A bi-objective algorithm for a reactive multi-skill project scheduling problem

Cheikh Dhib, Ameur Soukhal, Emmanuel Néron, Hafedh Mohamed-Babou,Bedine Kerim
Université des Sciences de Technologie et de Medecine, Route NDB, Nouakchott
dhib.cheikh@gmail.com
Université François Rabelais, tours- france
ameur.soukhal@univ-tours.fr
Université François Rabelais, tours- france
emmanuel.neron@univ-tours.fr
Ecole de Mines de Mauritanie
hafedh.mohamed-babou@emim.mr
Albaha University, college of computer science and IT, Department of computer science
bkerim@bu.edu.sa, bedkarim76@yahoo.com

ABSTRACT

The aim of this paper is to present project scheduling problem met in a an industrial context. The focus is mainly to the reactive model. In fact, the predictive case was studied in previous works, and this paper presents a solution for a reactive version of the model studied before. We proposed a linear mathematical model for the problem and then we show that this model cannot be used in practice to the solve problem. Then we present a bi-objectve genetic algorithm proposed to solve this problem. Experiment results are provided also.

Indexing terms/Keywords

Scheduling, project scheduling, resource staffing, linear programming, mata-huristics, genetic algorithms

Academic Discipline And Sub-Disciplines

Computer scinces, Applied mathematiques

SUBJECT CLASSIFICATION

E.g., Mathematics Subject Classification; Library of Congress Classification

TYPE (METHOD/APPROACH)

Mathematical programming, Exact methods, meta-huristics

INTRODUCTION

Project scheduling problems are among the most studied scheduling problems in the literature. Resource Constrained Project Scheduling Problem is the most classical version of these problems. In the RCPSP, a set of non-preemptable activities have to be processed. Activities require a given amount of each resource to be processed. Resources are available in limited amount. Activities are submitted to classical end-to-start precedence relationship. This problem is known to be NP-Hard (Blazewicz, Lenstra, & Rinnooy, 1983)and several states of the art can be found dealing with RCPSP (Demeulemeester & Herroelen, 1997).

Resource modeling has been one fruitful research direction for new project scheduling models. Resources can be renewable, non renewable or doubly constrained. Moreover resource requirements of activities may differ from one mode to another. These types of resources are modelled in the Multi-Mode Resource Constrained Project Scheduling problem (Sprecher & Drexl, 1998) (Hartmann, 1998). Several methods for solving MM-RCPSP have been proposed including exact methods such as branch-and-bound and heuristics.

Recently, authors have proposed to enlarge RCPSP model to take into account the notion of skills, i.e, staff members involved in the project realization can contribute only to a given subset of activities. This is known as the Multi-Skill Project Scheduling Problem (Bellenguez-Morineau & Néron, 2007) (Bellenguez-Morineau & Néron, 2004).

Resources considered are human resources, i.e. staff members, each of them are able to perform more than one kind of activities. This model is useful in several industrial contexts, and for instance in the context of IT companies where human resources are the most constrained resources. Moreover, availability periods are considered for staff members. This extension can be seen as a special case of Multi-mode RCPSP, but having potentially a huge number of modes per activity.

Recently several studies focus on of skill constraints, in a context of project scheduling (Walter & Zimmermann, 2010).

The industrial context underlying this paper is slightly different than the one addressed before. This work is a joint work with a company contributing on an open-source ERP platform. According to their need and some specificities encountered in the context of project scheduling embedded within a generic framework, we have defined a model for project scheduling.

Main differences between the model addressed in this paper and the MultiSkill Project Scheduling Problem are :

- **Activities can be either preemptive or non preemptive**. This is a key point of the model. On most of real-life projects, some activities can be interrupted without any penalties, (such as writing documentation, archiving etc), whereas some activities cannot be interrupted.
- In the case of preemption,**the resource assigned to one skill for one activity must be the one that completes the activity after the preemption**. This constraint is used to limit the preemption effect on the project organization, thus we do not consider non-resumable or semi-resumable activities, that would be implies by the fact that a person has to redo a part of the activity.
- **Exactly one resource is required for each skill for one activity**. This is mainly due to management of the project. Moreover in the case where several resources can contribute to the same skill, the problem of the workload estimation becomes hard. Notice that this constraint is relevant in the context of small and medium size projects.
- **All part of one activity, each corresponding to one skill required for this activity, must start simultaneously, but can be preempted and restarted at different time-points**. This is what we call synchronization constraint that corresponds to a short period needed for briefing all the persons contributing on this activity
- **Each resource, i.e., person, have one preferred skill**. This is not really a hard constraint but project manger attempt to satisfy these preferences.

We studied the case of reactive problem of *PMSPSP*. From a practical standpoint, this case is particularly interesting because the data available at the completion of the first schedule are forced to change. Several sources of interference exist, namely the underestimation of duaration by the project manager often leads to a reevaluation of the duration a resource increase to a task. Other duties may also be introduced into a project running. The absence of people can also happen in an unpredictable manner, etc. Conversely, as the hazards of over-estimation of the load, deleting a task, the presence of a resource that was not foreseen at the start of the project, etc., may also occur. We refer to (Billaut, Moukrim, & Sanlaville, 2010) for a complete description of possible hazards. Faced with this multiplicity of sources of uncertainty and in the absence of statistical data on the arrival of each type of uncertainty, it will not be possible to develop robust methods capable to absorbing these aleas.

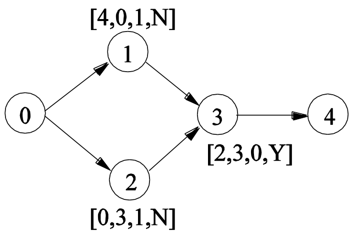
So our approach, is to develop a purely reactive method without relying on pro-active based solutions.

The model proposed here has the advantage of being generic to cope with several types of disruptions. The main idea is to add a second criterion to minimize schedule disruption occurred during project excution. In the first section, we formally define the model, and then we describe the instances used in experimental results. In the third part, a linear model is proposed. This linear model is used in the *epsilon-constraint* method proposed in Section 5 to compute the Pareto optimal fronts. Then we propose an evolutionary algorithm of type NSGA-II to solve this problem. Before conculusion, we present experimental results conducted on the generated instances.

1. PROBLEM DESCRIPTION

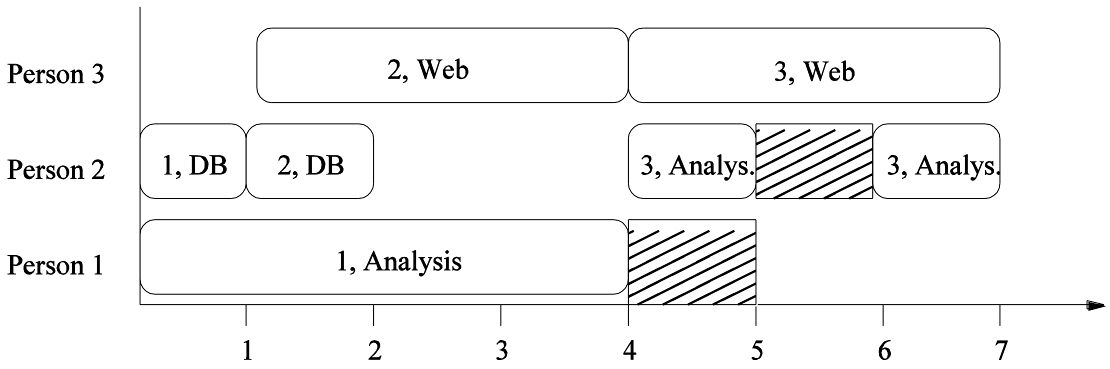
We consider a multi-skill project scheduling problem as in Figure 1. We suppose that the project is in progress at the time point τ. At this time point, we decide to re-schedule this project due to some perturbations (new tasks arrived, etc) that arrive after the project has been started. So we have *n* tasks to be scheduled. Among them, a sub-set is already in progress. For this sub-set, we distinguish *2* types: in progress preemptive tasks *IPP* and in progress non preemptive tasks *IPNP.* Other tasks are either have been planned but not yet started or new tasks that have just to be identified at the instant τ.

Figure 1: Exemple



	Skill			
Person	Analysis	Web	DB	Unavailability
1	1	1	1	[4,5[
2	1	0	1	[5,6[
3	1	1	0	

Figure 2: Feasible solution



In addition to constraints of the predictive problem for which we looked for a schedule minimizing the project total duration C_{max}, we add the following constraints :

- every non-preemptive task which is in progress atτ has to continue its execution until the end without interruption and without resource assignment change. Durations may change and in this case that modification will be applied from the instant τ
- For each skill of in progress preemptive task, the resource assignment does not change, but the time execution windows may change in the new schedule.

The goal is to find a new planning which respect all constraints (original and new constraints) in such a way that the new $C_{max}$is minimum and the maximum number of resource re-assignment is also minimal.

The change of resource assignment criterion is important from an organizational point of view.

because in projects such as IT projects, a person assigned to a task can be linked with a customer or another party from outside, so changing that person too frequently is not preferable and can contribute to deterioration of service quality. Another argument justifies this criterion: if a person is assigned to a task, it often happens that the person is prepared to be effective once the job is started. For example, if performing a given tasks needs to do some reflexion or thinking to find some ideas that are not trivial, the person sometimes develop ideas during his free time before starting this task. So changing it can affect the effectiveness of this task realization

We ignore here the prospective task sequencing modification which can arise for some resources. This seems to be less important, because even the proposed planning does not change the tasks order of execution, persons can do that by them selves when it is possible.

Hence, this model is a particular case of the predictive model studied in (Dhib, C., Soukhal, A., & Néron, E, 2011). The integration of a second criterion leads to a more complicated problem. Therefore, proposed methods for solving the predictive problem cannot be anymore used with this new problem.

2. LITERATURE REVIEW

In this section, we want to highlight the most important research axes studied by project scheduling researchers when the project information or data are considered uncertain. When it is a case, scheduling approaches anticipating the data change are considered (proactive scheduling). When a re-scheduling is necessary to modify in progress planning , due to arrived distributions , hence, we speak about reactive methods.

It is to note that, in reality, an established schedule at the start of project still very rarely without change until the end of project. In scheduling literature, we distinguish three types of schedules (Billaut, Moukrim, & Sanlaville, 2010):

1. *Predictive* schedule: in this case, an initial schedule based on provided deterministic data is established without anticipating eventual changes. Many resolutions methods was studied in the literature (Demeulemeester, E., & Herroelen, W, 1997).
2. *pro-active*schedule: it is a predictive schedule, which is realized with taking into consideration that initial data may change (perturbed) during project execution. For example, a task duration may be prolonged, a resource may become unavailable.

   In (Van de Vonder, Demeulemeester, & Herroelen, 2007), authors developed heuristic methods to find a stable solutions of the *RCPSP* problem, i;e, computed schedules are able to absorb arrived changes without perturbing the actual schedule. In this study, one type of perturbations is considered. It is, the change of tasks duration. For each task, authors define a weight which reflect the degree of importance or not of a change on a task start date in the computed schedule with respect to its start date in the actual schedule. The authors were, so, interested by minimizing the wighted sum of deviations of start dates of tasks.

First of all, they compute an optimal solution by using an exact algorithm such that these proposed in (Demeulemeester & Herroelen, A branch-and-bound procedure for the multiple resource-constrained project scheduling problem, 1992), (Demeulemeester & Herroelen, 1997). Then, they fix a deadline date for project completion time (C_{max}), which is equal to $\alpha \times C_{max}^{opt}$ with α= 1.3. Then, they create buffers after critical tasks. These buffers have to extend a critical task, without need to right shift its successors.

In (Drezet, 2005), authors have studied a real problem, in which the objective is to minimize the maximum lateness L_{max}. They point out that, from an industrial point of view, good quality solutions are not interesting except if they are not called into question, from the moment that a person planned work for a given task take more time than it was firstly planned. The authors are interested to a proactive approach, where the objective is to find robust solutions, with respect to some robustness indicators based on both person unavailability and task processing time extension probabilities. Priority rule based heuristics was studied.

In (Chtourou & Haouari, 2008), authors propose a proactive method for the classical *RCPSP* problem, with uncertain tasks processing time. The idea is to add another criterion to be optimized. This criterion is a measure of solutions robustness. Many measures based on slack was tested. The proposed approach can be summarized in two phases: In the first phase, a priority rules based heuristic algorithm is executed many times to optimize the C_{max} and the minimal found value is used in the second phase. In the second phase, we do the same thing but by considering the second criterion and keeping the C_{max} value equals or less than its value in the first phase. This study shows that introducing these measures for finding robust solutions is beneficial, and even, shows that some robust solutions may have best C_{max} than other solutions which are less robust.

There exists, other methods which are not dealing with perturbations anticipation. In contrast, they propose to manage better these perturbations when they arrive. For example, in (Artigues & Roubellat, 2000), authors propose a polynomial algorithm which insert a task into an instance of *RCPSP* which is in progress.

3. *reactive* scheduling: called also on-line or dynamical scheduling, it is to recompute the schedule after some perturbations arrive on the initial date based on which the actual schedule was calculated. Many approaches were studied in the literature. We distinguish manly:

    a. rebuilding a new schedule: this strategy consist of executing the same method used for finding the base schedule, for the remaining of project, without taking into account the actual planning;

    b. actual solution repair: in this strategy, we try to repair the existing solution by inserting the arrived perturbations;

    c. building a new solution based on the actual one, by adding one or more criteria to optimize, for ensuring the stability with respect to the actual planning.

In the best of our knowledge, the literature on reactive scheduling, for project management and scheduling with multi-skilled workforce is almost empty. We find only the work of (Drezet, 2005), where authors propose a reactive method which is limited to a very small horizon. In fact, they consider only perturbations, which have an impact on only one day (divided to 6 periods). The objective is to minimize the number of violated constraints, then minimizing the maximum number of of assignment change per person. A Tabou search method optimizing lexicographically, the two criteria is proposed in this work.

3. Instances generation

Reactive instances are generated from predictive ones by introducing some perturbations on this instances.

We apply these perturbations from a given instant τ. So, we developed an instance generator, which, from a given instance of the initial problem, generate a reactive problem instance by solving the predictive problem using a heuristic algorithm (Dhib, C., Soukhal, A., & Néron, E, 2011) Considered perturbations are : modification on tasks skill processing time, persons availability. We used data sets of *10, 16 and 20* tasks.

The following procedure describes the process of reactive instances generation:

- Actual planning: computed using the heuristic (Dhib, C., Soukhal, A., & Néron, E, 2011) with the *EDD* priority rule, we denote the project total duration by C^0_{max}
- Re-scheduling dates : $\tau \in \left\{0, \left\lfloor\frac{C^0_{max}}{5}\right\rfloor, 2 \times \left\lfloor\frac{C^0_{max}}{5}\right\rfloor, \dots, C^0_{max} - 1\right\}$
- Considered perturbations : Skill/task duration increase, availability or unavailability of a person.
- for all tasks, with completion date between $\tau + 1$ and$\tau + \left\lfloor\frac{C^0_{max}}{5}\right\rfloor$, they have a probability of 20% to increase their processing time of $1 \leq \delta \leq \max(1,0.3 \times p_{i,k})$
- for each person *m*and a period $t \in [\tau, \tau + \left\lfloor\frac{C^0_{max}}{5}\right\rfloor - 1]$, if person was available, he/she has a probability of 10%to become unavailable. Else, he/she has a probability of 5% to become unavailable.

- A solution feasibility verification is done, for non preemptive tasks which are in progress on the time interval $[\tau, \tau+1]$. This verification may redo some generated perturbations during the previous steps.

4. Mixed Integer Program (MIP)

In this section , we propose mixed integer variables linear program to solve this problem of re-scheduling from an instant τ. The proposed MIP is based on the model proposed for the predictive model (Dhib, C., Soukhal, A., & Néron, E, 2011). Because of the huge number of variables and constraints generated by this model, it cannot be used to solve a big size instances within a reasonable execution time. Despite of that, we used this model in an exact method based on an $\varepsilon - constraint$ approach which is described in next sections in order to compute a Pareto front for some instances with small sizes.

Additional data : in addition to initial instance data, we define here a reactive instance related data:

- $y_{i,k,m}$=1 if person P_m was assigned to a skill $S_k$of task A_i in the initial planning ; 0 else
- $z_{i,k}$ = 1 : if the skill S_k of the task A_i is in progress , 0 else
- s_i : start date of task A_i in the actual planning
- τ : rescheduling start date
- $x_{i,m,k,t} = 1$ if the skill S_k of task A_i was executed by person P_m during [t, t+1]; 0 else

Auxiliary Variables

- $\overline{y}_{i,k,m} = 1$if person P_m is assigned to skill $S_k$of the task $A_i$in the new planning; 0 else
- $g_{i,k,m} = 1$ if$y_{i,k,m} \neq \overline{y}_{i,k,m}$; o else. The boolean indicates if the assignment of the skillskill $S_k$of the task A_i to person $P_m$the new planning is different of the one in the base planning.
- $g_m = \sum_i \sum_k g_{i,k,m}$This variable count the number of assignment of skills of tasks to personP_m , which did not appear in the reference planning(planning initial).

Constraints

1. fixing the planning before τ :

   $\overline{x}_{i,m,k,t} = x_{i,m,k,t}\ \forall i,k,m, 0 \leq t \leq \tau - 1$**(1)**

2. do not interrupt non preemptive task which is in progress during $[\tau, \tau + 1]$

   $\overline{x}_{i,m,k,t} = y_{i,k,m}\ \forall\ (i,k) \in I_p\ \forall\ i \in A_{\overline{p}}\ \forall\ m\ s_i \leq t \leq s_i + p_{i,k}$**(2)**

   $\overline{y}_{i,k,m} = y_{i,k,m}\ \forall i,k,m\ \ si < \tau$**(3)**

3. $\overline{y}_{i,k,m} = 1$ if theperson$P_m$executes the skill $S_k$of the task A_i

   $\overline{y}_{i,k,m} = \frac{\sum_t \overline{x}_{i,m,k,t}}{p_{i,k}} \forall i,k,m\ \ (p_{i,k} > 0)$**(4)**

4. assignment change with respect of initial schedule (planning)

   $g_{i,k,m} \geq \overline{y}_{i,k,m} - y_{i,k,m}$**(5)**

   $g_{i,k,m} \geq y_{i,k,m} - \overline{y}_{i,k,m}$**(6)**

   $g_{i,k,m} \leq 2 - (y_{i,k,m} + \overline{y}_{i,k,m})$**(7)**

   $g_{i,k,m} \leq y_{i,k,m} + \overline{y}_{i,k,m}$**(8)**

5. Total person assignment change

   $g_m = \sum_i \sum_k g_{i,k,m}$**(9)**

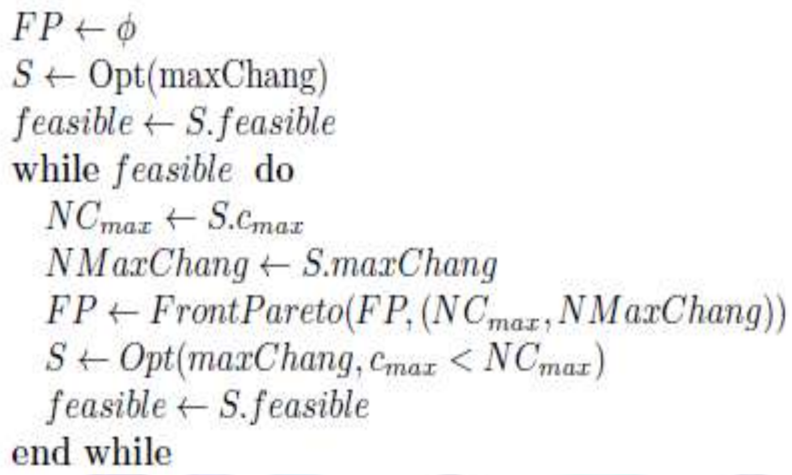
Objectives : we minimize at the same time, both project completion date C_{max} (Equation 10) and the maximum number of assignment change (Equation 12).

$minC_{max}$**(10)**

$\min max_m\, g_m$**(11)**

5. Pareto optimal solutions

From the linear model, a Pareto front is obtained using the Ɛ-constraint approach described by the algorithm **3.3**. We firstly, solve the linear model considering only the second criterion en ($\max g_m$). If a feasible solution is found with a completion project dateC^0_{max}, we solve again the model with an additional constraint$C_{max} \leq C^0_{max}$ -1. This procedure is repeated until the problem becomes unfeasible. This approach allowed to enumerate the Pareto optimal solutions for small size instances. We used these result, namely for measuring the performance of an NSGA-II algorithm proposed in 3.4.1. Experimental results of this method are presented in Section 6.



```
$FP \leftarrow \phi$
$S \leftarrow \mathrm{Opt(maxChang)}$
$feasible \leftarrow S.feasible$
while $feasible$ do
    $NC_{max} \leftarrow S.c_{max}$
    $NMaxChang \leftarrow S.maxChang$
    $FP \leftarrow FrontPareto(FP,(NC_{max},NMaxChang))$
    $S \leftarrow Opt(maxChang, c_{max} < NC_{max})$
    $feasible \leftarrow S.feasible$
end while
```

Algorithm 1 : Espsilon-constraint method

Heuristic methods

The just presented exact method do not allow to solve instances of real size. Consequently, we propose a bi-objective genetic algorithm of type NSGA-IIin order to compute an approximation of the Pareto front.

NSGA-II Algorithm

An implementation of the NSGA-IIalgorithm (Deb, Agrawal, Pratap, & Meyariva, 2000) is proposed to compute an approached Pareto front . NSGA-II algorithm is widely used for solving multi-criteria optimization problems. recently, (Ballestın & Blanco, 2011) realized a study on the multi-objective RCPSP, in which they compared the performance of NSGA-II with respect to other multi-objective meta-heuristics *SPEA2* and *PSA*. Their study gives a large advantage to NSGA-II compared to *PSA* and slightly more advantageous than *SPEA2* method. As a population based evolutionary algorithm, NSGA-II algorithm starts from a set of solutions (initial population) and iteratively, enhance them until some stop condition is reached (number of iterations or execution time limit, etc.). The NSGA-II algorithm principle can be defined as follows suit :

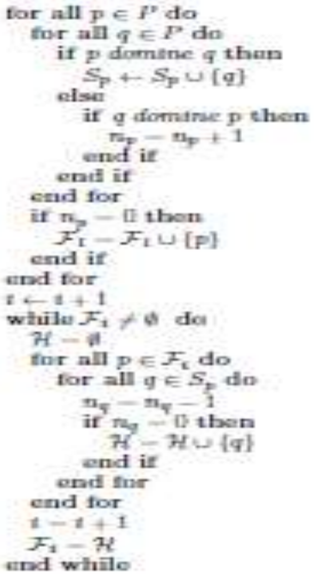
1. generate an initial population Q_0 with N individual,
2. reproduce a population Q_1 of size N from Q_0using reproduction operators(crossover and mutation),
3. merge the two populations into one $Q^{'} = Q_0 \cup Q_1$
4. select N individual of $Q^{'}$using a selection strategy and put them into Q_0,
5. repeat steps (2-4) until the stop condition becomes true.

To produce a novel generation, NSGA-II method uses an elitist strategy. It regroups the 2*N individuals which results from the genetic operations into fronts F_0, ...,F_n such that any solution in a front F_0does not dominate the other. Furthermore, solutions in front $F_{i+1}$are dominated by these of front F_i.

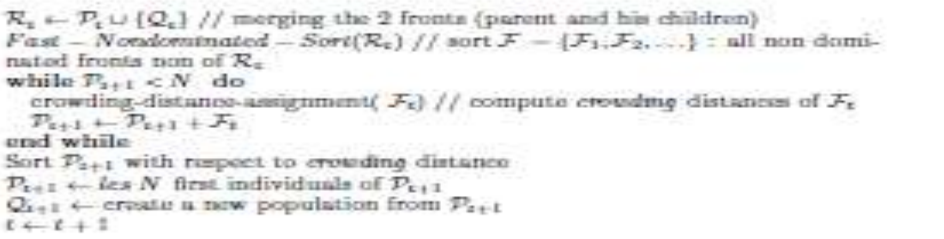
Individuals to be selected are taken from fronts F_0, ...,F_k. The index k corresponds to k^{ith} front such that the total number of individuals of fronts F_0, ...,F_{k-1} is less than N and the total number of individuals of fronts F_0, ...,$F_k$is greater than or equals to N. In such a way, All individuals belonging to F_0, ...,$F_{k-1}$are selected.

To complete the population size to N, other solutions in front F_k have to be selected. NSGA-II method uses naturally a dispersion mechanism. This mechanism is applied during the selection from the last front F_k. The selection is done in the decreasing order with respect to the Crowding distance. This distance is large for a solution if the density of solutions around it are weak, and is small if this density is strong.

Algorithms for Pareto fronts computation



```
for all $p \in P$ do
  for all $q \in P$ do
    if $p$ domine $q$ then
      $S_p \leftarrow S_p \cup \{q\}$
    else
      if $q$ domine p then
        $n_p \leftarrow n_p + 1$
      end if
    end if
  end for
  if $n_p = 0$ then
    $\mathcal{F}_1 \leftarrow \mathcal{F}_1 \cup \{p\}$
  end if
end for
$i \leftarrow i + 1$
while $\mathcal{F}_i \neq \emptyset$  do
  $\mathcal{H} \leftarrow \emptyset$
  for all $p \in \mathcal{F}_i$ do
    for all $q \in S_p$ do
      $n_q \leftarrow n_q - 1$
      if $n_q = 0$ then
        $\mathcal{H} \leftarrow \mathcal{H} \cup \{q\}$
      end if
    end for
  end for
  $i \leftarrow i + 1$
  $\mathcal{F}_i \leftarrow \mathcal{H}$
end while
```



```
$\mathcal{R}_t \leftarrow \mathcal{P}_t \cup \{Q_t\}$ // merging the 2 fronts (parent and his children)
$Fast - Nondominated - Sort(\mathcal{R}_t)$ // sort $\mathcal{F} = \{\mathcal{F}_1, \mathcal{F}_2, \ldots\}$ : all non domi-
nated fronts non of $\mathcal{R}_t$
while $\mathcal{P}_{t+1} < N$  do
  crowding-distance-assignment( $\mathcal{F}_t$) // compute crowding distances of $\mathcal{F}_t$
  $\mathcal{P}_{t+1} \leftarrow \mathcal{P}_{t+1} + \mathcal{F}_t$
end while
Sort $\mathcal{P}_{t+1}$ with respect to crowding distance
$\mathcal{P}_{t+1} \leftarrow$ les $N$ first individuals of $\mathcal{P}_{t+1}$
$Q_{t+1} \leftarrow$ create a new population from $\mathcal{P}_{t+1}$
$t \leftarrow t + 1$
```

Algorithm 2 : NSGA-II algorithm

Initial population

Initializing a genetic algorithm is an important phase and can have an impact on the quality of computed solutions during the other phases. If, for example, initial solutions are not not dispersed as enough, the algorithm may converge quickly and in a premature manner.

To have a diverge initial population, we execute the greedy algorithm 2*N times with the 10 priority rules presented in and with the random rule (2*N -10) times. Then, we use the same selection strategy as in the NSGA-IImethod, i.e, we use the selection by rank then based on the crowding distance applied on the k^{ith} front, we complete the population to N individual.

Solution representation and genetic operators :

We use a representation with an integrated assignment (Dhib, C., Soukhal, A., & Néron, E, 2013). We represent so a solution by a priority rule list which respects precedence relationships coupled with a list of resources (persons) associated to each task for its realization. The list of persons is sorted on the same order of skills which persons will do. We use the same crossover and mutation operators as described in Section 5. In progress preemptive tasks are not concerned by assignment change. Decoding a solution is done by using the heuristic algorithm, without assignment problem resolution (Dhib, C., Soukhal, A., & Néron, E, 2011).

Parents selection for reproduction

In genetic algorithms, two selection methods are necessary (may be similar) : The selection of a new generation to replace the actual one and the selection of parents to be used in the reproduction process.

In our case, for new generation selection, we use the native NSGA-II method for selection as described above. Pour the choice of individuals that will participate to the generation to a new population, we use a selection by tournament as in mono-criteria genetic algorithms. The fitness function used here is the rank of the solution in the Pareto front in which it belongs. Solutions of first rank have so more chance to be selected. On the other hand, two solutions with the same rank, have the same chance of selection.

6. Experimental results

In the following sections, we present the numerical results of our NSGA-II method. Firstly, we compare its results with optimal solutions computed by the exact method presented in Algorithm 1. For dowing this comparison, we use the generational distance metric as described above. For instances where we do not have optimal solutions, we study the performance of the NSGA-II using *hypervolume}* metric.

Used evaluation metrics

In contrast of mono-criteria problems, the quality measure of a set of potentially Pareto optimal solutions is not a trivial task. There exists many performance measures for the multi-objectives methods (Zitzler, Thiele, Laumanns, Fonseca, & Grunert da Fon-seca, 2003)}. In (Van Veldhuizen, D., & Lamont, G., 2000)., authors classified these measures into three groups :

1. according to Pareto optimal proximity ;
2. according to solutions diversity;
3. according to the two criteria at the same time.

In our case, we use *Generational Distance* metric, which is a metric of the first category, to compare the *NSGA-II* results with Pareto optima found by the *epsilon-constraint* exact approach.

In order to measure the NSGA-II performance for other instances where we don not know the optimal front of Pareto, we use the *hypervolume* metric, which measure at the same time both the diversity of solutions and the distance from the optimum.

Generational Distance : this metric compute the distance between the approached front pf Pareto OP* and a reference front OP using the following formula :

$$GD = \frac{(\sum_1^{|OP^*|} d_i^p)^{\frac{1}{p}}}{|PO*|})$$

where *p* is the number of criteria to optimize. I the case of two criteria:

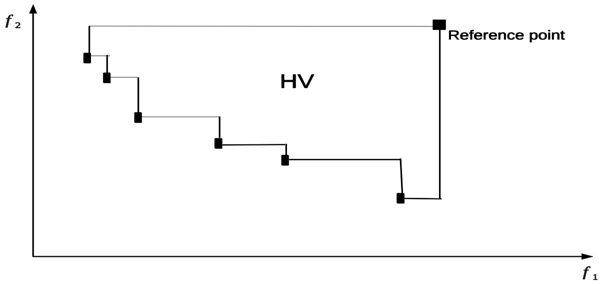
$$d_i = \min_{k=1}^{k=|PO|} \sqrt{\sum_1^{p} (f_j^i - f_j^k)^2}$$

This distance is computed by using normalized coordinates for both criteria. For each instance, the maximum value of each criterion taken from the two fronts: approached and optimal, is used.

Hypervolume : This indicator measures the volume produced by points of the front to be evaluated with respect to a reference point, which is dominated by all points of the front (see Figure 3). Because we minimize the economical functions, the quality of approached fronts is considered good when this indicator is high. In fact, in such a case, the solutions are very far from the reference point. It is clear that the choice of this reference point is very important. To this point, we must compute an upper bound for each criterion. Concerning the project completion date C_{max}, we consider the maximum planning horizon as an upper bound. For the second criterion, the maximum assignment change for a person is reached where he/she is assigned to all tasks where he/she was not assigned to in the initial schedule and for a skill different from the one he/she was assigned to for this task itself.

The upper bound for the second criteria is son the maximum among these values. We note that also the two criteria have not always the same magnitude. So, we use always the normalized values by dividing on the values of upper bounds. Hence, the reference point has the coordinates (1,1).

Figure 3: L'hypervolume engendré par un ensemble de points non dominés (bi-objective case)



NSGAII vs epsilon-constraint exact approach

In this section, we measure the quality of solutions computed by the genetic algorithm by comparing them with the Pareto optima obtained by the based *epsilon-constraint* exact method. From the original instances for the predictive case, experimentations has been conducted using 10 instances of 10 tasks. For each instance, 3 reactive instances were generated at periods $\left\{0,\ 3\times \left\lfloor\frac{C^{0}_{max}}{5}\right\rfloor, 5\ \times \left\lfloor\frac{C^{0}_{max}}{5}\right\rfloor\right\}$. NSGA-II parameters are:

- population size = **20** individuals,
- number of generations (iterations) : **250**,
- crossover rate : **0.8**,
- mutation rate : **0.05**.

Table 1: NSGAII vs epsilon-constraint

event	distance	NSGA-II time	EC time
0	0.66	1.02	429.67
3	0.39	1.02	19.9
5	0.31	0.89	5.03

NSGA-II performance using Hypervolume indicator

In Table 2 , we present experimental results obtained by *NSGA-II* on instances of 16 tasks by using generated perturbations at time periods $\left\{0,\ \left\lfloor\frac{\square^{0}_{\square\square\square}}{5}\right\rfloor,\ \ 2\times\left\lfloor\frac{\square^{0}_{\square\square\square}}{5}\right\rfloor\right\}$This gives in total 120 instances ($3\ \times 40$).

NSGA-II parameters :

- population size: **32** individuals
- number of generations: **250**
- crossover rate : **0.8**
- mutation rate : **0.05**

Table 2: NSGAII on instances of 16 tasks

event	HV	NSGA-II time
0	0.59	1.73
1	0.54	1.69
2	0.54	1.40

The results presented in Table 1, show a decrease of the distance between the approached front and the optimal one when the perturbations arrive late. For example, the distance was 0.6619 at the instant $\square = 0$, while this distance was only 0.3136 for instances corresponding to perturbations at time instant of $\square = 5\ \times \left\lfloor\frac{\square^{0}_{\square\square\square}}{5}\right\rfloor$. We note that the exact approach sometimes, finds best solutions than the initial schedule in terms of C_{max}, especially when the reactive instance corresponds to perturbations arrived very early (near to instant 0), but with cost of some modifications on initial resources assignments.

Another important remark, which can "call into question" the use of generational distance as a performance indicator, is the number of solutions. For example, in some cases NSGA-II algorithm find a Pareto front with only one point belonging to the optimal front, which gives as a result, a generational distance of zero, while the optimal front contains more than one point.

In Table 1, the second column (HV) presents the average values given by the hypervolume indicator engendre by non-dominated points computed by the NSGA-II algorithm.

We can note that this value is highest for the instances corresponding to a $\square = 0$, which is not expected, if we refer to results in Table 1. This is partly due to that the upper bound of $\square\square\square\ \square_{\square}$ is usually highest at the beginning of project because it depends to the number of tasks which are not yet completed**.**

CONCLUSION

In this paper, we studied a reactive Multi-Skill Project Scheduling problem. We proposed a global approach, which does not depend to the nature of perturbations arise on in progress planning. We propose to solve this problem by considering two criteria : the original criteria in the predictive planning which is the minimization of makespan and a second criteria which minimizes the maximum number of resource assignment change with respect to the base schedule.

To solve this problem, we proposed an exact approach based on a linear program and an approached method based on genetic algorithm of type NSGA-II

The first obtained results are satisfying. It will be important to realize more experimentations to analyze more accurately the performance of proposed methods.

REFERENCES

I. Artigues, C., & Roubellat, F. (2000). A polynomial activity insertion algorithm in a multi-resource schedule with cumulative constraints and multiple modes. *European Journal of Operational Research, 127* (2), pp. 294-316.

II. Ballestın, F., & Blanco, R. (2011, 1). Theoretical and practical fundamentals for multi-objective optimisation in resource-constrained project scheduling problems. *Computers & Operations Research* , pp. 51-62.

III. Bellenguez-Morineau, O., & Néron, E. (2007). A branch-and-bound method for solving multi-skill scheduling problem. *RAIRO Operations Research* (41(2)), pp. 155-170.

IV. Bellenguez-Morineau, O., & Néron, E. (2004). Lower bounds for the multi-skill project scheduling problem with hierarchical levels of skills. *Practice and Theory of Automated Timetabling (PATAT2004)*, (pp. 429-432.).

V. Billaut, J., Moukrim, A., & Sanlaville, E. (2010). *Flexibility and Robustness in Scheduling. Control systems, robotics and manufacturing series.* Wiley.

VI. Blazewicz, J., Lenstra, J., & Rinnooy, K. (1983). Scheduling subject to resource constraints: Classification and complexity. *Discrete Applied Mathematics* (5(1)), pp. 11-24.

VII. Chtourou, H., & Haouari, M. (2008). A two-stage-priority-rule-based algorithm for robust resource-constrained project scheduling. *Computers and Industrial Engineering* (55(1)), pp. 183–194.

VIII. Deb, K., Agrawal, S., Pratap, A., & Meyariva, T. (2000). *A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II.* Technical report, Kanpur Genetic Algorithms Laboratory (Kan-GAL), Indian Institute of Technolog.

IX. Demeulemeester, E., & Herroelen, W. (1992). A branch-and-bound procedure for the multiple resource-constrained project scheduling problem. *European Journal of Operational Research* (38(12)), pp. 1803-1818.

X. Demeulemeester, E., & Herroelen, W. (1997). New benchmark results for the resource-constrained project scheduling problem. *Management Science, 43* (11), pp. 1485-1492.

XI. Drezet, L.-E. (2005). *Résolution d’un problème de gestion de projets sous contraintes de ressources humaines: de l’approche prédictive a l’approche réactive.*PhD thesis, University of Tours.

XII. Hartmann, S. (1998). A competitive genetic algorithm for resource constrained project scheduling. *Naval Research Logistics* (45(7)), pp. 733-750.

XIII. Sprecher, A., & Drexl, A. (1998). Multi-mode resource-constrained project scheduling by a simple, general and powerful sequencing algorithm. *European Journal of Operational Research, 2* (107), pp. 431-450.

XIV. Van de Vonder, S., Demeulemeester, E., & Herroelen, W. (2007). A classification of predictive-reactive project scheduling procedures. *Journal of Scheduling* (10(3)), pp. 195-207.

XV. Van Veldhuizen, D., & Lamont, G. (2000). On measuring multi-objective evolutionary algorithm performance. *Congress of evolutionary computation*, (pp. 204-211).

XVI. Walter, M., & Zimmermann, J. (2010). Staffing projects with multi-skilled workers in matrix organisations. *12th International Workshop on Project Management and Scheduling (PMS2010)*, (pp. 387-391).

XVII. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C., & Grunert da Fon-seca, V. (2003). Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation, 7* (2), pp. 117-132.

XVIII. Dhib, C., Soukhal, A., & Néron, E. (2013). Etude et résolution de problèmes d’ordonnancement de projets multi-compétences : intégration à un progiciel intégré libre . PhD thesis, University of Tours.

XIX. Dhib, C., Soukhal, A., & Néron, E. (2011a). A multi-skill project scheduling problem in an industrial context. *International conference in Industrial and Systems Engineering Proceedings*, (pp. 316-325).

XX. Dhib, C., Kooli, A., Soukhal, A., & Néron, E. (2011b). Lower bounds for a multi-skill scheduling problem. *Operations Research Proceedings*, (pp. 471-476).

XXI. Dhib, C., Soukhal, A. , & Néron, E. (2014). Lower Mixed-Integer Linear Programming Formulation and Priority-Rule Methods for a Preemptive Project Staffing and Scheduling Problem. Handbook on Project Management and Scheduling , (pp. 603-617).

Author' biography with Photo

Dr.Cheikh Dhib obtaining his Ph.D in computer Science from University Francois Rabelais Tours, Tours, France.From 2013 until now. My research interest is mainly about Combinatorial optimization, Scheduling, Resource Constrained Project Scheduling Problem(RCPSP) and more precisely the multi-skill project scheduling problem(MSPSP) and their applications.

Dr.Emmanuel Néron is a professor at Politechnic school of Tours, France and since 2013 he is the director of the school. He is a curious researcher and rigorous academic his research mainly in the domains of Combinatorial optimization, Scheduling, Resource Constrained Project Scheduling Problem(RCPSP) and more precisely the multi-skill project scheduling problem(MSPSP) and their applications.

Dr.Ameur Soukhal is a professor at Politechnic school of Tours, France. He is a curious researcher and rigorous academic. His research mainly in the domains of Combinatorial optimization, Scheduling, Resource Constrained Project Scheduling Problem(RCPSP) and their applications.

Dr.Hafedh Mohamed babou obtained his Ph.D in Computer Science from university of nantes, France. He is curious researcher and rigorous academic with good background combinatorics and bioinformatics.

He is currently lecturer at school of mines- Nouakchott-mauritania

Dr.Bedine Kerim obtained his Ph.D in Computer Science from university of Le Havre, Le Havre, France. He is curious researcher and rigorous academic with good background in artificial intelligence, mathematical modeling, game theory, machine learning, cloud computing, Fuzzy Logic.

He is currently assistant professor in the college of computer science and information technology at Albaha University-KSA