



## **A new RCPSP variant for scheduling research activities in a nuclear laboratory**

Oliver Polo Mejia, Marie-Christine Anselmet, Christian Artigues, Pierre Lopez

### **► To cite this version:**

Oliver Polo Mejia, Marie-Christine Anselmet, Christian Artigues, Pierre Lopez. A new RCPSP variant for scheduling research activities in a nuclear laboratory. 47th International Conference on Computers

Industrial Engineering (CIE47), Oct 2017, Lisbonne, Portugal. 8p., 2017. <hal-01630977>

**HAL Id: hal-01630977**

**<https://hal.laas.fr/hal-01630977>**

Submitted on 8 Nov 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A NEW RCPSP VARIANT FOR SCHEDULING RESEARCH ACTIVITIES IN A NUCLEAR LABORATORY

O. Polo Mejia<sup>1,2\*</sup>, M-C. Anselmet<sup>1</sup>, C. Artigues<sup>2</sup>, P. Lopez<sup>2</sup>

<sup>1</sup>Service LECA-STAR (CEA/DEN/CAD/DEC/SLS) - CEA Cadarache  
St Paul lez Durance CEDEX, France

<sup>2</sup>LAAS-CNRS, Université de Toulouse, CNRS  
Toulouse, France

## ABSTRACT

There are not many research works dedicated to the optimization of scheduling activities in research laboratories. In this paper, we consider the weekly scheduling problem of activities within a nuclear research laboratory. We show that this problem can be modeled as a preemptive multi-skill variant of the Resource-Constraint Project Scheduling Problem where the preemption of some activities is allowed and induces a penalty every time the activity is preempted. We propose two integer linear programs to find exact solutions. Computational tests are carried out and described.

**Keywords:** RCPSP, PRCSP, MSRCSP, Scheduling, Nuclear laboratory, Optimization

## 1 INTRODUCTION

The applications of operations research for the optimization of research projects and experimentation activities are mainly related to project management, focusing mainly on the use of the PERT method and other techniques aimed at the planning of macro tasks. At the operational level, the scheduling of research experiments becomes a complex activity, as Mancel [1] explains in her thesis work, and the literature on this subject is rather sparse. The complexity lies mainly in the fact that the order and types of activities to be carried out during an experiment or a research project can vary enormously due to the results obtained in the early stages of the experiment.

Working with a relatively short scheduling horizon may reduce the subjectivity of the activities to be scheduled in a research project, thus allowing the use of an elementary activity approach, as proposed by Mancel [1]. In her approach, each experiment or research project consists of a series of basic tasks or activities that are well defined. This elementary task approach allows us to use standard scheduling methods to schedule the activities of the research laboratory. After analyzing the process of the studied laboratory, we conclude that the problem under consideration can be assimilated to the Resource-Constrained Project Scheduling Problem.

The Resource-Constrained Project Scheduling Problem (RCPSP) is a combinatorial optimization problem that covers a large number of "classical" scheduling situations. The problem consists in scheduling tasks or activities on renewable resources with limited capacities. These tasks are linked together by precedence relationships (task  $i$  can't start until task  $l$  is finished). The idea is to find a solution that minimizes the makespan of the project, while respecting both precedence constraints and resource constraints [2].

---

\* Corresponding Author: Oliver POLO MEJIA – E-mail address: [oliver.polomejia@cea.fr](mailto:oliver.polomejia@cea.fr)

Although the classic version of the RCPSP is very powerful, being able to model a large number of scheduling problems, it cannot cover all the situations that can happen in real-life situations. That is why researchers have developed more general versions of the RCPSP using the classic version as starting point. Surveys on this topic are proposed for example by Harmann et al. [3] and Orji et al. [4]. Among all these variants, we distinguish two that are of great interest for the modeling of the scheduling problem at hand: the preemptive RCPSP with penalty for preemption and the preemptive multi-skill RCPSP.

There are very few works dealing with preemptive RCPSP with penalty for preemption [5] or the preemptive multi-skill RCPSP [6]. For the best of our knowledge, the problem we propose, a Preemptive Multi-Skill RCPSP with penalty for preemption (PMSRCPSP), has not been addressed in the literature.

The remainder of the paper is as follows. In Section 2, we briefly describe the problem under consideration. In Section 3, we present two integer linear programming models to find an exact solution to the PMSRCPSP. Section 4 presents the characteristics of instances used to test the proposed models. In Section 5, we show the computational experiments carried out. Finally, in Section 6 we conclude and discuss future research.

## 2 PROBLEM DESCRIPTION

In this work, we are interested in optimizing the weekly scheduling of research activities carried out within a nuclear laboratory. The studied lab is a hot laboratory, which supports research programs on nuclear fuels irradiated in reactors. Typical activities of the lab are destructive and non-destructive examination on irradiated fuel, manufacturing of instrumented rods with irradiated fuel, and fuel reconditioning operations before fuel storage.

The classic version of the RCPSP assumes that, once started, an activity must run continuously until its completeness. However, in practice it may happen that an activity can be preempted to be finished later. The inclusion of preemption may lead to reductions in the total duration of the project, especially when resource availability is very limited. Including the preemption of activities increases the number of possible solutions and consequently the computational complexity of the problem [7]. In the Preemptive RCPSP (PRCPSP), activities can be preempted at discrete time periods and continued later without any additional cost [8]. In our case, preemption is important because of the absence of some resources during the night, which results in the impossibility of working continuously for some activities.

The possibility of resuming a preempted activity without any cost does not appear as realistic enough for some authors [5], [9], due mainly to the cost/time of setup for resuming or simply to the reduction of the production rate. A more realistic version can thus be conceived: the Preemptive RCPSP with penalty for preemption, where preemption is allowed but cost is incurred every time an activity is preempted. This variant is adapted to our specific context for convenience and safety reasons (technicians must put the working area in the safety configuration every time an activity is preempted).

Other assumption of the RCPSP is that each resource has a specific function, or in other words the resources are mono-skilled. This hypothesis can quickly become false when we are also interested in the allocation of human resources working in the project. In the considered application, a resource could perform several functions leading us to a multi-skill RCPSP (MSRCPSP). In the MSRCPSP, a resource is therefore characterized by the skill set it possesses; and a task is no longer only defined by the quantities required of each resource, but also by the number of resources with a specific competence [10]. In our project, the need to model human resources as multi-skilled resources is essential since the execution of some activities requires authorizations and training each technician may or not have.

Taking into account all the aforementioned aspects, and aiming at the most realistic modeling, we must then work with a variant of the problem mixing the characteristics of the MSRCPSP and the PRCPSP. In this new variant, which we called Preemptive Multi-Skill RCPSP with penalty for preemption, the resources are renewable, multi-skilled and limited in capacity. An activity is defined by its duration, precedence relationships and constant requirements of both resources and skills. Preemption is allowed but a penalty is applied every time an activity is preempted. Some specific conditions of the laboratory's operations are given below.

Most of the activities carried out by the laboratory can be considered as preemptive. However, there is also a group of activities ( $NP$ ) for which preemption is not possible due to technical and safety constraints. Each activity is characterized by its duration ( $D_i$ ), its requirements of resource  $k$  ( $Br_{i,k}$ ), its requirements of skill  $c$  ( $Bc_{i,c}$ ) and the minimum number of technicians needed to perform it ( $Nt_i$ ). Activities may have either a due date ( $dd_i$ ) or a deadline ( $dl_i$ ) (e.g. contractual activities or mandatory controls). Finally, activities might be also constrained by a release date ( $Dmin_i$ ).

The technicians ( $j$ ) work in 12-hour shifts, which allows us to schedule activities continuously from Monday to 7:00 until Friday 19:00 (108 hours). Each team is made up of four technicians, each with specific training and authorizations (skills) ( $CO_{j,c}$ ). We assume that each technician can manage only one activity for each period of time but can perform more than one skill on the same activity. The availability of resources ( $DR_{k,t}$ ) and technicians ( $DO_{j,t}$ ) changes over time.

### 3 MODELING

Traditionally, RCPSP modeling can be done using different approaches: using continuous time-based models based on flows [11], discrete-time integer linear programming (ILP) formulations, or event-based ILP formulations [12]. Among the discrete-time formulations, Artigues [13] presents an analysis of three well-known time-indexed formulations: pulse formulation, step formulation, and on/off formulation.

In the pulse formulation, the binary pulse variables  $X_{i,t}$ ,  $i \in V$ ,  $t \in H$ , are such that  $X_{i,t} = 1$  if activity  $i$  begins at period  $t$ . An activity that begins at  $t$  is to be interpreted as the fact that the activity is ongoing during the interval  $[t, t + 1]$  while it was not in progress at the time interval  $[t - 1, t]$  if  $t > 0$ .

The step formulation uses the binary step variables  $Z_{i,t}$  such that  $Z_{i,t} = 1$  if the activity  $i$  begins at time  $t$  or before. For a given activity, the variables  $Z_{i,t}$  with  $t <$  starting time of the activity are all equal to 0, whereas the variables with  $t \geq$  starting time are all equal to 1.

Finally, the on/off formulation uses binary variables  $Y_{i,t}$ , where  $Y_{i,t} = 1$  if activity  $i$  is in progress at time  $t$  and  $Y_{i,t} = 0$  otherwise. This formulation, which seems to be the most suitable for the preemptive case, is the basic formulation for designing the scheduling models presented in the following.

The two proposed models are similar in essence, formulation "on / off" is used in both cases and most restrictions are modeled in the same way. The main difference lies in the way in which the precedence, deadline, and release date constraints are modeled.

For the studied laboratory, we were able to identify three objectives that should be taken into account for designing our models and more precisely at the moment of choosing the objective function. First, we need to always try to respect the due dates of research activities, which means that we must minimize the tardiness. Second, as mentioned before, we have to minimize the number of time an activity is preempted for convenience and safety reasons. And third, in research activities it is interesting to schedule the units of duration of each activity as soon as possible, because in this way researchers can quickly have data to work with and even decide whether or not to continue/modify their experiments according to the partial data. For our problem, we propose to

convert the multiple objectives optimization problem into a single objective optimization problem using linear scalarization, what means get a single objective function using the weighted sum of the three mentioned objectives.

### 3.1 Model 1

In model 1, the main idea was to design a functional ILP model easily understood by the laboratory engineers (who are not familiar with the operations research models). The use of aggregated constraints and supplementary variables, for the start time and end time of the activities, allowed us to share and validate our approach with them. The mathematical model is as follows:

$$Y_{i,t} = \begin{cases} 1 & \text{if activity } i \text{ is in progress at time } t \\ 0 & \text{otherwise} \end{cases}$$

$$O_{j,i,t} = \begin{cases} 1 & \text{if technician } j \text{ is allocated to activity } i \text{ at time } t \\ 0 & \text{otherwise} \end{cases}$$

$F_i$  = end date of activity  $i$

$G_i$  = start date of activity

$Tard_i$  = Tardiness of activity  $i$

$$Div_{i,t} = \begin{cases} 1 & \text{if activity } i \text{ is preempted at time } t \\ 0 & \text{otherwise} \end{cases}$$

$$Z(\text{Min}) = \alpha * \sum_i Tard_i + \beta * \sum_i \sum_t Div_{i,t} + \gamma * \sum_i \sum_t [t * Y_{i,t}] \quad \alpha, \beta, \gamma \text{ are the weight of each objective (1.1)}$$

$$\sum_i O_{j,i,t} \leq DO_{j,t} \quad \forall j, \forall t \quad (1.2)$$

$$F_l + 1 \leq G_i \quad \forall t, \forall i, l \in E \quad (1.9)$$

$$\sum_i (Y_{i,t} * Br_{i,k}) \leq DR_{k,t} \quad \forall t, \forall k \quad (1.3)$$

$$F_i \leq dl_i \quad \forall i \quad (1.10)$$

$$Y_{i,t} * Bc_{i,c} \leq \sum_j (O_{j,i,t} * O_{j,c}) \quad \forall i, \forall t, \forall c \quad (1.4)$$

$$G_i \geq Dmin_i \quad \forall i \quad (1.11)$$

$$\sum_j O_{j,i,t} \geq Y_{i,t} * Nt_i \quad \forall t, \forall i \quad (1.5)$$

$$Div_{i,t} \geq Y_{i,t} - Y_{i,t-1} \quad \forall i, \forall t \geq 2 \quad (1.12)$$

$$\sum_t Y_{i,t} \geq D_i \quad \forall i \quad (1.6)$$

$$Div_{i,1} = Y_{i,1} \quad \forall i \quad (1.13)$$

$$F_i \geq Y_{i,t} * t \quad \forall i, \forall t \quad (1.7)$$

$$\sum_t Div_{i,t} \leq 1 \quad \forall i \in NP \quad (1.14)$$

$$G_i \leq Y_{i,t} * t + (1 - Y_{i,t}) * M \quad \forall i, \forall t \quad (1.8)$$

$$Tard_i \geq F_i - dd_i \quad (1.15)$$

The objective in (1.1) represents the minimization of the tardiness and the penalties for preemption and also ensures the scheduling of units of duration of each activity as soon as possible. Equations (1.2) and (1.3) ensure that operator and resource capacities are respected. The constraints given in (1.4), (1.5) and (1.6) ensure the respect of skill requirements, minimal number of technicians and durations of activities, respectively. The end and start dates of activities are calculated using equations (1.7) and (1.8) ( $M$  is a “sufficiently-large” constant). Precedence constraints are given in (1.9). Inequalities (1.10) and (1.11) are the constraints for deadlines and release dates. Equations (1.12) and (1.13) determine whether an activity is either preempted or not. Inequality (1.14) is the non-preemption constraint. Finally, inequality (1.15) calculates the tardiness.

### 3.2 Model 2

Further research showed that using aggregated constraints, “big  $M$ ” constraints and supplementary variables may have a negative impact in the practical performance of the model. That is why in model 2, we opt for disaggregated formulation of precedence, deadline and release date constraints and eliminate the “big  $M$ ” constraints and variables for the start and end time of activities.

For model 2, we keep the same objective function (1.1), resources and technicians' allocation constraints (1.2), (1.3), (1.4) and (1.5), duration constraints (1.6) and preemption constraints (1.12), (1.13) and (1.14). Constraints (1.7) and (1.8) are deleted since we do not need to calculate the start and end times. Precedence (1.9), deadline (1.10) and release date (1.11) constraints are replaced by their disaggregated versions (2.9), (2.10) and (2.11), respectively. Finally, the tardiness is now calculated using (2.15). In theory, model 2 should obtain better performance than model 1, but in practice we cannot be sure until we test it.

$$D_l * (1 - Y_{i,t}) \geq \sum_{t'=t}^T Y_{l,t'} \quad \forall t \forall i, l \in E \quad (2.9) \quad \sum_{t=1}^{t=Dmin_i-1} Y_{i,t} \leq 0 \quad \forall i \quad (2.11)$$

$$\sum_{t=d_i+1}^T Y_{i,t} \leq 0 \quad \forall i \quad (2.10) \quad Tard_i \geq (t * Y_{i,t} - dd_i) \quad \forall i, \forall t \quad (2.15)$$

#### 4 TEST INSTANCE GENERATION

As Herroelen et al. [7] explain, generating instances with a specific complexity is a very difficult task, which heavily depends on the possibility to isolate the factors that precisely determine the problem hardness. In our practical problem, we can identify the number and the total duration of the activities to be scheduled as two important factors for the hardness (hardness increases when the number and the total duration of the activities increase). Consequently, in order to have a sufficient difficulty to be able to deal with real-life instances, we must ensure that generated instances have a number and total duration of the activities greater than the value found in a typical weekly operation of the laboratory.

In order to set a minimum limit for the number and the total duration of activities, the unilateral tolerance interval theory is used. The tolerance interval provides limits within which at least a certain proportion of the population falls to a given level of confidence [14]. We decided to ensure with 90% confidence that at least 90% of the weeks to be scheduled will have a total number and duration of activities below the calculated unilateral tolerance limits. So as to respect these conditions, according to the historical data, our instance must have at least 28 activities with a total duration of 387 hours.

A list of 37 most representative activities was established using the analysis historical planning and several meetings with the research engineers. Each of these activities has been characterized by its duration, resource needs, competence requirements and precedence constraints. Thanks to the experience and expertise of the research engineers, an empirical occurrence probability was assigned to each activity. A range of values was assigned to each activity according to their empirical occurrence probability.

To generate the instances, a random number is generated and the activity whose assigned interval encloses the random number is chosen to be included in the instance to be tested. The activity is cleared from the main list and the intervals are recalculated. Then new random number is generated. This process is repeated until getting an instance that meets the minimum size previously calculated (at least 28 activities and 387 hours of total duration).

#### 5 COMPUTATIONAL RESULTS

In this section, we present the results after testing the two models we proposed for the MSPRCSP. We also tried testing a third model, an adaptation of the model proposed by Afshar-Nadjafi [5] to the preemptive RCPSP with penalty (this was the only ILP model easily adaptable to our problem found in the literature); however, this model seemed to be too heavy and we were not able to find any feasible solution after several hours of computation for none of the instances. We are mainly interested in the computation time necessary to find the optimal value of the objective function and the value of the objective function when the computation time is limited.

The correct choice of values for the weights of the objective functions  $(\alpha, \beta, \gamma)$  is undoubtedly a complex process requiring further study. However, for these first tests, we decide to work with  $\alpha = \beta = \gamma = 1$ .

The test conditions were as follows:

- 4 realistic instances were generated according to the conditions of the procedure proposed in the previous paragraph.
- The original planning horizon is 108 units of time (hours). However, to ensure that a feasible solution is found, it is necessary to broaden the horizon to 156 hours. During these additional periods, it is assumed that the capacity of the resources and technicians is infinite. It is assumed that all units of activities duration planned after period 108 will not be realized.
- To prevent incomplete planning of non-preemptive activities, period 109 was set with zero availability of all resources and technicians.
- All calculations were made using the "IBM ILOG CPLEX Optimization Studio 12.7" software on a laptop with an Intel® i5-5257U processor at 2.70 GHz and 4Gb of RAM.

Initially, we are interested in the capacity of the models to find the optimal solution of the different instances. The computation time limit was set at one hour (3,600 s). Table 1 shows the results.

**Table 1. Testing models with computation time limited to 1 hour**

|                      | Instance 1 |         | Instance 2 |          | Instance 3 |          | Instance 4 |         |
|----------------------|------------|---------|------------|----------|------------|----------|------------|---------|
|                      | Model 1    | Model 2 | Model 1    | Model 2  | Model 1    | Model 2  | Model 1    | Model 2 |
| Obj. Funct. Value    | 43405      | 43405   | 24589      | 24584    | 32916      | 32915    | 35923      | 35923   |
| Computation time (s) | 2229.265   | 976.375 | 3616.391   | 3603.875 | 3600.844   | 3600.312 | 206.796    | 245.781 |

Model 2 seems to be the most efficient, giving the optimal value at least twice faster than model 1 for instance 1. Additionally, model 2 gives better values for the instances where computation was stopped by the limit time (instances 2 and 3). Even though model 1 is not the fastest to find the optimal solution for instance 4, its computation time is not far from the time used by model 1 (a 39-second difference).

As mentioned above, it is important that the models are able to give a good (not necessarily optimal) response in a reduced computation time. To test this capability, we run the models for the different instances, limiting the computation time to 2 minutes (120 s).

The results in Table 2 shows that for very small computation time, there are no big differences in the performance of the two models. Model 1 gives better results for instances 1 and 3, while model 2 gives better results for instances 2 and 4. Overall the proposed models provide solutions with a small optimality gap (percentage of deviation between the feasible found solution and the optimal solution), less than 4%, on all the realistic size instances.

**Table 2. Testing models with computation time limited to 2 minutes**

|                   | Instance 1 |         | Instance 2 |         | Instance 3 |         | Instance 4 |         |
|-------------------|------------|---------|------------|---------|------------|---------|------------|---------|
|                   | Model 1    | Model 2 | Model 1    | Model 2 | Model 1    | Model 2 | Model 1    | Model 2 |
| Obj. Funct. Value | 43476      | 43468   | 24715      | 24949   | 33654      | 33396   | 35925      | 35973   |
| GAP               | 0.820%     | 0.482%  | 2.236%     | 2.464%  | 3.502%     | 2.208%  | 0.080%     | 0.298%  |

In general, model 2 shows better performance than model 1. This better performance may be due to the fact that in model 1 we use a “big M” in constraints (1.8). Using “big M” in ILP can generate LP-relaxation problem leading to loose node bounds, so more nodes need to be examined to find a solution, and computation time increases. Another possible reason for the lower performance of model 1 is the use of aggregated constraints. The reason why the model derived from Afshar-Nadjafi [5] did not work may be that their approach needed about 100 times more binary variables than the models we proposed to solve the instances we tested. In fact, a greater number of binary and/or integer variables implies a larger size (with exponential growth) of the search tree.

In practice, the preparation of weekly scheduling of laboratory activities is done manually. This is a very complex process and it needs about two hours to obtain a feasible schedule that is not always the most performant. The proposed models give engineers the ability to have a more performant schedule in a matter of minutes. In addition, but not least, the use of the models eliminates the risk of forgetting constraints, which can happen easily in the manual scheduling process. Reducing the time required to obtain a good scheduling gives research engineers the opportunity to spend more time on research activities and also allows them to have a greater reactivity against unforeseen events that may occur during the week.

## 6 CONCLUSIONS

The literature about scheduling research activities is very sparse. The objective of this paper was to show how operations research techniques may be applied to scheduling activities in a research laboratory. To work with a relatively short scheduling horizon allows us to manage the inherent variability of research activities and hence to treat the problem as a traditional scheduling problem. Using operations research models can reduce the time spent by researchers in the weekly planning of activities, giving them more time to devote to research and also a greater reactivity to the unexpected.

Although the classic version of the RCPSP is very powerful, it is not able to represent all the specific features we can find in real life, forcing researchers to develop a lot of new variants of the problem. However, none of the variants proposed in the literature seem to be able to model correctly the scheduling problem of the studied laboratory. To fill this gap, we proposed a new variant of RCPSP: a Multi-Skill RCPSP with penalty for preemption, and its respective ILP models for exact solving.

We proposed two models: model 1 using aggregated constraint for the precedence relationships and model 2 where these constraints have been disaggregated. Computational results show that model 2 can give better performance than model 1. This better performance of model 2 may be explained by the fact that model 2 do not use “big M” in constraints and the use of disaggregated constraints. Computational results also show that the models we proposed are more efficient than the approach proposed by Afshar-Nadjafi [5] for the RCPSP with penalty for preemption. Their approach need about 100 times more binary variables to solve our instances.

As future work, we must study the ways of improvements of model 2 in terms of the quality of the linear relaxation. We could, for example, continue to disaggregate some of the constraints present in the models, in order to have a better linear relaxation. The practical performance of the resulting models should be compared to the performance of the initial models, since better linear relaxation does not guarantee better practical performance for solving the problem [13].

## 7 REFERENCES

- [1] Mancel, C. 2004. Modélisation et résolution de problèmes d’optimisation combinatoire issus d’applications spatiales, INSA, Toulouse.



- [2] **Damay, J.; Quilliot, A. and Sanlaville, E. 2007.** Linear programming based algorithms for preemptive and non-preemptive RCPSP, *European Journal of Operational Research*, vol. 182, no. 3, pp. 1012-1022.
- [3] **Hartmann, S. and Briskorn, D. 2010.** A survey of variants and extensions of the resource-constrained project scheduling problem, *European Journal of Operational Research*, vol. 207, no. 1, pp. 1-14.
- [4] **Orji, I. M. J. and Wei, S. 2013.** Project Scheduling Under Resource Constraints: A Recent Survey, *International Journal of Engineering Research and Technology*, vol. Vol.2-Issue 2, pp. 1-20.
- [5] **Afshar-Nadjafi, B. 2014.** Resource Constrained Project Scheduling Subject to Due Dates: Preemption Permitted with Penalty, *Advances in Operations Research*, vol. 2014, pp.1-10.
- [6] **Javanmard, S.; Afshar-Nadjafi, B. and Akhavan Niaki, S. 2017.** Preemptive multi-skilled resource investment project scheduling problem: Mathematical modelling and solution approaches, *Computers & Chemical Engineering*, vol. 96, pp. 55-68.
- [7] **Herroelen,W.; De Reyck, B. and Demeulemeester, E. 1998.** Resource-constrained project scheduling: A survey of recent developments, *Computers & Operations Research*, vol. 25, no. 4, pp. 279-302.
- [8] **Ballestín, F.; Valls, V. and Quintanilla, S. 2008.** Pre-emption in resource-constrained project scheduling, *European Journal of Operational Research*, vol. 189, no. 3, pp. 1136-1152.
- [9] **Vanhoucke, M. 2008.** Setup times and fast tracking in resource-constrained project scheduling, *Computers & Industrial Engineering*, vol. 54, no. 4, pp. 1062-1070.
- [10] **Correia, I.; Lourenço, L. L. and Saldanha-da-Gama, F. 2008.** Multi-Skill Resource-Constrained Project Scheduling: Formulation and Inequalities, *CIO-Working Papers*, no. 17, pp. 1-25.
- [11] **Artigues, C.; Michelon, P. and Reusser, S. 2003.** Insertion techniques for static and dynamic resource-constrained project scheduling, *European Journal of Operational Research*, vol. 149, no. 2, pp. 249-267.
- [12] **Koné, O.; Artigues, C.; Lopez, P. and Mongeau, M. 2011.** Event-based MILP models for resource-constrained project scheduling problems, *Computers & Operations Research*, vol. 38, no. 1, pp. 3-13.
- [13] **Artigues, C. 2017.** On the strength of time-indexed formulations for the resource-constrained project scheduling problem, *Operations Research Letters*, vol. 45, no. 2, pp. 154-159.
- [14] **Janiga, I. and Garaj, I. 2005.** On one-sided tolerance intervals of normal distribution with unknown parameters, *Journal of the Applied Mathematics, Statistics and Informatics (JAMSI)*, no. 1, pp. 87-100.