Finally, ALGO 3 integrates ALGO 1 and ALGO 2 in a cyclic process, until new players cannot be generated. In practice, the final solution yielded by ALGO 3 will be the solution of the method.

We assume a game with M players and consider a matrix $\Gamma_{M\times M}$. Let Γ_{ij} be the payoff of player i and player j. The case Γ_{ij}, $i \neq j$ means that the strategy of row player i is to buy to column player j, the strategy of player j is placed on sale, and the strategies of all player $k \neq i \neq j$ is not to buy to player j. Note that by definition, both the payoff of player i and the player j are equal. The case of Γ_{ii} means the payoff of player i if his strategy is not “to negotiate" with any player. This matrix Γ has two properties: a) it contains all payoffs of each player that could be different to $-\infty$ by definition; and b) at least, Γ_{ii} for all i is different to $-\infty$ by definition.

Now, we define the ALGO 1 to find a strategies profile of the game.

Algorithm 2 ALGO 1
Data: Instance data of problem
Result: Strategies profile
Step 0: Generate Matrix Γ
Step 1: Select the maximum value of matrix Γ.
Step 2:
a) Assign the strategies of row and columns players of maximum value.
b) Fix the strategies to component players of the row-column players found.
Step 3: Redefine Γ such that
a) Delete the row and column equal to index row-column players found.
b) Delete the row and column equal to index of component players of row-column players found.
c) Delete the row and column equal to index of player, whose component players are some row-column players found.
Step 4:
If $\Gamma \neq \emptyset$ **then**
Return **Step 1**
End if
Step 5: Generate the strategies profile.

Theorem 1. *ALGO 1 finds a PNE for the proposed game.*

Proof: We consider a game with M players and a strategies profile found by ALGO 1. Without loss of generality, we assume that $2L$ players have a one-to-one correspondence between buy-sale strategies and $M-2L$ players are not no correspondence. Let $\{s_{i_n} = j_n, s_{j_n} = S\}_{n=1}^{L}$ be the notation for a correspondence between the "buyer" player i_n and the on "sale" player j_n. Let $\{s_{k_n} = k_n\}_{n=2L+1}^{M}$ be the notation for the case with no correspondence. In addition, we define the strategy index $p \in \{1, 2...M\}$. We show that the strategies profile found by ALGO 1 is PNE by definition, i.e., a unilateral deviation in strategy by any single player is not profitable for that player. Formally, we prove:

1. $r_{i_n}(s_{i_n} = j_n, s_{-i_n}) \geq r_{i_n}(s_{i_n} = i_n, s_{-i_n})\ \forall i_n$
2. $r_{i_n}(s_{i_n} = j_n, s_{-i_n}) \geq r_{i_n}(s_{i_n} = S, s_{-i_n})\ \forall i_n$
3. $r_{i_n}(s_{i_n} = j_n, s_{-i_n}) \geq r_{i_n}(s_{i_n} = p, s_{-i_n}), \forall p \neq j_n, i_n, S\ \forall i_n$
4. $r_{j_n}(s_{j_n} = S, s_{-j_n}) \geq r_{j_n}(s_{j_n} = j_n, s_{-j_n})\ \forall j_n$
5. $r_{j_n}(s_{j_n} = S, s_{-j_n}) \geq r_{j_n}(s_{j_n} = p, s_{-j_n}), \forall p \neq j_n, S\ \forall j_n$
6. $r_{k_n}(s_{k_n} = k_n, s_{-k_n}) \geq r_{k_n}(s_{k_n} = S, s_{-k_n})\ \forall k_n$
7. $r_{k_n}(s_{k_n} = k_n, s_{-k_n}) \geq r_{k_n}(s_{k_n} = p, s_{-k_n}), \forall p \in \{1, ..., M\} \backslash k_n\ \forall k_n$

For the conditions 1 and 4, we have a proof by a contradiction. Without loss of generality, suppose that the condition 1 and 4 are false for some players pair i_n and j_n with correspondence between buy-sale strategies and that ALGO 1 selected this correspondence in the step k. Now, we note that ALGO 1 in the step k fixed the "buy" and "on sale" strategies for the players, when they obtain a payoff value greater than the obtained payoff value for them with the "alone" strategy. Therefore, we have a contradiction with the assumption case. For conditions 2, 3, 5, 6 and 7, we have a proof by the definition of payoff. In fact, for the case 2 given that the "buyer" player $p \neq i_n$ with $s_p = i_n$ does not exist, then $r_{i_n}(s_{i_n} = S, s_{-i_n})$ is equal to $-\infty$. For the case 6, a similar argument is hold. For the case 3, if the player $p \neq j_n, i_n, S$ is "on sale", it is bought by another player, then $r_{i_n}(s_{i_n} = p, s_{-i_n})$ is equal to $-\infty$; else the player p is not "on sale", then the player p is a "buyer" player or a "alone" player, therefore $r_{i_n}(s_{i_n} = p, s_{-i_n})$ is equal to $-\infty$. For the case 5 and 7, we have a similar argument. □

The input of ALGO 2 considers a PNE of the game. This PNE can have one-to-one correspondences between buy-sale strategies of players. If there are one-to-one correspondences, then a new player is generated by each of them. These new players will have the above definition of prices, considering the "buyer" and "on sale" players.

Algorithm 3 ALGO 2

Data: PNE of game
Result: New players
Step 0:
While one-to-one correspondences exists in PNE **do**
a) Generate a new player and compute sale and buy prices, considering the "buyer" and "on sale" players.
b) Extract the players with one-to-one correspondences of PNE.
End while

Finally, we present the ALGO 3. We consider a sequence of games SG. Let PNE_k be the Nash Equilibrium of the game $k \in SG$. Let NP_k the new players generated in the game in the position k of sequence SG, shortly game k.

ALGO 3 generates a new game $k+1$, integrating each player of NP_k into the game $k+1$, if and only if this player is not in the game k. The game k will be the last game of SG if and only if no new player is integrated.

Algorithm 4 ALGO 3

Data: Instance data of problem
Result: Solution of instance
Step 0: Call to ALGO 1.
Step 1: Call to ALGO 2.
Step 2:
If New players generated are not in set of players k **then**
Incorporate the new players generated to game.
$k := k + 1$
Return **Step 0**
End if
Step 3: Generate the solution from PNE of game k.

5 Computational experiments

We analyze the resolution method performance by using computational experiments. For this, we generate four projects by the following process:

- We define four intervals for the number of activities of the project: 1-10, 11-20, 21-40 and 41-80. For each interval i we define a project P_i with a number of activity n_i by random way.
- We determine $\lceil \frac{n_i}{2} \rceil$ as maximum value for the processing time of activities of project P_i. For each activity of a project P_i, a integer value from the interval $[1, \lceil \frac{n_i}{2} \rceil]$ is assigned as processing time by random way.
- We fix three predecessor activities as maximum per activity. For each activity of a project, a number of predecessor activities is defined, and later, the predecessor activity(ies) is (are) assigned. Both processes by random way.
- For each project, we compute the number of feasible instances by changing the workload availability of resources within the feasible interval (i.e., W between the maximum processing time p_{max} and the critical path d_i of project P_i). Each instance is resolved in optimal way by using the MILP proposed. This MILP is programmed and run in GAMS 22.7.2/CPLEX 11.0 (Intel Core i3, 3.10 GHz, 4.00 GB RAM) for each instance into time window of two hours. Finally, we consider a total of 118 instances, in which the optimal value is obtained.

Table 2 shows the features of generated projects.

Project i	1	2	3	4
Activities	10	18	30	75
p_{max}	4	8	12	27
d	14	30	47	99
Instances	11	23	36	48

Table 2: Principal features of generated projects.

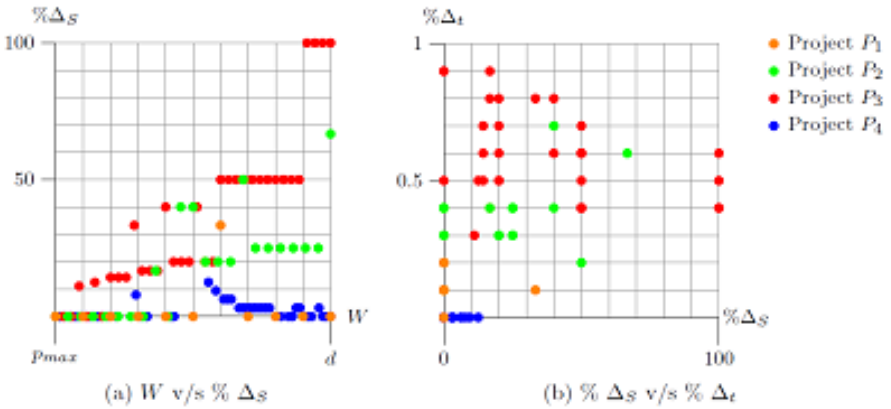
Once the instances are defined, the resolution method is programmed in C++ in the same hardware. In case of ties in the maximum value of the matrix, we adopt the priority rule: first row, first column.

The metric of analysis considers: the differences between the solution given by the proposed method resolution and the optimal solution (Δ_S) and, between the computation time in seconds

required by the optimal solution and proposed method (Δ_t). Also, we compute the percentages obtained by the ratios between the Δ_S and the optimal solution (% Δ_S), and between the computation time required by the optimal solution and proposed method (% Δ_t). The obtained results are shown in Table 3 and Figure 4.

	Δ_S	% Δ_S	Δ_t	%Δ_t
Average	0.85	15.44	744.46	0.25
Stand.Dev.	0.94	23.36	1,123.59	0.27
Minimum	0.00	0.00	0.22	0.00
Maximum	4.00	100.00	3,402.86	0.88
Median	0.00	3.13	2.08	0.15
Hits	54 (45.76 %)			

Table 3: Results of analysis.



(a) W v/s % Δ_S (b) % Δ_S v/s % Δ_t

Figure 4: Obtained results for projects.

Figure 4: Obtained results for projects.

6 Conclusions

In this paper we presented a particular disaster response project scheduling problem (DRPSP) motivated by Fukushima's nuclear accident of Japan in 2011. We described this problem by using Hartmann and Briskorn scheme [12] and represented it by a mixed integer linear programming (MILP).

By considering the NP-hardness of the problem, we developed a new resolution method based on game theory. This method ensures feasibility of the solution and convergence. The computational experiments yielded good results in comparison with the optimal solution, specially the computational times. On average, the method finds a feasible solution with a difference of 15.44% with respect to optimal solution within one percent of the time required by the MILP over GAMS 22.7.2/CPLEX 11.0. This computational speed is worth in disaster situations where decisions must be made quickly.

Further research is proposed in order to improve the resolution method results. In particular, analysis of initial data and generation of subroutines for diversification. The first subject is very important because it can give a best lower bound for the problem, whereas the latter fact allows to jump the local solutions and to improve the final result.

Acknowledgments

This research has been supported by DICYT (Scientific and Technological Research Bureau) of the University of Santiago of Chile (USACH), Project 061117SS and Department of Industrial Engineering. The authors wish to thank Dr. Nicolás Figueroa (PUC) and Christoph Dürr (UPMC) for their valuable comments for improving the game definition, to Dr. Pedro Palominos (USACH) and Dr. Richard Weber (UCH) for their valuable comments on the generation of the testing instances.

Bibliography

[1] Jackson B.A., K.S Faith., *Evaluating the reliability of emergency response systems for large-scale incident operations*, RAND Corporation monograph series, 2009

[2] Altay N., Green III W.G., OR/MS research in disaster operations management, *Eur J Oper Res*, 175(1):475-493, 2006.

[3] Simpson N.C., Hancock P.G., Fifty years of operational research and emergency response, *J Oper Res Soc*, 60(S1):126-139, 2009.

[4] Green III W.G., Four phases of emergency management, Electronic of Civil Defense and Emergency Management, 2002. Available from: https://facultystaff.richmond.edu/ wgreen/Ecd4phases.htm

[5] De Angelis V., Mecoli M., Nikroi C., Storchi G., Multiperiod integrated routing and scheduling of World Food Programme cargo planes in Angola, *Comput Oper Res*, 34(6):1601-1615, 2007.

[6] Fukura H., Ishibashi K., Nakatsu K., Hotta S., Optimal restoration scheduling of damaged networks under uncertain environment by using improved genetic algorithm, *Tsinghua Sci Techn*, 13(S1):400–405, 2008.

[7] Yan S., Shih Y.L., Optimal scheduling of emergency roadway repair and subsequent relief distribution, *Comput Oper Res*, 36(6):2049-2065, 2009.

[8] Yan L., Jinsong B., Xiaofeng H., Ye J., A heuristic project scheduling approach for quick response to maritime disaster rescue, *Int J Project Manage*, 27(6):620-628, 2009.

[9] Zhang L., Lin Y., Yang G., Chang H., Emergency resources scheduling based on adaptively mutate genetic algorithm, *Comput Hum Behav*, 27(5):1493-1498, 2011.

[10] Córdova F.M., Yanine F.F., Homeostatic control of sustainable energy grid applied to natural disasters, *Int J Comput Commun*, 8(1): 50-60, 2013.

[11] Rolland E., Patterson R.A., Ward K., Dodin B.; Decision support for disaster management, *Comput Oper Res*, 3(1-2):68-79, 2011.

[12] Hartmann S., Briskorn D., A survey of variants and extensions of the resource-constrained project scheduling problem, *Eur J Oper Res*, 207(1):1-14, 2010.

[13] Węglarz J., Project scheduling with continously-divisible, doubly constrained resources, *Manage Sci*, 27(9): 1040-1057, 1981.

[14] Zhang J., Shan H., Multi-mode double resource-constrained time/cost trade-offs project scheduling problems, In: *Proceeding of International Conference on Management and Service Science-MASS*, Wuhan/Beijing, China, 2009.

[15] Bianco L., Dell'Olmo P., Grazia Speranza M., Heuristics for multimode scheduling problems with dedicated resources, *Eur J Oper Res*, 107 (2):260-271, 1998.

[16] De P., Dunne J., Ghosh J.B., Wells C.E., Complexity of discrete time-cost tradeoff problem for project networks, *Oper Res*, 45(2):302-305, 1997.