



**HAL**  
open science

# MATHEMATICAL FORMULATION AND APPROXIMATE METHODS FOR AN EXTENSION OF THE RCPSP : THE MULTI-SITE RCPSP

Arnaud Laurent, Nathalie Grangeon, Laurent Deroussi, Sylvie Norre

► **To cite this version:**

Arnaud Laurent, Nathalie Grangeon, Laurent Deroussi, Sylvie Norre. MATHEMATICAL FORMULATION AND APPROXIMATE METHODS FOR AN EXTENSION OF THE RCPSP : THE MULTI-SITE RCPSP. CIE 45 45th International Conference on Computers & Industrial Engineering, 2015, Metz, France. hal-02023753

**HAL Id: hal-02023753**

**<https://hal.science/hal-02023753>**

Submitted on 11 Mar 2019

**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.

# MATHEMATICAL FORMULATION AND APPROXIMATE METHODS FOR AN EXTENSION OF THE RCPSP : THE MULTI-SITE RCPSP

Arnaud Laurent, Nathalie Grangeon, Laurent Deroussi, Sylvie Norre

---

## Abstract

This article proposes an extension of the Resource Constrained Project Scheduling Problem: the Multi-Site RCPSP with resource pooling between several sites. This extension considers new constraints for the RCPSP like transportation times and choice of the site where tasks are performed. A linear program of this problem is given. Four approximate methods are described: local search, simulated annealing and Iterated Local Searches with two different acceptance criteria: Simulated Annealing type acceptance criterion and Better Walk acceptance criterion. We compare the results obtained with each method. Simulated Annealing and Iterated Local Searches give good results.

*Keywords:* RCPSP, Multi-Site, Scheduling, Transportation time, Resource pooling, Metaheuristic

---

## 1. INTRODUCTION

Resource pooling management for multi-site organisations receives an increasing interest from decision workers. Resources are shared between several sites and have the possibility to move from one site to another. A lot of practical applications can be mentioned to illustrate the interest of resource pooling:

- Factory 4.0 with production cells that can move from a site to another, at the tactical level
- The moving of the construction machinery between different building sites

- Hospital systems, in which health personnel can work on several sites (“Groupement Hospitalier de Territoire” in the recent French health law)

This kind of considerations leads us to propose a new variant of RCPSP called the Multi-Site RCPSP because literature variants do not permit to model the resource moves, when some resources and tasks have to be assigned on a site. So this paper is devoted to the presentation and the study of the Multi-Site RCPSP. It is organised as follows. First we present the problem and the economic impact of resource pooling. Secondly we present a literature review of similar problems. Then we propose resolution methods, a mathematical model and approximate methods for solving the proposed problem. We present some results of the mathematical model and of the approximate methods. Finally we conclude about our work and our prospects.

## 2. PRESENTATION OF THE MULTI-SITE RCPSP

### 2.1. Description of the problem

This problem is an extension of the classical Resource Constraint Project Scheduling Problem (RCPSP) in which a set of  $N$  tasks has to be scheduled. The duration of the task  $j$  is  $p_j$ . Each task has a set  $P_j$  of precedence relations. Each task  $j$  needs a set of  $r_{j,k}$  resources of type  $k$ , for each  $K$  types. Mathematical models have been proposed by Oğuz and Bala[24] and Correia and al.[3].

Because of the multi-site context, new characteristics which are not considered in the classical RCPSP have to be used. We first introduce the notion of site. Each task needs a site to be performed. The second concept is the distinction between fixed resources and mobile resources. For example, a fixed resource can be a machine that can not be moved. Thus, its use for a task will determine the site where the task will be assigned. A fixed resource can not be assigned to a task performed on a site where the resource is not located. In contrast, a mobile resource can execute tasks on every site.

If a mobile resource executes two consecutive tasks on two different sites, time constraints must be considered to model the transportation time of the resource from one site to another. Mobile resources are available on the site where they realized their first task. A transportation time results in a minimum delay between the end of the execution of the first task and the beginning of the execution of the second task. This time depends on the pair

of sites where both tasks are performed. There is another case in which a transportation time is applied: when two tasks are linked by a precedence relation and are realized on different sites. In this case, it represents the transportation time of a semi-finished product between two sites.

This problem extends the definition of the RCPSP. If only one site is considered for an instance of the RCPSP multi-site, the remaining problem is a classical RCPSP. As the RCPSP is NP-hard in the strong sense, this extension is NP-hard too.

### *2.2. Example*

To illustrate this new problem, we consider an instance composed of seven tasks and five resources of 4 different types. There are two resources of type 1 (R1,1 and R1,2), one of types 2,3 and 4 (R2, R3 and R4 respectively). R1,1 and R2 are assumed to be fixed, the other ones are mobile. There are two sites. It takes 2 time periods to travel from one site to the other one. Precedence graph is represented in Figure 1. Each task is represented by a circle, the duration of the task is written above the circle, the needed resources are below. Tasks 1 and 9 are fictitious tasks of beginning and end of the project.

A schedule is given in Figure 2. With this solution, two mobile resources are moving. R3 performs task 2 and 4 on site 1, and then goes to site 2 to perform task 5. At the same time, R1,2 performs task 2 and 4 on site 1 and then moves to site 2 to perform task 6. Thus, two transportation times are applied for mobile resources. Two more transportation times are applied: between tasks 2 et 3 and between tasks 4 and 5. Indeed these tasks are linked by precedence relation and they are not performed on the same site. The makespan of this solution is equal to 12 periods.

### *2.3. Economic relevance*

The practical application for which we define this problem comes from the public health sector. A pool of human resources is shared on several distant hospitals within a community. The involved problem is to find a hospital assignment for the patients and their operations and for each operation, to assign the needed resources. The hospitals are distant, so the patients and human resources have to take into account transportation times. The goal is to improve the productivity by pooling human resources and patients within the community. Other applications could be imagined (production sites with

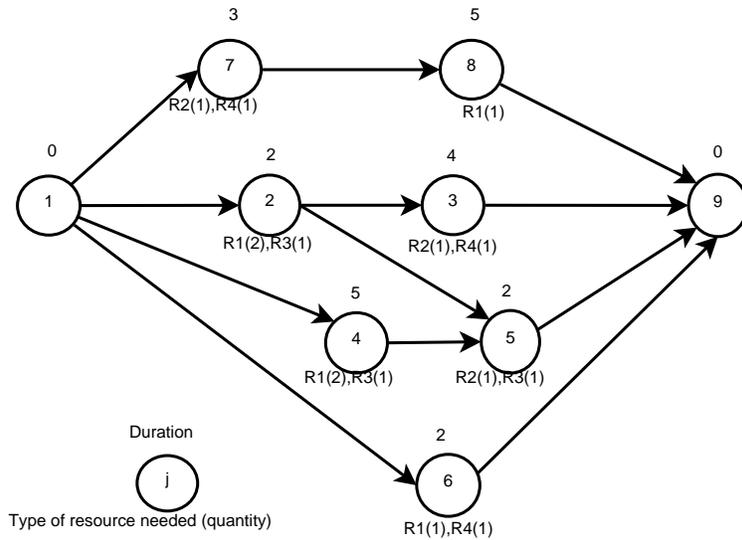


Figure 1: Precedence graph

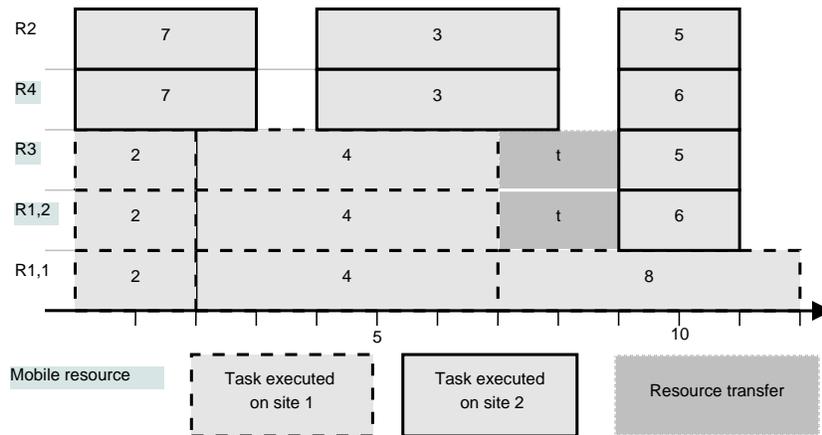


Figure 2: Gantt chart of an optimal solution

shared machines, multi-site time tabling, ...), rising yet the interest of taking into account the resource transport in a project scheduling context.

We present a small example in order to illustrate the interest of resource pooling. We consider a hospital community with 2 sites. On these sites, 13 medical examinations have to be programmed. The transportation time between the two sites is equal to 1 period. On site 1 there are a scanner (S1) and 2 manipulators (M3 and M4). On site 2 there are a MRI (Magnetic

Resonance Imaging) (MR1) and 2 manipulators (M1 and M2). The medical examinations are:

- 9 MRI, which last 1 period each
- 4 scanners, which last 2 periods each

In this case, we consider 11 patients with one or two medical examinations to do. The examinations 1 and 13 involve the same patient. The examinations 3 and 12 also involve the same patient. In the case of several medical examinations, precedence constraints can be used to model the patient. Medical examinations from 1 to 9 are MRI examinations and the ones from 10 to 13 are scanner examinations. A scanner and a MRI examinations require a manipulator and a scanner and a MRI respectively, to be executed. The examinations 2, 4, 7, 11 and 13 involved patients with reduced mobility and require a second manipulator to do their exam.

When each resource is assigned to a site, an optimal solution is given in Figure 3. This schedule has a makespan of 9 periods. In this case none of the resources are pooled, the resources can work only on their employing units. In the case of manipulator pooling between the two sites a new schedule can be proposed.

When resources are pooled and can move from one site to another, an optimal schedule is given by figure 4 with only 3 manipulators. In this schedule the manipulator M3 executes 3 medical examination on site 1 and then goes to the site 2 to execute 3 MRI. This solution has still a makespan of 9 unless only three manipulators are consumed.

Gourgand and *al.*[7] have shown the interest of resource pooling at tactical level for the same practical problem. Their problem consists in planning the medical examinations at a day granularity without consideration of the schedule of activities. They consider 100 medical examinations to plan in a week. Without resource pooling, only 88 can be planned in the time horizon. When the same resources are pooled between the sites, all the medical examinations are planned in the time horizon.

There are several interests in resource pooling in this context. The schedule can be more robust to material failure or the lack of personnel. There is also the economic impact on the consumption of the resources. We present a small example in order to illustrate that the resource pooling reduces the consumption of resources.

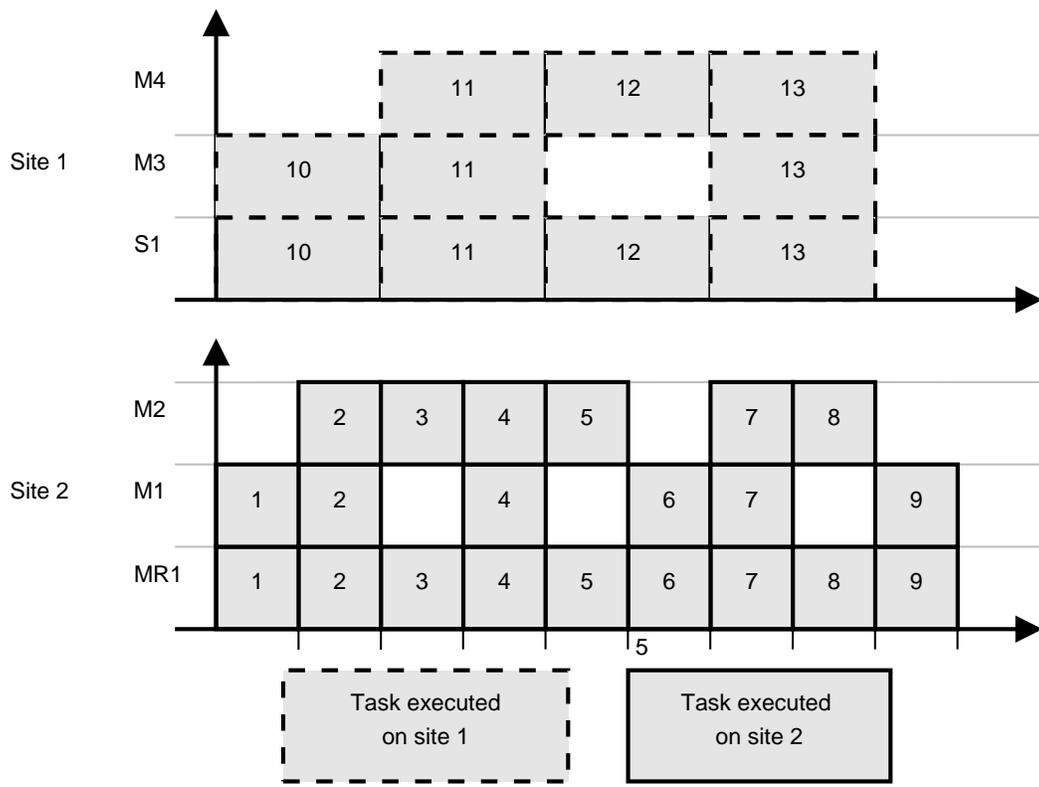


Figure 3: Schedule with no resource pooling

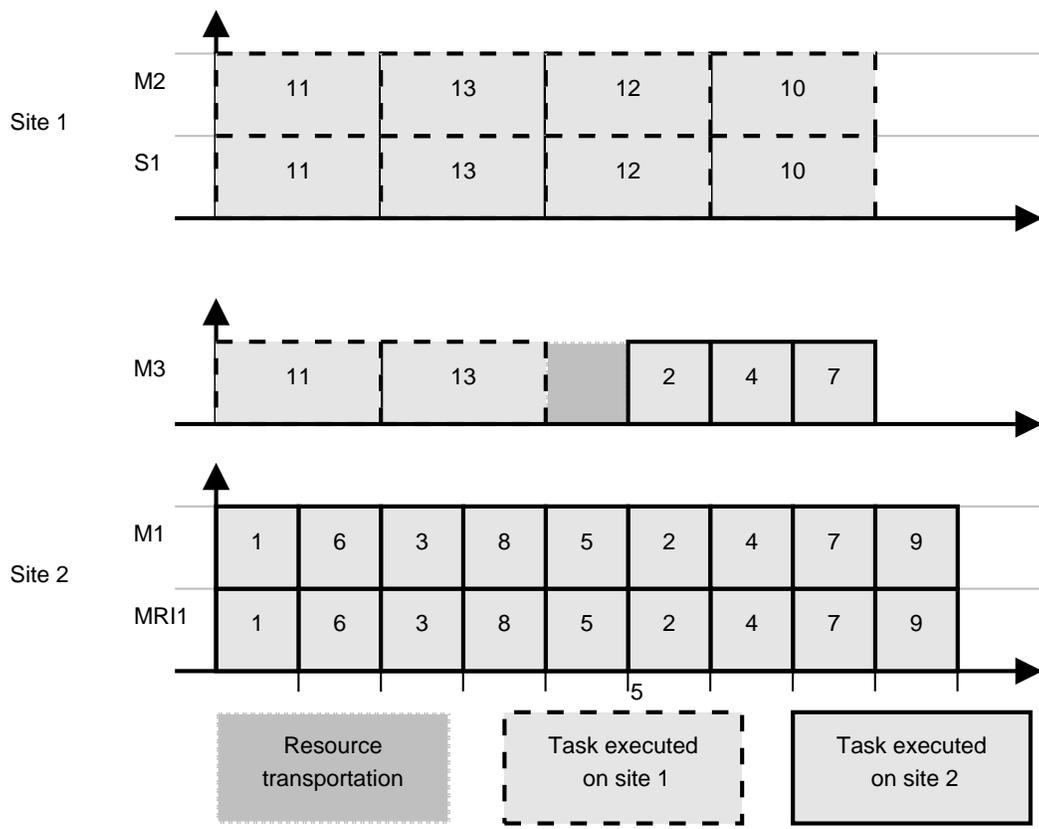


Figure 4: Schedule with resource pooling

### 3. SIMILAR PROBLEMS IN LITERATURE

#### 3.1. Extensions of the RCPSP

The classical Resource Constraint Project Scheduling Problem, written  $PS|prec|C_{max}$  by Kan[9], consists in scheduling each task and assigning resources to it. A version of this problem exists with several modes of execution for each task. This problem is called MRCPSP for Multi-mode RCPSP.

Two extensions of the RCPSP and three extensions of the MRCPSP are related to our problem:

- RCPSP with minimum time lags (RCPSP min) proposed by Klein[11]
- RCPSP with conditional minimum time lags (RCPSP-CTL) proposed by Toussaint[27]
- MRCPCP with generalized precedence relations (RCPSP-GPR) proposed by De Reyck and Herroelen[4]
- Multi-mode Resource-Constrained Project Scheduling Problem with Mode Dependent Time Lags proposed by Sabzehparvar and Seyed-Hosseini[26]
- MRCPSP with schedule dependent set-up time (RCPSP-SST) proposed by Mika and al.[23]

All of these RCPSP extensions can model transportation times. The RCPSP with minimum time lags considers a time between the end of a task and the beginning of another task. This time is known and given for each pair of tasks. With the notations given in section 3 and a minimum time lag  $lagmin(i, j)$  between tasks  $i$  and  $j$ .

This extension can model transportation time between two tasks, like the transportation time of a job from a machine to the next one. Many studies have focused on the integration of these minimum time lags such as Klein[11], Klein[12], Lombardi and Milano[18].

In the RCPSP with minimum conditional time lags proposed by Toussaint[27], the minimum time lag is applied only in two cases:

- There is a precedence relation between the tasks
- At least one resource is transmitted from one task to another

This extension allows us to model transportation time of a resource and semi-finished product which has to move from the location where the first task is performed to the location where the second task is performed.

The Multi-mode Resource-Constrained Project Scheduling Problem with Generalized Precedence Relations (MRCPSp-GPR) proposed by De Reyck and Herroelen[4] added minimum and maximum time lags. This problem can also model transportation between tasks.

The main interest of the multi mode aspect is that a task has different ways to be executed, so a mode can model a way to execute a task on a specific site. If this is the case, the time lag between two tasks will depend on the mode used to execute each task. A model is proposed by Sabzehparvar and Seyed-Hosseini[26] for this problem called Multi-mode Resource-Constrained Project Scheduling Problem with Mode Dependent Time Lags. For each couple of tasks the time lag between them depends on which mode is applied for each task. This problem allows to model transportation time between tasks without the knowledge of where each task will be executed.

Another way to model transportation time with the MRCPSp is proposed by Mika and al.[23]. In this problem the authors model transportation times by set-up time between each task. In addition of a mode, a task needs a location to be executed. Therefore, the set-up time between two tasks is schedule dependent, which means that it depends on the sequence of tasks and resources assignment. Every resource is fixed on a location and if two tasks need to transfer set-up required resources, the set-up time will depend on which location each task is executed.

There is also some papers that address the Resource Constraint Multi-Project Scheduling Problem (RCMPSP) with transfer time between each project. Yang and Sum[28] study different priority rules for resources assignment in a multi-project problem with resource transfer time. Several other substantial work on this problem have been done by Dodin and Elimam[5], Yang and Sum[29] and Krüger and Scholl[14].

### *3.2. Transportation in scheduling problems*

Transportation constraints have been considered in many other scheduling problems. One of them, considered by Maggu and Das[20] and Maggu and al.[21], is the flow shop problem with two machines. Several other papers consider transportation times in flow shop problems with more than two machines, as Maggu and al.[21], Kise[10].

Some papers deal with hybrid flow shop problem. For instance in Langston [16], the transportation times between two tasks are deduced according to the pair of machines assigned to execute the job. There are a lot of papers which consider a vehicle to carry jobs, as in Lee and Chen[17]. For the job shop problem, Bilge and Ulusoy [1] were interested in the simultaneous scheduling of transportation resources and production resources. There is still a lot of work done on these extensions like Kumar and Kumar[15] and Gupta[8].

Another type of scheduling problem is the multiprocessor scheduling problem proposed by Garey and Johnson[6]. This problem considers a transportation time if two tasks with precedence relation are not performed by the same processor.

To conclude, all the different extensions of the RCPSP can not model our problem. None of them can model the assignment of resources to sites. The goal of our problem is to add a transportation time to the RCPSP which depends on the location where tasks are performed. Compared to the literature, we extend the RCPSP with conditional time lags by adding the assignment of tasks to the site. Time lags become transportation times, because they are no longer given for a pair of tasks, but for a pair of sites.

#### 4. MATHEMATICAL MODEL

The extension consists in adding the multi-site aspect to the classical RCPSP problem. So, two new elements are added: the sites and the fact that a resource is mobile (it can move between two sites) or not (a resource is assigned to a site and cannot move to another site). Transportation times are the main new aspect. Two distinct cases must be considered:

- Two tasks consecutively assigned to a same mobile resource are realized on two distinct sites. A time lag must be taken into account between the end of the first task and the beginning of the second, it corresponds to the transportation time of the resource between the two sites.
- There is a precedence constraint between two tasks and these two tasks are assigned to two distinct sites. The time lag between the end of the first task and the beginning of the second corresponds to the transportation time between the two sites (the result of the first task must be moved to the second task).

Unlike the RCPSP with conditional time-lags, lags are not known in advance because they depend on the sites the tasks are assigned to.

#### 4.1. Data

- $N$  Number of tasks, with 1 and  $N$  the two fictitious tasks of beginning and end of the project
- $p_j$  Duration of task  $j = 1, N$
- $P_j$  Set of tasks which have to precede task  $j = 1, N$
- $K$  Number of different types of resources
- $R_k$  Number of resources of type  $k = 1, K$
- $r_{j,k}$  Number of resources of type  $k = 1, K$  needed for task  $j = 1, N$
- $T$  Maximum number of periods
- $M_{k,r} = 1$  if resource  $r = 1, R_k$  of type  $k = 1, K$  is mobile, 0 otherwise
- $S$  Number of sites
- $\delta_{s,s'}$  Transportation time between site  $s = 1, S$  and site  $s' = 1, S$
- $loc_{k,r}$  Location site for resource  $r = 1, R_k$  of type  $k = 1, K$
- $H$  A large number

#### 4.2. Variables

- $X_{j,t} = 1$  if task  $j = 1, N$  ends in period  $t = 1, T$ , 0 otherwise
- $Y_{j,k,r} = 1$  if resource  $r = 1, R_k$  of type  $k = 1, K$  is assigned to task  $j = 1, N$ , 0 otherwise
- $Z_{j,s} = 1$  if task  $j = 1, N$  is performed on site  $s = 1, S$ , 0 otherwise
- $\omega_{j,h} = 1$  if a transportation time has to be applied between the end of task  $j = 1, N$  and the beginning of task  $h = 1, N$ , 0 otherwise. When two tasks are consecutively assigned to a same resource or when two tasks are subject to a precedence constraint. However, the corresponding transportation time will be equal to 0 if the two tasks are assigned to a same site.

#### 4.3. Objective

$$\text{Minimize } \sum_{t=1}^T t \times X_{N,t} \quad (1)$$

#### 4.4. Constraints

$$\sum_{t=1}^T X_{j,t} = 1; j = 1, N; \quad (2)$$

$$\omega_{h,j} = 1; j = 1, N; h \in P_j; \quad (3)$$

$$Y_{j,k,r} + Y_{h,k,r} \leq \omega_{j,h} + \omega_{h,j} + 1; \quad (4)$$

$$1 \leq j < h \leq N; k = 1, K; r = 1, R_k; \quad (4)$$

$$\sum_{t=1}^T t \times X_{j,t} \geq \sum_{t=1}^T t \times X_{h,t} + p_j + (Z_{j,s} + Z_{h,s'} - 1) \times \delta(s, s') - H \times (1 - \omega_{h,j}); \quad (5)$$

$$j, h = 2, N - 1; s, s' = 1, S; \quad (5)$$

$$\sum_{t=1}^T t \times X_{j,t} \geq \sum_{t=1}^T t \times X_{h,t} + p_j; \quad (6)$$

$$((j = N) \wedge (h = 2, = N - 1)) \cup ((h = 1) \wedge (j = 2, N - 1)); \quad (6)$$

$$\sum_{r=1}^{R_k} Y_{j,k,r} = r_{j,k}; j = 1, N; k = 1, K; \quad (7)$$

$$Y_{j,k,r} \leq Z_{j,loc_{k,r}}; j = 1, N; (k = 1, K; r = 1, R_k;) \wedge (M_{k,r} = 0); \quad (8)$$

$$\sum_{s=1}^S Z_{j,s} = 1; j = 1, N; \quad (9)$$

$$X_{j,t} \in \{0; 1\}; j = 1, N; t = 1, T; \quad (10)$$

$$Y_{j,k,r} \in \{0; 1\}; j = 1, N; k = 1, K; r = 1, R_k; \quad (11)$$

$$Z_{j,s} \in \{0; 1\}; j = 1, N; s = 1, S; \quad (12)$$

$$\omega_{j,h} \in \{0; 1\}; j, h = 1, N; \quad (13)$$

The proposed model is based on the literature models from Oğuz and Bala[24] and Correia and al.[3]. The aim is to minimize the makespan (1). The non-preemption and the realization of each task are expressed by the constraint (2). The constraints (3), (4) and (5) model that a transportation time must be taken into account in the two following cases:

- Two tasks are subject to a precedence constraint (constraint (4))
- A same resource is assigned to two tasks. In this case, these two tasks cannot overlap (constraint (5)).

The transportation times are equal to zero if the two tasks are assigned to a same site. The constraint (5) computes the completion time of the tasks by integrating the transportation times which depend on the sites tasks are assigned to. It can be noticed that this modeling can be applied in the case of conditional time lags for which the notion of site does not appear. The term  $(Z_{j,s} + Z_{h,s'} - 1) \times \delta(s, s')$  must be simply replaced by the time lag between the tasks  $j$  and  $h$ . The completion times of the fictitious tasks are computed by constraint (6). The constraint (7) expresses that the right amount of resources is assigned to a task. If no mobile resource is assigned to a task, this task must be assigned to the associated site (constraint (8)). Each task is assigned to one and only one site (9). Constraints (10)(11)(12)(13) are binary constraints.

In this model, we propose to consider each resource individually. Thus it is easier to deal with the site assignment and to take into account the transportation times. The standard constraint in RCPSP which expresses that the number of assigned resources to tasks at a given time is lower than the number of available resources does no longer appear as such. Constraints (5) and (7) both mention this point. At a given time, either a resource is assigned to a task or moves between two sites.

## 5. APPROXIMATE APPROACHES

As this problem is NP-hard in the strong sense, approximate methods form a good alternative to solve large instances. Among them, metaheuristics are a family of generic methods that are considered to be efficient for solving hard problems. We propose to implement individual-based metaheuristics. We describe first the solution representation used for this problem. Then, we define two neighborhood systems before we present the metaheuristics.

### 5.1. Solution representation and list-scheduling based algorithm

A solution is represented by the concatenation of two vectors  $X = (\sigma, l)$ . The vector  $\sigma$  is the sequence of tasks while the vector  $l$  assigns a site to each task. More formally,

- $\sigma = (\sigma_1, \dots, \sigma_N)$  is a permutation of the tasks. We assume that we will only work with valid permutations, e.g. permutations for which the precedence constraints are all satisfied.  $\sigma_1$  and  $\sigma_N$  are two fictitious project tasks, representing respectively the beginning and the ending

of the project. Resources have to respect the execution order of the tasks. Thus, if task  $j$  precedes task  $i$  in  $\sigma$  then no resource can execute  $i$  before  $j$ .

- $l = (l_1, \dots, l_N)$  with  $l_j \in \{1, \dots, S\}$  is the site where the task  $j$  will be performed. Only the sites having enough resources for processing a task can be selected.

Starting from a solution thus defined, a list-scheduling based algorithm is then applied, inspired by the works of Carlier[2]. This list algorithm permits to schedule the tasks by computing the earliest ending date  $d_j$  of each task  $j$  as shown below.

For each resource  $r$  of type  $k$ , its availability date  $av_{k,r}$  is determined, according to three possible cases:

- The resource  $r$  is mobile. Its availability date is defined by  $av_{k,r} = 0$  if the current task  $j$  is the first one that is assigned to  $r$ ;  $av_{k,r} = d_h + \delta(l_h, l_j)$  if  $h$  is the last task assigned to  $r$  before the current task  $j$ .
- The resource  $r$  is fixed and  $l_j \neq loc_{k,r}$ . The resource cannot be assigned to  $j$  and by convention  $av_{k,r} = \infty$ .
- The resource  $r$  is fixed and  $l_j = loc_{k,r}$ . The availability task is  $av_{k,r} = d_h$  if  $h$  is the last task assigned to  $r$ .

For each required type  $k$ , the  $r_{j,k}$  earliest available resources are assigned to the task  $j$  ( $Y_{j,k,r} = 1$ ). Then, its completion time  $d_j$  can be determined using the formula (14) to (16).  $A$  represents the date for which all the preceding tasks of  $j$  are completed (including the transportation times).  $B$  ensures that all the resources required for the execution of task  $j$  are available.

$$A = \max_{h \in P_j} (d_h + \delta(l_h, l_j)) \quad (14)$$

$$B = \max_{k \in \{1, \dots, K\}; r \in \{1, \dots, R_k\} / Y_{j,k,r} = 1} (av_{k,r}) \quad (15)$$

$$d_j = \max(A, B) + p_j \quad (16)$$

The list-scheduling based algorithm is shown in algorithm 1.

---

**Algorithm 1** List-scheduling based algorithm for the multi-site RCPSP

---

**Require:**  $X = (\sigma, l)$

$av := \{0, \dots, 0\};$

$d := \{0, \dots, 0\};$

**for**  $j = \sigma_1$  to  $\sigma_N$  **do**

    Computation of the availability date of each resource  $av_{k,r}$

    Assignment of the selected resources to the task  $j$

    Computation of the completion time  $d_j$  (Eq. 14 to 16)

**end for**

---

### 5.2. Neighborhood system

The neighborhood system used in the different methods is composed of two basic moves. The first one is dedicated to the permutation of the tasks, and the second one concerns the site assignment  $l$ . At each iteration, each move is randomly chosen. Let us describe these moves.

#### 5.2.1. Permutation of the tasks

The first move modifies the tasks by applying an insertion move. A task is moved from a position  $p$  to another position  $p'$  ( $p' \neq p$ ). A move is said to be feasible if the resulting permutation of tasks satisfies the precedence constraints. This neighborhood system is implemented such that only the feasible moves are considered. This means in particular that the starting and the ending project task respectively stay in the first and last position.

#### 5.2.2. Site-assignment move

The second move modifies the site assigned to a task  $j$ . Like the previous one, this move preserves the feasibility of the solution. For each type of resources, the sum of the fixed resources located in the new site and the mobile resources must be lower than the amount of resources required for executing the task  $j$ .

### 5.3. Resolution methods

We use 4 different methods:

- a local search (LS)
- an inhomogeneous Simulated Annealing (SA) proposed by Metropolis and al.[22]. The principle is to do a local search but with a time-based

decreasing probability to accept lowest quality solutions. The probability to accept a lowest quality solution  $X'$  with a current solution  $X$  is given by equation 17.

$$p(X', i) = \exp\left(\frac{H(X) - H(X')}{T_i}\right) \quad (17)$$

The value  $T_i$  corresponds to the temperature at iteration  $i$ . The temperature follows a geometric sequence with common ratio  $\alpha < 1$ . We use the algorithm of Romeo and Sangiovanni-Vincentelli [25] to fix the initial temperature  $T_0$  for the simulated annealing. At each iteration the temperature is multiplied by  $\alpha$ . The value of  $\alpha$  is given by the following equation:

$$\alpha = \sqrt[iterMax]{\frac{T_a}{T_0}} \quad (18)$$

$T_a$  represents the final temperature. For our experimentations the final temperature is set to 0.001.

- an Iterated Local Search proposed by Lourenço and al.[19] (algorithm 2)

---

**Algorithm 2** Algorithm of the Iterated Local Search (ILS)

---

**Require:**  $X_0$  : initial solution;

$X^* \leftarrow$  local search on  $X_0$ ;

**while** Stopping criterion is not satisfied **do**

$X' \leftarrow$  Perturbation of  $X^*$ ;

$X' \leftarrow$  Local search on  $X'$ ;

$X^* \leftarrow$  Acceptance criterion of  $X'$  to  $X^*$  taking into consideration the history;

**end while**

**return**  $X^*$

---

The two acceptance criteria used are:

- Better walk:  $X''$  is returned if  $H(X'') < H(X^*)$ , otherwise  $X^*$
- Simulated Annealing acceptance type criterion (17)

We only use one neighborhood system which consists in applying V1 and V2 with the same probability. The stopping criterion for the local searches in the ILS is reached when 500 iterations with no upgrade of the solution are done.

## 6. RESULTS

The goal of this section is to compare the results obtained by the different metaheuristics in a short amount of time. First, we test our resolution methods by solving literature instances of the classical RCPSP for which the optimal solution is known. In the second part, we adapt these instances to our problem and we solve them. We compute our experimentations on a processor Intel(R) Xeon(R) CPU E7-8870 @ 2.40GHz. The mathematical model has been solved with IBM ILOG CPLEX Optimization Studio v 12.4. Metaheuristics have been implemented in JAVA 1.7. To compare the metaheuristics we use the four approximate methods described previously with a maximum of 100k iterations (stopping criterion). Each method is run 20 times by instance with the parameters described in the previous part.

### 6.1. RCPSP Results

Our problem is an extension of the classical RCPSP, so we want to test our method on this problem first before applying it to our problem. To do that we use the PSPLIB, a library of instances for the RCPSP. This library is composed of four sets of instances with 30, 60, 90 and 120 tasks. For each set there are 480 instances of 48 different classes. A class is a set of 10 instances generated with the same parameters. All optimal solutions (OPT) for the 30 tasks instances of PSPLIB are known, so we compare our results with the optimal solution of each instance. To this particular case where all the resources are considered on a unique site (classical RCPSP) we do not use the site-assignment move. We compare the results obtained with the four methods, to the optimal solution. We present the number of classes where all the optimal solution are found by the metaheuristic (row 2), and the number of classes where no instance is solved optimally (row 3). For all obtained solutions, we compute the relative gap (RG) compared to the optimal solutions for 30 tasks instances (row 4 and 5). The results are presented in table 1.

The obtained results show that these methods perform well for the classical problem. These results are good but they don't imply that our methods

Table 1: Results obtained on RCPSP instances

Metaheuristic	LS	<i>ILS</i>  BW	<i>ILS</i>  SA	SA
# of classes solved to OPT	20	<b>31</b>	30	29
# of classes where no OPT is found	1	<b>0</b>	<b>0</b>	<b>0</b>
Average RG for all instances	0,60	0,16	0,16	<b>0,13</b>
Worst RG for an instance	15,28	5,26	<b>3,57</b>	<b>3,57</b>
% of instances solved to optimal	80,42	92,08	91,67	<b>92,29</b>

work well on the studied problem. We now have to test our methods for the multi-site RCPSP.

## 6.2. Multi-site RCPSP

### 6.2.1. Results of the mathematical model

We test the mathematical model presented in section 4. The literature instances of the PSPLIB [13] have instances from 30 to 120 tasks, which are too big for our mathematical model. So we create smaller instances with 2 or 3 sites and 10 or 20 resources. In total, 192 small instances have been generated with 5, 10, 15, 20, 25, 30 tasks. We run CPLEX on our model and stop it if it has not finished after 30 minutes. For the instances not solved in less than 30 minutes, we give the Relative Gap (RG) between the Lower (LB) and the Upper Bound (UB). The results on these instances are given in the table 2 depending on the number of tasks, in table 3 depending on the number of resources and table 4 depending on the number of sites.

# of tasks	% of execution finished	Average RG between LB and UB
5	100	0
10	100	0
15	87.5	2.2
20	9.4	20.1
25	12.5	34.91
30	0	54.09
All	51.56	18.69

Table 2: Results depending on the number of tasks

The computation times are 0.89 seconds and 56.33 seconds respectively for 5 and 10 tasks instances. The other size of instances didn't finish within

# of resources	% of execution finished	Average RG between LB and UB
10	54.17	15.58
20	48.96	21.80
All	51.56	18.69

Table 3: Results depending on the number of resources

# of sites	% of execution finished	Average RG between LB and UB
2 sites	52.08	15.87
3 sites	51.04	21.50
All	51.56	18.69

Table 4: Results depending on the number of sites

the 30 minutes limit. The results show that the model gives optimal results or good results for instances with 15 tasks or less. For instances with 20 tasks and more we do not obtain satisfactory results (RG > 20% between the lower and upper bound). The number of sites and resources also impact the results (table (3) (4)).

### 6.2.2. Results of approximate methods

To test our approximate methods on this new problem, we need to create instances. We adapt literature instances of the PSPLIB created by Kolisch and Sprecher[13]. To transform these instances into RCPSP Multi-Site instances, we add to them 2 or 3 sites. Each resource is fixed with a probability of 0,5. Fixed resources are assigned on site randomly. We set transportation time between each site with a value in the same range as the duration of the tasks. These instances are available at [http : //www.isima.fr/ ~ laurenta/RCPSPMS](http://www.isima.fr/~laurenta/RCPSPMS). The best known solution is also provided on the web site. For each method the Computational Time (CT) is nearly the same, we report the interval of value of the CT for one replication in table 5.

Table 5: Results obtained on Multi-site RCPSP instances with 3 sites

Number of tasks	Min CT (s)	Max CT (s)	Average CT (s)
30	9,9	163,5	61,1
60	28,8	522,4	199,8
90	37,8	682,3	276,7
120	73,4	1026,2	353,4

The best results found by all the methods is noted as the Best Known Solution ( $BKS_i$ ) for instance  $i$ . For all the replications  $j = 1, 20$  with the method  $m$  on instance  $i = 1, 480$ , the value obtained is noted  $x_{i,j}^m$ .

$$BKS_i = \min_{\forall m; \forall j=1,20} x_{i,j}^m; \quad \forall i = 1, 480 \quad (19)$$

We compute the relative gap for an instance  $i$  ( $RG_i^m$ ) compared to BKS with the equation 20.

$$RG_i^m = \frac{\sum_{j=1}^{20} \frac{x_{i,j}^m - BKS}{BKS}}{20} \quad (20)$$

The Average Relative Gaps ( $RG_i^m$ ) are computed for each instance and  $RGC_c^m$  (equation 21) for each class  $c$ . We report in the results table the  $RGAv^m$ , the Average Relative Gap for all instances (equation 22).

$$RGC Av_c^m = \frac{\sum_{i \in c} RG_i^m}{10}; \quad (21)$$

$$RG Av^m = \frac{\sum_{j=1}^{480} RG_i^m}{480}; \quad (22)$$

We report in the result table the number  $NBC^m$  of time each method  $m$  gives the best Average Relative Gaps ( $RGCAv_c^m$ ) for a class  $c$ . The results are presented in table 6 for instances with 2 sites and table 7 for instances with 3 sites.

Table 6: Results obtained on Multi-site RCPSP instances with 2 sites

Metaheuristic	LS	<i>ILS</i>  BW	<i>ILS</i>  SA	SA
Results for instances with 30 tasks				
<i>RGAv<sup>m</sup></i>	8,13	<b>2,09</b>	2,95	3,40
<i>NBC<sup>m</sup></i>	0	<b>39</b>	10	0
Results for instances with 60 tasks				
<i>RGAv<sup>m</sup></i>	9,22	6,13	8,97	<b>5,75</b>
<i>NBC<sup>m</sup></i>	1	18	1	<b>28</b>
Results for instances with 90 tasks				
<i>RGAv<sup>m</sup></i>	12,26	11,28	14,55	<b>9,41</b>
<i>NBC<sup>m</sup></i>	4	4	0	<b>40</b>
Results for instances with 120 tasks				
<i>RGAv<sup>m</sup></i>	13,59	13,48	18,16	<b>10,21</b>
<i>NBC<sup>m</sup></i>	16	2	0	<b>30</b>

Table 7: Results obtained on Multi-site RCPSP instances with 3 sites

Metaheuristic	LS	<i>ILS</i>  BW	<i>ILS</i>  SA	SA
Results for instances with 30 tasks				
<i>RGAv<sup>m</sup></i>	9,55	<b>3,26</b>	5,27	4,69
<i>NBC<sup>m</sup></i>	0	<b>45</b>	3	0
Results for instances with 60 tasks				
<i>RGAv<sup>m</sup></i>	16,81	17,51	22,45	<b>13,59</b>
<i>NBC<sup>m</sup></i>	0	10	0	<b>38</b>
Results for instances with 90 tasks				
<i>RGAv<sup>m</sup></i>	10,22	11,48	16,59	<b>7,74</b>
<i>NBC<sup>m</sup></i>	6	0	0	<b>42</b>
Results for instances with 120 tasks				
<i>RGAv<sup>m</sup></i>	10,83	14,75	20,76	<b>9,27</b>
<i>NBC<sup>m</sup></i>	18	0	0	<b>30</b>

The optimal solution (OPT) for an instance of the RCPSP is now a lower bound of the optimal solution of the new instance created. The Best Known Lower Bound (BKLB) for RCPSP in literature is also a lower bound for RCPSP Multi-Site. We report in this table, the Relative Gap ( $RGLB_i$ )

between the Best Known Lower Bound ( $BKLB_i$ ) in the literature (can be optimal) and the Best Known Solution ( $BKS_i$ ) for RCPSP Multi-Site instance  $i$ .

$$RGLB_i = \frac{BKS_i - BKLB_i}{BKS_i}; \quad \forall i = 1, 480 \quad (23)$$

We report in the results table the Average Relative Gap to Lower Bound  $RGLBAv$  for all instances.

$$RGLBAv = \frac{\sum_{i=1}^{480} RGLB_i}{480}; \quad (24)$$

We report these values in the table 8 for instances with 2 sites and 9 for instances with 3 sites.

Table 8: Relative gap to lower bounds for instances with 2 sites

Number of tasks	RGLB Av
30	11,01%
60	14,17%
90	17,92%
120	37,37%

Table 9: Relative gap to lower bounds for instances with 3 sites

Number of tasks	RGLB Av
30	17,12%
60	17,69%
90	34,80%
120	63,67%

All the instances and the best solutions found to this day are available at <http://www.isima.fr/~laurenta/RCPSPMS>. On the small instances (30 tasks) the best results are obtained by the iterated local search with local search acceptance criterion. The worst results are obtained with the local search: there are a lot of local minimums in which the local search is trapped. For the other instances with more than 60 tasks, the simulated annealing gives the best results in average. The iterated local searches clearly didn't converge to the optimal within the 100K iterations limit, but increase the number of iterations will also increase the computation times.

## 7. CONCLUSION

In this paper, we presented a new problem that extends the classical Resource Constrained Project Scheduling Problem. We modelled this problem as an integer linear programming. We proposed a solution representation composed of two vectors with a list algorithm to schedule the tasks and assign the resources. We created and solved a set of instances based on the literature ones. We shown that the metaheuristics behave well or not depending on the size of the problem.

Our work on the multi-site RCPSP generalizes the problems of scheduling with transportation time. Moreover, our resolution method is easily adjustable to consider new constraints such as incompatibility resources or availability resources. These new constraints could be considered in the list algorithm.

One of our highest priority is to improve our methods on largest instances. We could also see the impact on results if we use a solution structure with one or more vectors. Another perspective is the use of population-based metaheuristics to explore different task distributions on sites.

- [1] Bilge, Ü., Ulusoy, G., 1995. A time window approach to simultaneous scheduling of machines and material handling system in an FMS. *Operations Research* 43 (6), 1058–1070.  
URL <http://dx.doi.org/10.1287/opre.43.6.1058>
- [2] Carlier, J., 1984. Problèmes d’ordonnancement à contraintes de ressources : algorithmes et complexité. Ph.D. thesis, Thèse de doctorat d’état, Université Paris VI, Paris.
- [3] Correia, I., Lourenço, L. L., Saldanha-da Gama, F., 2012. Project scheduling with flexible resources: formulation and inequalities. *OR spectrum* 34 (3), 635–663.
- [4] De Reyck, B., Herroelen, W., 1999. The multi-mode resource-constrained project scheduling problem with generalized precedence relations. *European Journal of Operational Research* 119 (2), 538–556.
- [5] Dodin, B., Elimam, A., 1997. Audit scheduling with overlapping activities and sequence-dependent setup costs. *European Journal of Operational Research* 97 (1), 22–33.

- [6] Garey, M. R., Johnson, D. S., 1979. Computers and intractability: a guide to the theory of NP-completeness. WH Freeman & Co., San Francisco.
- [7] Gourgand, M., Grangeon, N., Klement, N., 2015. Activities planning and resources assignment on distinct places: a mathematical model. *RAIRO - Operations Research* 49 (1), 79–98.  
URL <http://dx.doi.org/10.1051/ro/2014028>
- [8] Gupta, D., 2015. 3-stage specially structured flow shop scheduling to minimize the rental cost including transportation time, job weightage and job block criteria. *European Journal of Business and Management* 7 (4), 1–6.
- [9] Kan, A. R., 1976. Machine scheduling problem: Classification , complexity and computations. Martinus Nijhoff.
- [10] Kise, H., 1991. On an automated two-machine flowshop scheduling problem with infinite buffer. *J. Oper. Res. Soc. Japan* 34, 354–361.
- [11] Klein, R., 2000. Project scheduling with time-varying resource constraints. *International Journal of Production Research* 38 (16), 3937–3952.  
URL <http://www.tandfonline.com/doi/abs/10.1080/00207540050176094>
- [12] Klein, R., Scholl, A., 2000. Progress: Optimally solving the generalized resource-constrained project scheduling problem. *Mathematical Methods of Operations Research* 52 (3), 467–488.  
URL <http://dx.doi.org/10.1007/s001860000093>
- [13] Kolisch, R., Sprecher, A., 1997. {PSPLIB} - a project scheduling problem library: {OR} software - {ORSEP} operations research software exchange program. *European Journal of Operational Research* 96 (1), 205 – 216.  
URL <http://www.sciencedirect.com/science/article/pii/S0377221796001701>
- [14] Krüger, D., Scholl, A., 2010. Managing and modelling general resource transfers in (multi-) project scheduling. *OR spectrum* 32 (2), 369–394.

- [15] Kumar, P., Kumar, M., 2015. Production scheduling in a job shop environment with consideration of transportation time and shortest processing time dispatching criterion. *International Journal of Advanced Engineering Research and Applications*.
- [16] Langston, M. A., 1987. Interstage transportation planning in the deterministic flow-shop environment. *Operations Research* 35 (4), 556–564. URL <http://dx.doi.org/10.1287/opre.35.4.556>
- [17] Lee, C.-Y., Chen, Z.-L., 2001. Machine scheduling with transportation considerations. *Journal of Scheduling* 4 (1), 3–24.
- [18] Lombardi, M., Milano, M., 2009. A precedence constraint posting approach for the RCPSP with time lags and variable durations. In: Gent, I. (Ed.), *Principles and Practice of Constraint Programming - CP 2009*. Vol. 5732 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, pp. 569–583. URL <http://dx.doi.org/10.1007/978-3-642-04244-7-45>
- [19] Lourenço, H. R., Martin, O. C., Stutzle, T., 2001. Iterated local search. *Handbook of metaheuristics*.
- [20] Maggu, P., Das, G., 1980. On  $2 \times n$  sequencing problem with transportation times of jobs. *Pure and Applied Mathematika Sciences* 12 (1), 6.
- [21] Maggu, P. L., Das, G., Kumar, R., 1981. On equivalent-job for job-block in  $2 \times n$  sequencing problem with transportation-times. *Journal of the Operations Research Society of Japan* 24 (2), 136–146.
- [22] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., Teller, E., 1953. Equation of state calculations by fast computing machines. *The journal of chemical physics* 21 (6), 1087–1092.
- [23] Mika, M., Waligora, G., Weglarz, J., 2004. A metaheuristic approach to scheduling workflow jobs on a grid. In: *Grid resource management*. Springer, pp. 295–318.
- [24] Oğuz, O., Bala, H., 1994. A comparative study of computational procedures for the resource constrained project scheduling problem. *European Journal of Operational Research* 72 (2), 406 – 416.

URL [http://www.sciencedirect.com/science/article/pii/0377221794903190](http://www.sciencedirect.com/science/article/pii/S0377221794903190)

- [25] Romeo, F. I., Sangiovanni-Vincentelli, A., 1984. Probabilistic hill climbing algorithms: Properties and applications. Electronics Research Laboratory, College of Engineering, University of California.
- [26] Sabzehparvar, M., Seyed-Hosseini, S. M., 2008. A mathematical model for the multi-mode resource-constrained project scheduling problem with mode dependent time lags. *The Journal of Supercomputing* 44 (3), 257–273.
- [27] Toussaint, H., 2010. Algorithmique rapide pour les problèmes de tournées et d’ordonnancement. Ph.D. thesis, Université Blaise Pascal-Clermont-Ferrand II.
- [28] Yang, K.-K., Sum, C.-C., 1993. A comparison of resource allocation and activity scheduling rules in a dynamic multi-project environment. *Journal of Operations Management* 11 (2), 207–218.
- [29] Yang, K.-K., Sum, C.-C., 1997. An evaluation of due date, resource allocation, project release, and activity scheduling rules in a multiproject environment. *European Journal of Operational Research* 103 (1), 139–154.