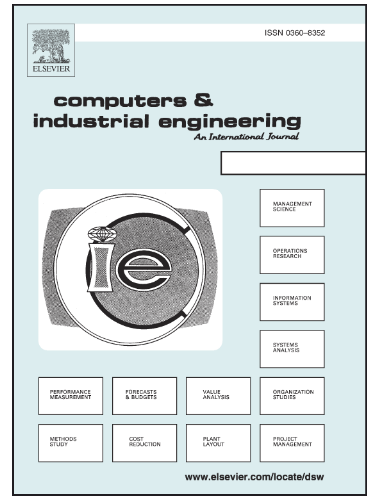
Accepted Manuscript



A new extension of the RCPSP in a Multi-Site context: Mathematical Model and Metaheuristics

A. Laurent, L. Deroussi, N. Grangeon, S. Norre

PII: S0360-8352(17)30330-3
DOI: http://dx.doi.org/10.1016/j.cie.2017.07.028
Reference: CAIE 4835

To appear in: *Computers & Industrial Engineering*

Accepted Date: 19 July 2017

A new extension of the RCPSP in a Multi-Site context: Mathematical Model and Metaheuristics

A. Laurent, L. Deroussi, N. Grangeon and S. Norre
LIMOS CNRS UMR 6158, Université Blaise Pascal,
Campus Universitaire des Cézeaux, 2 rue de la Chebarde, TSA 60125, CS 60026, 63173 Aubière Cedex, France.
arnaud.laurent@isima.fr
{laurent.deroussi,nathalie.grangeon,sylvie.norre}@moniut.univ-bpclermont.fr

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 Search with two different acceptance criteria: Simulated Annealing type acceptance criterion and Better Walk acceptance criterion. We compare the results obtained with each method. The best results are obtained with Simulated Annealing and Iterated Local Search metaheuristics.

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

A new extension of the RCPSP in a Multi-Site context: Mathematical Model and Metaheuristics

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 Search with two different acceptance criteria: Simulated Annealing type acceptance criterion and Better Walk acceptance criterion. We compare the results obtained with each method. The best results are obtained with Simulated Annealing and Iterated Local Search metaheuristics.

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 a 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 et 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 determines the site where the task must 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 this resource from the first site to the second one. 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. In other words, the transportation times that must be applied are schedule dependent. 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. These tasks require no resource and no site.

A feasible 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. The corresponding transportation times are represented by the letter "t" on the Gantt Chart. Moreover, transportation times are also required between pair of tasks (2, 3), (2,5), (4,5), and (7,8) because these tasks are linked by precedence relation and they are not performed on the same site. We can notice that no transportation times are applied when the precedence constraints concern a fictitious task. 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

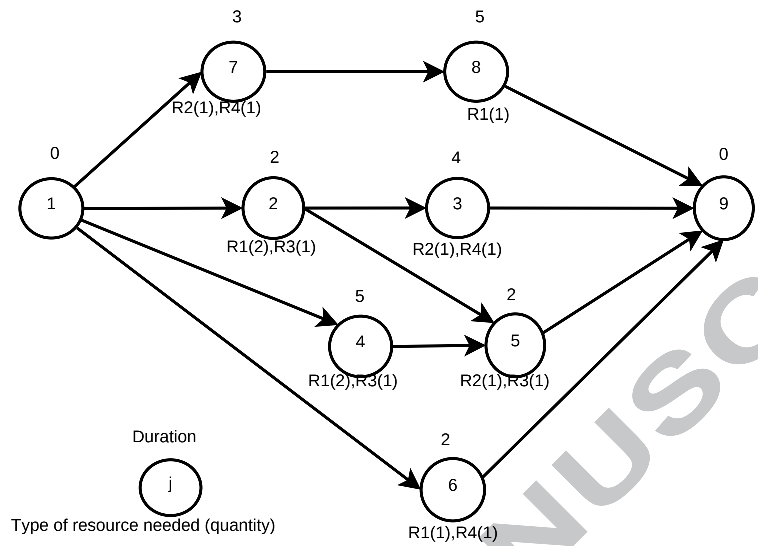


Figure 1: Precedence graph

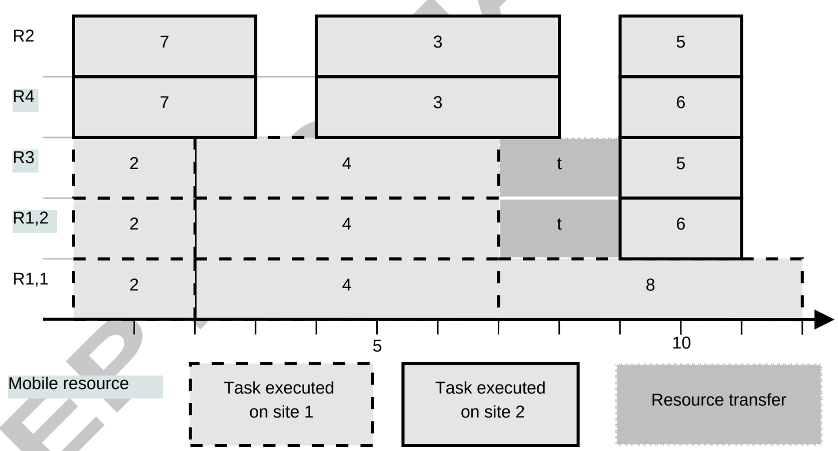


Figure 2: Gantt chart of an optimal solution

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 that the resource pooling reduces the consumption of resources. 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 involve patients with reduced mobility and require a second manipulator to be performed.

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 some resources (manipulators) 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.

This work is in the continuity of Gourgand et al.[7]. The authors 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:

- increase of the productivity of the system,
- increase of the robustness to material failure or lack of personnel,
- economic impact on the consumption of the resources.

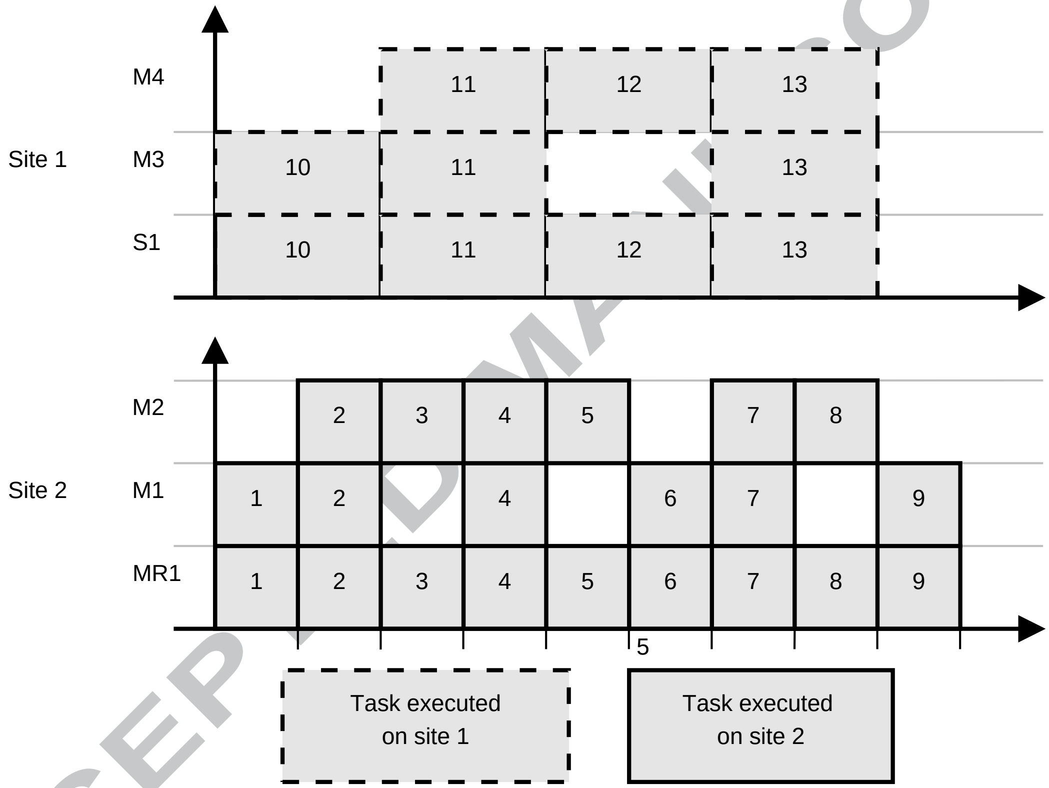


Figure 3: Schedule with no resource pooling

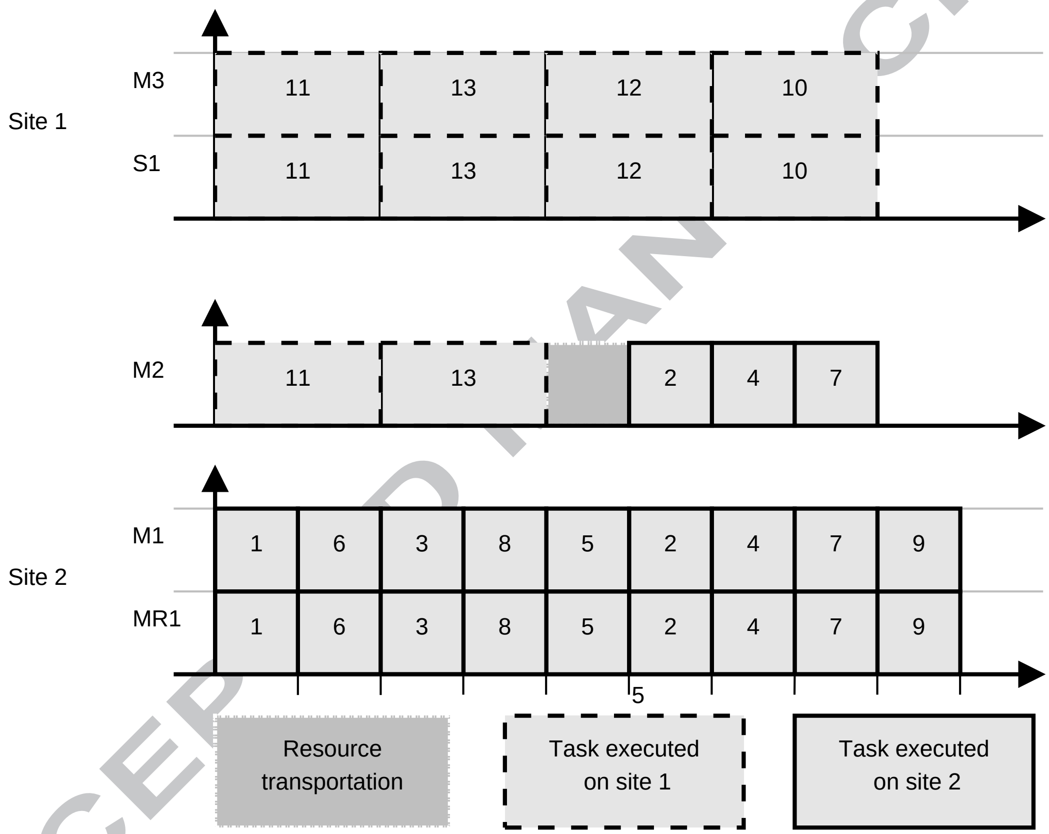


Figure 4: Schedule with resource pooling

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 et al.[23]

All 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.

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 have 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. 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 et 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 are other 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 has 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 et 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 et 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, the literature extensions of the RCPSP do not allow to model our problem. None of them takes into account 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, respectively
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	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. A transportation time is applied 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

$$Minimize \sum_{t=1}^{T} t \times X_{N,t} \tag{1}$$

4.4. Constraints

$$\sum_{t=1}^{T} X_{j,t} = 1; j = 1, N; \quad (2)$$
$$\omega_{h,j} = 1; j = 1, N; h \in P_j; \quad (3)$$
$$Y_{j,k,r} + Y_{h,k,r} \leq \omega_{j,h} + \omega_{h,j} + 1;$$
$$1 \leq j < h \leq N; k = 1, K; r = 1, R_k; \quad (4)$$
$$\sum_{t=1}^{T} t \times X_{j,t} \geq \sum_{t=1}^{T} t \times X_{h,t} + p_j + (Z_{j,s} + Z_{h,s'} - 1) \times \delta_{s,s'} - H \times (1 - \omega_{h,j});$$
$$j, h = 2, N-1; s, s' = 1, S; \quad (5)$$
$$\sum_{t=1}^{T} t \times X_{j,t} \geq \sum_{t=1}^{T} t \times X_{h,t} + p_j;$$
$$((j = N) \wedge (h = 2, = N-1)) \cup ((h = 1) \wedge (j = 2, N-1)); \quad (6)$$
$$\sum_{r=1}^{R_k} Y_{j,k,r} = r_{j,k}; j = 1, N; k = 1, K; \quad (7)$$
$$Y_{j,k,r} \leq Z_{j,loc_{k,r}}; j = 1, N; (k = 1, K; r = 1, R_k;) \wedge (M_{k,r} = 0); \quad (8)$$
$$\sum_{s=1}^{S} Z_{j,s} = 1; j = 1, N; \quad (9)$$
$$X_{j,t} \in \{0; 1\}; j = 1, N; t = 1, T; \quad (10)$$
$$Y_{j,k,r} \in \{0; 1\}; j = 1, N; k = 1, K; r = 1, R_k; \quad (11)$$
$$Z_{j,s} \in \{0; 1\}; j = 1, N; s = 1, S; \quad (12)$$
$$\omega_{j,h} \in \{0; 1\}; j, h = 1, N; \quad (13)$$

The proposed model is based on the literature models of Oğuz and Bala[24] and Correia et 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 which 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. METAHEURISTICS 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, 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 σ is the sequence of tasks while the vector l assigns a site to each task. More formally,

- $\sigma = (\sigma_1, ..., \sigma_N)$ is a sequence of tasks. We assume that we will only work with feasible sequence, e.g. sequence for which the precedence constraints are all satisfied. σ_1 and σ_N are two fictitious project tasks, representing respectively the beginning and the ending of the project.

In order to preserve the feasibility of the sequence, we fix $\sigma_1 = 1$ and $\sigma_N = N$. Resources have to respect the execution order of the tasks. Thus, if task j precedes task i in σ then no resource can execute i before j.

- $l = (l_1, ..., l_N)$ with $l_j \in \{1, ..., S\}$ 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 $av_{k,r} = \infty$. This case leads to a non feasible solution.

- 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.

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

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

$$d_j = max(A, B) + p_j \tag{16}$$

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

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

Require: $X = (\sigma, l)$
  $av := \{0, ..., 0\}$;
  $d := \{0, ..., 0\}$;
  for $j = \sigma_2$ to σ_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. It consists in choosing at random a new site $l'_j \in \{1, ..., S\}/l'_j \neq l_j$ such that the feasibility is preserved. 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 et al.[22]. The principle is to do a local search but with a time-based

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

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

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 α. The value of α is given by the following equation:

$$\alpha = \sqrt[iterMax]{\frac{T_a}{T_0}} \tag{18}$$

T_a represents the final temperature. For our experimentations the final temperature is set to 0.001. This value has been experimentally fixed.

- an Iterated Local Search proposed by Lourenço et 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 $H(X') \leq H(X^*)$, otherwise X^*
- Simulated Annealing acceptance type criterion (17)

Better walk consists in a pure intensification search around the best found local minimum in the subset of local minimum. The Simulated

Annealing acceptance criterion permits to introduce diversification. So acceptance criteria is tuned according to the same rule than the metaheuristic except for the stopping criteria. The number of iterations is equal to the number of local searches.

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 5000 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 and mathematical model 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 first solve small instances and then we adapt the literature 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 for the 30 tasks instances. The methods are denoted:

- Local Search (LS)
- Simulated Annealing (SA)
- Iterated Local Search with the Better Walk acceptance criterion (ILS|BW)
- Iterated Local Search with the Simulated Annealing acceptance criterion (ILS|SA)

We present the number of classes where all the optimal solutions are found by the metaheuristics (row 2), and the number of classes where no instance is solved optimally (row 3). We compute the best, the average and the worst relative gap (RG) compared to the optimal solutions (row 4, 5 and 6). We compute the percentage of instances solved to optimal (row 7). The results are presented in table 1.

Table 1: Results obtained on RCPSP instances

Metaheuristic	LS	ILS|BW	ILS|SA	SA
# of classes solved to OPT	20	**31**	30	29
# of classes where no OPT is found	1	**0**	**0**	**0**
Best RG for all instances	**0,00**	**0,00**	**0,00**	**0,00**
Average RG for all instances	0,60	0,16	0,16	**0,13**
Worst RG for all instances	15,28	5,26	**3,57**	**3,57**
% of instances solved to optimal	80,42	92,08	91,67	**92,29**

More than 90% of instances are solved to optimality by metaheuristics (methods ILS and SA). This shows that these methods are effective, although they have been designed to work on the multi-site context. This allows to validate our approach with a single site before extending it to several sites.

6.2. Multi-site RCPSP

The goal of this section is to give results on the multi-site RCPSP. We present in the first subsection the results obtained with the mathematical model and approximate methods on small size instances, before we present the results obtained with approximate methods on PSPLIB-inspired instances.

6.2.1. Results on small size instances

Results with mathematical model. In order to validate our mathematical model, and to study the solution limit of this model, we propose to generate a library of small size instances of the multi-site RCPSP. This library is composed of 192 instances, divided in 24 classes. Each class contains 8 instances and is characterized by a number of sites (2 or 3), a number of resources (10 or 20) and a number of tasks (5, 10, 15, 20, 25 or 30). For each instance, CPLEX is executed and stopped when it reached a time limit of 3600 seconds. For the remaining unsolved instances, we compute the Relative Gap (RG) between the Lower Bound (LB) and the UB (Upper Bound). The results obtained are given in table 2, 3 and 4, according to the three defined characteristics (respectively the number of tasks, resources and 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 of the mathematical model depending on the number of tasks

# 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 of the mathematical model depending on the number of resources

In average, the smallest instances are solved in 0.89 second (5 tasks) and 56.33 (10 tasks) seconds. As it would be expected, we can see that the number of tasks has a major impact on the results. Instances with less than 15 tasks are generally solved to optimality within the allocated time limit, whereas only few of them are solved with 20 or 25 tasks (around 10%) and

# 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 of the mathematical model depending on the number of sites

none of them with 30 tasks. The impact of the two other characteristics is weak. We can only notice that the RG seems to increase with the number of resources and sites (it exceeds 20% for 20 resources or 3 sites).

Results with metaheuristics . Table 5 gives the results obtained by our solution methods (LS, ILS|BW, ILS|SA and SA) for each class of instances with N>= 10 tasks.

The best results found by all the methods are noted as the Best Known Solution (BKS_i) for instance i. We note:

- M the set of methods used ($M = \{LS; SA; ILS|BW; ILS|SA\}$)
- $x_{i,j}^m$ the value of the makespan obtained by the jth ($j = 1, 20$) replication of the method m on instance i ($i = 1, 480$).
- x_i^{MM} the solution obtained by the Mathematical Model on instance i (the optimal solution for solved instances and the upper bound otherwise).
- BKS_i (named best known solution) the best makespan obtained on instance i ($i = 1, 480$). It is given by equation (19).

$$BKS_i = \min\{\min_{\forall m; \forall j=1,20} x_{i,j}^m, x_i^{MM}\}; \quad \forall i = 1, 480 \tag{19}$$

- $AvBKS_c$ the average makespan of the best known solution BKS_i for all instances i in class c. It is given by equation (20).

$$AvBKS_c = \frac{\sum_{i \in c} BKS_i}{8} \tag{20}$$

- $AvRG_i^m$ the average relative gap between the makespan obtained by method m (for the 20 replications) and the best known solution for an instance i. It is given by equation (21)

$$AvRG_i^m = \frac{\sum_{j=1}^{20} \frac{x_{i,j}^m - BKS_i}{BKS_i}}{20} \tag{21}$$

- $AvRG^m$ the average gap between the solution obtained by method m and the best known solution for all instances. It is given by equation (22)

$$AvRG^{m,d} = \frac{\sum_{i=1}^{480} AvRG_i^m}{160} = \frac{\sum_{c=1}^{160} AvRGC_c^m}{20} \tag{22}$$

The metaheuristics find optimal solutions or near optimal solution in a very short computational time. For N>=20 tasks, the metaheuristics find better solutions than in few seconds. We recall that the time limit for MM is fixed at 3600 seconds. These results show a suitable behaviour of the metaheuristics. SA seems to give the best results for the smallest instances whilst ILS—BW performs well on the largest ones. We propose in the next section to apply them on more challenging instances, built from the PSPLIB instances.

6.2.2. Results on the PSPLIB-inspired library

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]. The PSPLIB instances are adapted for the multi-site case in the following way:

- We consider that the project takes place in several sites. For each instance of the PSPLIB, we generate two instances: one with two sites and one with three sites.
- Two sites are distant and a transportation time is randomly computed in the same range than for the task duration. Each element (task or resource) that goes from one site to another one is unavailable during all the transport duration.
- For each resource, it is randomly determined if it is fixed or mobile (with a probability of 50%).Fixed resources are randomly assigned to one site and cannot move to another one.

				LS	ILS—BW	ILS—SA	SA	MM
N	S	Resources	$AvBKS^c$	$AvRG_c^{LS}$	$AvRG_c^{ILS\|BW}$	$AvRG_c^{ILS\|SA}$	$AvRG_c^{SA}$	$AvRG_c^{MM}$
10	2	10	26,25	1,09	0,16	0,12	**0,00**	**0,00**
	3	10	18,13	5,62	0,72	0,79	0,20	**0,00**
	2	20	28,25	2,45	0,06	0,25	**0,00**	**0,00**
	3	20	25	4,12	0,85	0,50	**0,00**	**0,00**
15	2	10	46	2,73	0,64	0,50	0,52	**0,00**
	3	10	29,38	9,36	2,23	1,95	2,25	**0,00**
	2	20	39,88	4,17	0,64	0,60	0,57	**0,00**
	3	20	39,38	10,30	2,80	3,28	3,29	**0,00**
20	2	10	62,50	2,12	0,93	0,94	0,86	**0,00**
	3	10	45,38	9,98	3,42	4,36	4,61	**0,27**
	2	20	58,38	5,53	**2,09**	2,37	3,20	4,83
	3	20	49	11,96	**3,45**	3,46	5,30	16,26
25	2	10	64,75	6,50	**2,26**	2,53	3,57	2,58
	3	10	56,25	13,73	**4,85**	5,38	7,15	8,94
	2	20	70,88	9,35	3,80	**3,25**	4,71	9,57
	3	20	72,63	9,89	**2,80**	2,99	4,31	25,95
30	2	10	79,25	4,42	2,18	**1,94**	4,13	X
	3	10	77,63	4,20	2,58	**2,46**	2,86	X
	2	20	74,13	11,40	**5,26**	5,69	6,93	X
	3	20	80,88	16,53	**7,22**	8,00	10,01	X
		Average	52,78	7,27	**2,45**	2,57	3,22	X

Table 5: Results obtained on the small size instances

- A test is done to ensure the feasibility of the instances. For each task, at least one site must allow its execution. This means that for each type of resources, this site must have a potential number of resources (mobile and fixed) greater than the required ones. If an instance does not satisfy this feasibility criterion, it is rejected and another one is generated.

These instances are available at $http://www.isima.fr/\sim laurenta/RCPSPMS.html$. 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 6.

Table 6: Results obtained on Multi-site RCPSP instances with 3 sites

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

We use the same notation as described in section 6.2.1. We just redefine the best known solution for an instance as equation (23) (the mathematical model has not been applied on these large instances).

$$BKS_i = \min_{\forall m \in M; \forall j=1,20} x^m_{i,j}; \quad \forall i = 1,480 \tag{23}$$

We report in the table 7 and table 8 the number NBC^m of time each method m gives the best Average Relative Gaps ($AvRG^m_c$) for a class c. The results are presented in table 7 for instances with 2 sites and table 8 for instances with 3 sites.

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.

$$RGLB_i = \frac{BKS_i - BKLB_i}{BKS_i}; \quad \forall i = 1,480 \tag{24}$$

Table 7: Results obtained on Multi-site RCPSP instances with 2 sites

Metaheuristic	LS	ILS|BW	ILS|SA	SA
Results for instances with 30 tasks				
$Min_{i=1,480}\ AvRG_i^m$	**0,00**	**0,00**	**0,00**	**0,00**
$AvRG^m$	8,13	**2,09**	2,95	3,40
$Max_{i=1,480}\ AvRG_i^m$	35,00	**12,34**	14,89	15,54
NBC^m	0	**39**	10	0
Results for instances with 60 tasks				
$Min_{i=1,480}\ AvRG_i^m$	**0,00**	**0,00**	**0,00**	**0,00**
$AvRG^m$	9,22	6,13	8,97	**5,75**
$Max_{i=1,480}\ AvRG_i^m$	61,13	**30,65**	57,42	44,03
NBC^m	1	18	1	**28**
Results for instances with 90 tasks				
$Min_{i=1,480}\ AvRG_i^m$	**0,00**	**0,00**	**0,00**	**0,00**
$AvRG^m$	12,26	11,28	14,55	**9,41**
$Max_{i=1,480}\ AvRG_i^m$	52,36	55,44	59,37	**45,95**
NBC^m	4	4	0	**40**
Results for instances with 120 tasks				
$Min_{i=1,480}\ AvRG_i^m$	**0,21**	0,41	0,72	0,41
$AvRG^m$	13,59	13,48	18,16	**10,21**
$Max_{i=1,480}\ AvRG_i^m$	77,56	72,44	**62,33**	79,30
NBC^m	16	2	0	**30**

Table 8: Results obtained on Multi-site RCPSP instances with 3 sites

Metaheuristic	LS	ILS|BW	ILS|SA	SA
Results for instances with 30 tasks				
$Min_{i=1,480}\ AvRG_i^m$	**0,00**	**0,00**	**0,00**	**0,00**
$AvRG^m$	9,55	**3,26**	5,27	4,69
$Max_{i=1,480}\ AvRG_i^m$	29,83	**15,81**	24,03	22,10
NBC^m	0	**45**	3	0
Results for instances with 60 tasks				
$Min_{i=1,480}\ AvRG_i^m$	**0,00**	**0,00**	**0,00**	**0,00**
$AvRG^m$	16,81	17,51	22,45	**13,59**
$Max_{i=1,480}\ AvRG_i^m$	72,76	**68,10**	69,11	**68,10**
NBC^m	0	10	0	**38**
Results for instances with 90 tasks				
$Min_{i=1,480}\ AvRG_i^m$	**0,33**	0,38	1,44	**0,00**
$AvRG^m$	10,22	11,48	16,59	**7,74**
$Max_{i=1,480}\ AvRG_i^m$	**38,63**	39,88	47,25	36,88
NBC^m	6	0	0	**42**
Results for instances with 120 tasks				
$Min_{i=1,480}\ AvRG_i^m$	0,96	4,89	8,62	**0,45**
$AvRG^m$	10,83	14,75	20,76	**9,27**
$Max_{i=1,480}\ AvRG_i^m$	58,48	57,44	61,98	**42,42**
NBC^m	18	0	0	**30**

We compute the Average Relative Gap to Lower Bound $AvRGLB$ for all instances.

$$AvRGLB = \frac{\sum_{i=1}^{480} RGLB_i}{480}; \tag{25}$$

We report these values in the table 9 for instances with 2 sites and table 10 for instances with 3 sites.

Table 9: Relative gap to lower bounds for instances with 2 sites

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

Table 10: 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.html*. On the small instances (30 tasks) the best results are obtained by the iterated local search with better walk 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 solution within the 100K iterations limit, but increasing the number of iterations would also increase the computation times. The choice of Local Search (an optimization schema but not really a metaheuristic according to the Osman definition) and simple individual-based metaheuristics, directly inspired from local search technique (iterated local search and simulated annealing a local search in which is embedded an exploration mechanism thanks to the temperature) is justified because the studied problem seems very challenging and the results obtained for the largest instances show that local search performs as good as metaheuristics. This means that the search space is too large for metaheuristics and the number of iterations is not big enough. The exploration mechanisms don't have enough time to be effective, and the exploitation mechanism is predominant.

Additional experiments (with longer runs) show that the results obtained by metaheuristics on these instances are improved.

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 small instances and a set of instances based on the literature ones. We also solved the small instances with the mathematical model. 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 resources incompatibilities or resources availabilities. 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× 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× 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

[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.

*Highlights (for review)

A RCPSP extension: the multisite RCPSP with scheduling dependent transportation times

A MILP for multi site RCPSP including conditional minimum time lags

Solution approaches: proposition and comparison of 4 individual based metaheuristics