CIRRELT

Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation

A Heuristic Method for Resource-Constrained Project Scheduling with Activity Overlapping

Lucas Grèze
Robert Pellerin
Patrice Leclaire
Nathalie Perrier

December 2012

CIRRELT-2012-75

Bureaux de Montréal :
Université de Montréal
C.P. 6128, succ. Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

Université Laval · UQÀM · HEC Montréal · École Polytechnique Montréal · Université de Montréal

A Heuristic Method for Resource-Constrained Project Scheduling with Activity Overlapping

Lucas Grèze[1], Robert Pellerin[1,2,*], Patrice Leclaire[3], Nathalie Perrier[1]

[1] Department of Mathematical and Industrial Engineering, École Polytechnique de Montréal, C.P. 6079, succursale Centre-ville, Montréal, Canada H3C 3A7

[2] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

[3] Institut Supérieur de Mécanique de Paris (Supméca), 3 rue Fernand Hainaut, 93407 Saint-Ouen, Cedex France

Abstract. The overlapping of activities is a common practice to accelerate the execution of engineering projects. This technique consists in executing in parallel two activities, normally executed in a sequential way, by allowing the downstream activity to start before the end of the upstream activity based on preliminary information. In this paper, we propose a constructive heuristic for the Resource-Constrained Project Scheduling Problem with Overlapping Modes (RCPSP-OM). Given a set of activities to execute, the RCPSP-OM consists in determining the order of execution in time of a set of activities so as to minimize the total project duration, while respecting precedence relations, resource constraints and overlapping possibilities. The heuristic implies that rework tasks related to overlapping are added to downstream activities and that the consumption of the resources is constant throughout the execution of the project (including rework). The method also considers that the possible overlapping modes for every couple of activities and the duration of rework tasks associated with every mode are known in advance. Results show that, when the objective consists in minimizing the project duration, the consideration of the costs associated to activity overlapping allows to significantly reducing the cost of reworks. On the other hand, when the objective consists in maximizing the gains related to the project execution, the search for the best trade-off between acceleration and increase of project costs enables to avoid losses.

Keywords. Activity overlapping, concurrent engineering, project management, project scheduling.

Acknowledgments. This work has been supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Jarislowsky/SNC-Lavalin Research Chair in the Management of International Projects. This support is gratefully acknowledged.

---

* Corresponding author: Robert.Pellerin@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2012

1. Introduction

The RCPSP (Resource-Constrained Project Scheduling Problem) has been addressed in numerous papers (Brucker et al. 1999; Hartmann and Briskorn 2010). Introduced by Pritsker et al. (1969), the RCPSP consists in scheduling a set of activities linked by precedence relations in order to minimize project duration, while satisfying precedence and resource constraints. The RCPSP is known to be a NP-hard optimization problem (Blazewicz et al. 1983) and several heuristic methods have been proposed to treat this problem in different contexts (Neumann and Zhan 1995; Pellerin 1997; Hartmann 1999; Shue and Zamani 1999; Lee et al. 2003; Herroelen 2005; Kolisch and Hartmann 2006; Banaszak and Zaremba 2006; Hasgül et al. 2009; Oddi et al. 2010).

Among these methods, various acceleration techniques were developed with the aim of reducing the duration of project execution. These techniques include activity overlapping, compression and activity substitution (Gerk and Qassim 2008). They can be used either to produce an initial schedule, or to modify it in the course of execution. In practice, activity overlapping is a very wide-spread technique to accelerate engineering projects (Bogus et al. 2005a). This technique consists in executing in parallel two sequential activities by allowing a downstream activity to start before the end of an upstream activity based on preliminary information (Figure 1). However, the overlapping of activities can entail rework tasks and modifications further to the receipt of complementary information transmitted after the start of the downstream activity (Roemer et al. 2000). Activity overlapping thus allows reducing the total duration of project execution, at the expense of additional workload and execution cost associated to rework tasks.

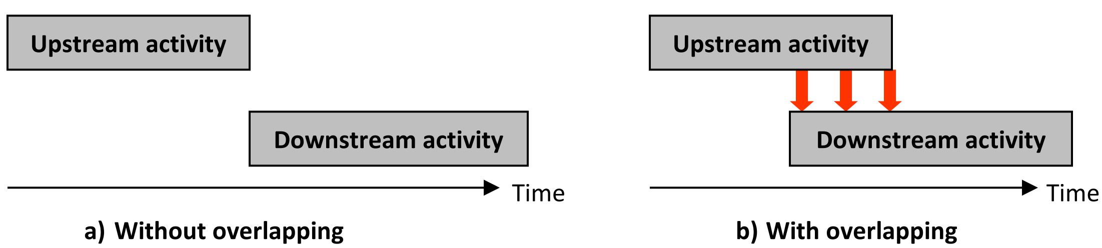

Figure 1. Upstream activity and downstream activity

Additional workload required to accommodate the information changes transmitted by upstream activities to the overlapped downstream activities are often ignored in practice. On the other hand, in spite of all research efforts accomplished in evaluating the relation between rework and the amount of overlap and determining the optimal overlapping strategy for two activities without resource constraints (Krishnan et al. 1997; Bogus et al. 2005b; Bogus et al. 2006; Lin et al. 2009; Lin et al. 2010; Loch and Terwiesch 1998; Roemer et al. 2000), only few papers have incorporated overlapping in the RCPSP (Cho and Eppinger 2005; Liberatore and Pollack-Jonhson 2006; Gerk and Qassim 2008). However, these papers studied simplified rework models, where the relation between overlap amount and rework is continuous and linear.

The objective of this paper is to extend the classical RCPSP to deal with realistic overlapping assumptions in order to partially fill this gap. To this end, we propose a project scheduling method that integrates activity overlapping as a project acceleration technique. We assume that the information flow is unidirectional from upstream to downstream activities. Consequently, the rework caused by execution of activities based on preliminary information is only assigned to the downstream activities of the identified overlappable activities. Information exchange is assumed to be costless and instantaneous. The main difference with the aforementioned overlapping models is that overlapping is restricted to a set of feasible overlap durations for each couple of overlappable activities, instead of considering a continuous and linear relation between overlap amount and rework. This assumption is more realistic considering that scheduling is performed in practice on a period-by-period basis (i.e., hour, day, week): resource availabilities and allocations are estimated per period, while activity durations are discrete multiples of one period (Hartmann 1999). The proposed method allows generating a schedule that takes into account resource constraints and overlapping possibilities. In order to evaluate the efficiency of the proposed approach, the proposed scheduling method was integrated within a common project scheduling software. A data generation process that enables the generation of various acceptable overlapping configurations was also developed to test it.

The remainder of this paper is organized as follows. Section 2 first presents a brief review of overlapping models. Section 3 describes the project scheduling problem with resource constraints and activity overlapping. Section 4 presents a heuristic method to solve the problem. An extension of the method allowing the study of the trade-off between acceleration and increase of project cost is also presented in Section 4. Section 5 presents the case studies generated and the results. The conclusion is presented in the last section.

2. Literature review

Overlapping models proposed in the literature can be classified into two categories. The first category includes models focusing on a couple of activities: an upstream activity of which depends the realization of a downstream activity. Krishnan et al. (1997) developed a planning model to find the best trade-off between overlapping and rework by means of two concepts: the evolution (slow or fast) of the upstream activity, which reflects the level of information that can be transmitted to the downstream activity, and downstream sensibility (low or high), which corresponds to the necessary time of work to palliate a change in the upstream activity. From these concepts, Krishnan et al. (1997) defined four overlapping strategies, each adapting to a specific evolution/sensibility scenario. For example, when the sensibility to changes is low and evolution speed is fast, Krishnan et al. (1997) suggested applying distributive overlapping which consists in starting the downstream activity with preliminary information, and then quickly integrating final information. Terwiesch and Loch (1999) statistically demonstrated the influence of the evolution of the upstream activity on the actual overlapping gain. Bogus et al. (2005a) improved the model developed by Krishnan et al. (1997) and identified eight overlapping strategies, as well as factors influencing the evolution and sensibility of activities. In two subsequent papers, Bogus et al. (2005b, 2006) proposed a basic decision algorithm to address overlapping of sequential activities without resource constraints. The heuristic is based on the selection of appropriate strategies, such as early freezing of design criteria, overdesign, and early release of preliminary information, to efficiently implement overlapping in practice. In the definition of the evolution of the upstream activity, some authors considered that the efforts of communication and coordination are "free" and immediate (Krishnan et al. 1997; Roemer et al. 2000). In that case, planning and organizing information exchanges between activities (e.g., plans, data of sizing, material orders) are not necessary. Other authors considered that communication involves cost and execution time (Loch and Terwiesch 1998; Lin et al. 2009).

Contrary to the first category of models geared to the interactions between two activities, which limits their application to small-sized projects, the second category includes more global models composed of several couples of overlappable activities and adapted to industrial projects. These models, which can be grouped into two classes, do not consider a couple of activities, but rather a project as a whole. The first class includes deterministic optimization models, where the relation between overlapping duration and rework duration is preliminary known for overlappable activities (Roemer et al. 2000; Lin et al. 2009). Most of these models use DSM (Design Structure Matrix), introduced by Steward (1981), to represent dependencies between activities, to minimize feedbacks, and to identify activity overlapping opportunities (Maheswari and Varghese 2005). Roemer et al. (2000) developed a heuristic method to find a trade-off between the duration and the cost of a project. Liberatore and Pollack-Johnson (2006) proposed a quadratic mixed integer programming model for crashing and overlapping without resource constraints. Also, Gerk and Qassim (2008) proposed a project acceleration model via activity crashing, overlapping and substitution. These three acceleration techniques are combined to find a schedule with minimal cost. The problem is formulated as a linear integer program. Thiagarasu and Devi (2009) also proposed a local search algorithm to reduce project duration by means of activity overlapping. The second class of models includes simulation models. These models are used when it is not possible to estimate the duration of rework required further to the overlapping of activities. For example, Cho and Eppinger (2005) developed a simulation model with activity and overlapping stochastic durations, and resource constraints. The model also takes into account rework and iterations (i.e. interactions between several activities requiring rework due to information feedbacks stemming from downstream activities). The authors showed that the resource constraints can delay the overlapping of certain activities and thus, the finish time of the project. Finally, Wang and Lin (2009) developed a stochastic overlapping model to take into account risks during the scheduling of activities. The model considers iterations and rework probabilities. However, the model does not take into account resource constraints. All these papers assume a simple linear relationship between rework and overlapping amount.

Table 1 summarizes the characteristics of overlapping models. Researches in the field of project planning largely ignore the constraints derived from the context of activity overlapping. Indeed, the studies on project acceleration techniques only rarely consider the constraints related to the limitation of available resources. Now, the management of resources has a crucial impact on the progress of projects. Furthermore, in the models studied, overlapping possibilities are often limited to the values of an interval. However, overlapping configurations are not quite acceptable in practice. In fact, every overlapping configuration should correspond to a precise execution mode of a couple of activities, with its associated information exchange.

Recently, Grèze et al. (2011) showed that the level of resource constraints, the number of overlapping opportunities and the maximum overlapping amount allowed have a significant impact on the efficiency of activity overlapping as a project acceleration technique. Furthermore, Berthaut et al. (2011) formulated the project scheduling problem with resource constraints and activity overlapping as a linear program with binary variables. The model is solved using CPLEX. Tests performed on a project involving 30 activities showed that an optimal solution can be reached within reasonable computation time. These experimental results are rather limited. In a subsequent paper, Berthaut et al. (2012) addressed the time-cost trade-off problem when communication and coordination delays are not negligible. However, no solution method is proposed.

Table 1. Characteristics of overlapping models

Overlapping models			Authors	Model characteristics
Coupling of two activities			Krishnan et al. (1997)	- Trade-off between overlapping and rework - No communication effort
			Loch et Terwiesch (1998)	- Communication cost and delay
			Terwiesch and Loch (1999)	- Statistical measure of overlapping efficiency
			Bogus et al. (2005a)	- Overlapping strategies
			Bogus et al. (2005b, 2006)	- Overlapping strategies - Trade-off between time savings and increased cost of rework
Industrial project application	Deterministic models	Heuristics	Roemer et al. (2000)	- No communication effort - Trade-off between project duration and cost
			Thiagarasu and Devi (2009)	- Resource constraints
		Exact methods	Liberatore and Pollack-Johnson (2006)	- Quadratic mixed integer program
			Gerk and Qassim (2008)	- Integer linear program - Min schedule cost
			Lin et al. (2009)	- Communication cost and delay
	Simulation models		Cho and Eppinger (2005)	- Activity and overlapping stochastic durations - Resource constraints - Rework and iterations
			Wang and Lin (2009)	- Stochastic model - Iterations - Rework probabilities

In summary, most contributions in the related literature fail to consider a realistic relationship between overlapping and rework in the RCPSP. A significant part of the literature in overlapping is dedicated to the determination of the optimal overlap amount for two activities without resource constraints (Krishnan et al. 1997; Loch and Terwiesch 1998; Terwiesch and Loch 1999; Roemer et al. 2000; Lin et al. 2009; Lin et al. 2010). An important finding of these papers is that the duration of rework is a convex increasing function of the amount of overlap. This statement is intuitive: if the amount of overlap increases, then the preliminary information at the downstream activity's start will be more unreliable and more downstream changes must be incorporated. The models proposed in the literature for a whole project with several overlappable couples of activities also consider a simplistic linear relation between the rework and the amount of overlap (Cho and Eppinger 2005; Liberatore and Pollack-Jonhson 2006; Gerk and Qassim 2008).

In order to study the interaction between overlapping and resource constraints in project scheduling with multiple activities and overlapping opportunities, the relation between rework and overlap amount is required for a range of overlap amounts for each couple of overlappable activities. Indeed, the optimal overlap amounts for a resource-constrained project composed of several couples of overlappable activities are not necessarily set to the optimal values found for each couple of activities (Browning and Eppinger 2002; Cho and Eppinger 2005; Gerk and Qassim 2008). The objective of this paper is to extend the classical RCPSP to deal with overlapping and additional workloads incurred by rework. The main difference with the aforementioned overlapping models is that overlapping is assumed to be defined for discrete values of overlap durations. This assumption is more realistic considering that scheduling is performed in practice on a period-by-period basis (i.e., hour, day, week): resource availabilities and allocations are estimated per period, while activity durations are discrete multiples of one period (Hartmann 1999).

3. Project scheduling with resource constraints and activity overlapping

The Resource-Constrained Project Scheduling Problem with Overlapping Modes (RCPSP-OM) is an extension of the classical RCPSP. Given a set of activities to execute, the RCPSP-OM consists in determining the order of execution in time of a set of activities so as to minimize the total project duration, while respecting precedence relations, resource constraints and overlapping possibilities. The RCPSP-OM is similar to the resource-constrained project scheduling problem with generalized precedence relations (RCPSP-GPR), but the proposed models do not consider reworks (De Reyck and Herroelen 1998; Bartusch et al. 1988). The RCPSP-OM also shares several characteristics with the traditional multi-mode resource-constrained scheduling problem (MRCPSP), as activities can be executed in several modes with different durations. However, considering the limit of exact solution procedures encountered with MRCPSP (Herroelen et al. 1998; Kolisch and Padman 2001; Hartmann and Briskorn 2010), we anticipate that solving the RCPSP-OM for real and large projects requires the use of metaheuristics or heuristic methods. In this paper, project scheduling with resource constraints and activity overlapping is based upon the following assumptions:

(1) *Information exchanges are unidirectional from upstream to downstream activities*.
Feedback information exchange from downstream activities leads to modifications performed by the upstream activities, and iterations can virtually occur (Wang and Lin 2009). In order to

eliminate feedbacks, DSMs (Design Structure Matrix) can be used to represent the relations between activities and to determine a feasible sequence of activities without any feedback, using a block triangularization algorithm (Browning 2001). As a last resort, activities can be aggregated or decomposed to eliminate feedbacks. We assume that such preliminary studies have been conducted. The project is then only composed of independent and dependent activities and information exchanges are unidirectional from upstream to downstream dependent activities.

(2) *Information exchanges between overlapped activities are free and immediate*.
Communication policies should be considered along overlapping if the information exchange between activities require non-negligible time and cost (Lin et al. 2009). We here assume that information exchanges are costless and instantaneous.

(3) *Overlappable activities are identified a priori*.
The dependent couples of activities can be categorized as non-overlappable or overlappable couples. The former case arises when a downstream activity requires the final output information from an upstream activity to be executed or its completion. Non-overlappable activities are connected with the classical finish-to-start precedence constraint. The latter case arises when a downstream activity can begin, with preliminary information, before an upstream activity is finished. Overlappable activities are connected with a finish-to-start-plus-lead time precedence constraint where the lead-time accounts for the amount of overlap.

(4) *For each couple of activities, overlapping possibilities are defined by modes (in a discrete way)*.
In practice, activity progress is measured according to the completion of internal milestones which corresponds to important events, such as design criteria frozen, detailed design completed, drawings finalized, or any activity deliverables. This preliminary information is issued at intermediate points and used as input for a downstream activity. Therefore, the start time of an overlapped downstream activity is restricted to a finite number of instants corresponding to upstream activities' milestones which constitutes different feasible modes for the execution of overlapping activities. These overlapping modes can also been seen as different overlapping strategies: no overlapping, conservative overlapping and aggressive overlapping, etc. Each overlapping mode is characterized by an amount of overlap expressed as a fraction of the downstream activity's duration and a rework duration.

(5) *For each overlapping mode, rework duration is estimated a priori* .
The main issue with the overlapping problem is to quantify the amount of rework as a function of the amount of overlap. A significant part of the literature in overlapping is dedicated to the determination of the optimal overlap amount for two activities without resource constraints. Indeed, the overlapping problem requires exploring the behavior and interaction of activities during their processes (Krishnan et al. 1997; Loch and Terwiesch 1998; Roemer et al. 2000). An important finding of these papers is that the duration of rework is a convex increasing function of the amount of overlap. In this paper, we suppose that the relation between the amount of overlap and the duration of rework is linear.

Figure 2 illustrates the various steps leading to the scheduling and acceleration of projects by means of activity overlapping. Steps 1,2, 6 and 7 are classical steps of project scheduling (Hegazy 2002). Steps 3,4 and 5 are specific to the project scheduling with overlapping modes and are linked to our hypothesis (3), (4) and (5).

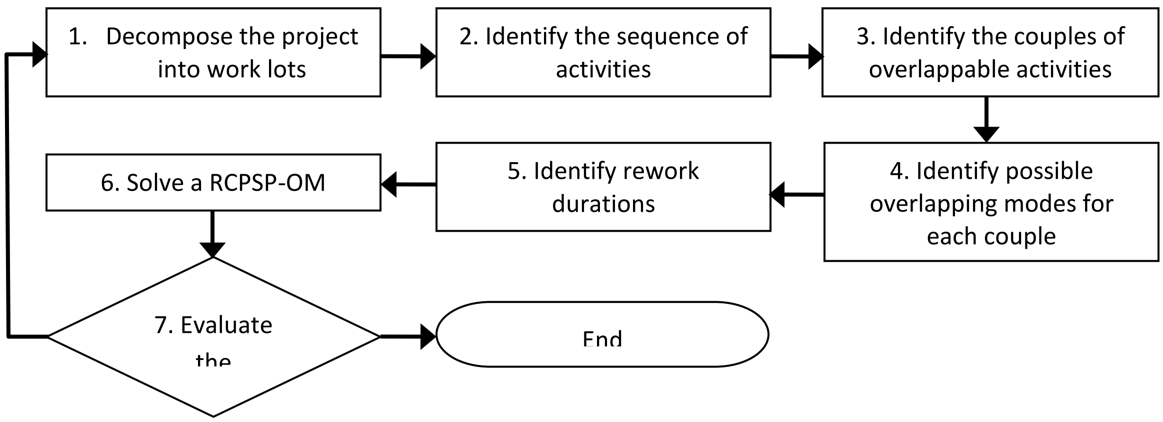

Figure 2. Process of project acceleration

In step 1, the project is decomposed into work lots. The size of lots depends on the level of detail required to manage the project. Work lots have to include similar activities of complexity and of duration, because the overlapping possibilities will be established from these activities. Decomposing a project into numerous short-term activities requires management and control efforts during the execution, but increases the chances to detect, among the couples of activities, overlapping opportunities during project planning. The decomposition of a project into work lots includes three stages (Hegazy 2002):

1) determination of the Work Breakdown Structure (WBS);
2) definition of the dependencies between the activities;
3) grouping of the work lots into independent working groups.

The first stage determines the WBS, which is a logical hierarchical decomposition of a project into different levels of detail, from a broad level (definable areas), down to a very detailed level (work lots), usually of reasonable and manageable size. A work lot may consist of one or more cost-significant activities and the size of a work lot can be measured in units, such as labour hours, budget or weight (Globerson, 1994). The optimal level of detail of the WBS depends on the nature and size of the project (Jung and Woo 2004). A fine decomposition allows to better control project expenses, but requires more managerial work throughout the execution of the project. There is not one unique construct of a WBS for a project. Therefore, its final design should be the result of a group effort of professionals from different functions and responsibilities. To help planners in performing the WBS and identifying the project activities, checklists based on past company records are often used. In order to identify the dependencies between the tasks in the second stage, the planning team needs to determine, for each activity, the activities that must be finished before the current one can start, the activities that may be constructed concurrently with the current one, and the activities that must follow the current one. The activity dependencies are defined by means of meetings with the persons in charge of the tasks and analyses of input data necessary for the execution of the activities. Finally, each work lot must be defined in such a way that only one organizational unit is responsible for its implementation (Globerson, 1994). This principle is however difficult to implement. In fact, even if all the activities in a given work lot are performed by one organizational unit, this unit may still not have complete control over the execution of the lot, since some activities belonging to that work lot are preceded by other activities which are not within the responsibility of that organizational unit. Some work lots thus need to be grouped into coherent working groups so as to reduce the degree of dependency between the working lots of a project. In the last stage, this integration can be achieved by various means, such as concurrent engineering, or other contents of the work lot (e.g., integration of two organizational units). The DSMs are also often used to group the dependent activities so as to better manage the information flow (Eppinger et al. 1994; Browning 2001; Chen and Li 2003; Fayez et al. 2003).

In project planning methods, projects are often represented as networks where the nodes correspond either to the activities (activity-on-node network) or the relations between the activities (activity-on-arc network). Although these networks describe the sequence of execution of the activities (step 2), they do not allow to represent the interactions between the activities or to model the information flows between the activities. Now, the study of interactions between activities is important in order to identify overlappable activities. These additional relations between activities can however be modeled by means of the decomposition DSMs (Browning 2001). Indeed, the DSMs aim at representing the information flows between the various activities, so allowing the search of information feedbacks. These feedbacks, derived from changes involved by the downstream activity, result in modifications and additional work at the upstream activity level. In order to avoid feedbacks and so obtain a sequence of activities where precedence relations and information flows circulate in the same direction, the DSMs are triangularized. The activities can also be aggregated or decomposed to eliminate feedbacks. In this article, we assume that preliminary studies have been conducted so as to identify the relations between activities and eliminate feedbacks. The projects considered are thus only composed of dependent and independent activities and the flow of information (between dependent activities) is unidirectional.

The third step of the process consists in identifying the couples of overlappable activities. The study of information exchanges between the activities by means of the DSMs and the historical data allows splitting the couples of activities into two categories: the overlappable couples of activities and the couples that are not overlappable. Two activities are overlappable if the downstream activity can start, with preliminary information, before the upstream activity ends, while receiving, during its execution, final information from the upstream activity. For the couples of activities with high sensibility, the changes of information transmitted by the upstream activity involve numerous rework tasks on the downstream activity. The associated gain is thus small (Krishnan 1996). Consequently, activity overlapping is generally allowed when sensibility is low.

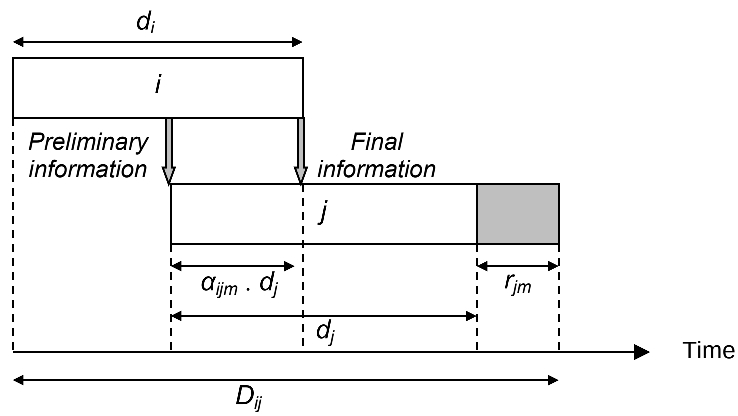

Figure 3. Overlapping process of two activities *i* and *j* in mode *m*

In steps 4 and 5, for every couple of activities, the possible overlapping modes and the associated rework durations are identified. Figure 3 represents the overlapping process of two overlappable activities i and j. Here, the downstream activity j starts with preliminary information transmitted by the upstream activity i. The amount of overlap associated with the overlapping mode m, noted α_{ijm}, corresponds to a whole fraction of the duration of the downstream activity j, noted d_j. So, we consider $\alpha_{ijm} \cdot d_j$ as an integer. The duration of rework associated with mode m for activity j, r_{jm}, is added to take into account the update of information transmitted during the execution of activity j. The total execution time of activities i and j for the overlapping mode m is defined as follows: $D_{ijm} = d_i + d_j(1\text{-}\alpha_{ijm}) + r_{jm}$.

In practice, project scheduling is established on the basis of discrete periods (hours, days, weeks, etc.). The overlapping percentages and the rework durations related to the modes are thus defined in a discrete way. Furthermore, the progress of activities is measured from the realization of milestones which correspond to major progresses or to realization of deliverables (specifications, plans, raw material orders, etc.) defined during the initial phase of the project. Preliminary information necessary for the start up of the downstream activity arises from the realization of these milestones. Every overlapping mode is thus characterized by a percentage of overlapping corresponding to a transmission of information by the upstream activity, and an amount of rework to fulfill in the downstream activity. On one hand, the values of α_{ijm} are determined in connection with the delivery of major goods of activity i so as to ensure that preliminary information used is approved. On the other hand, the definition of overlapping modes with relation to the milestones enables to have easily recourse to historical data to estimate rework times caused by activity overlapping. The problem of determining rework times with relation to overlapping times is treated in the literature (Krishnan et al. 1997; Lin et al. 2009).

Figure 4 illustrates four possible overlapping modes for two activities. The overlappable activities can be executed without overlapping (m = 1) or in accelerated regime (m > 1). In this article, we assume that there is no restriction concerning the number of overlappable predecessors for an activity. If an activity has several predecessors, then the associated mode corresponds to a combination of the various precedence relations. In that case, the total amount of rework to carry out corresponds to the sum of reworks associated with the precedence modes (overlapping modes).

Further to the evaluation of the overlapping parameters, the scheduling of the project is undertaken (step 6). In Section 4, we present a heuristic method for the scheduling of projects with resource constraints and overlapping modes. Once the scheduling is completed, the obtained schedule is measured against the initial schedule, in terms of cost and duration of execution (step 7). If an optimal solution is obtained (when the optimal value is known) or a solution for which the objective-function reaches a predetermined value, then the process stops. (In this study, we consider three objective-functions. See equations (5), (7) and (8) of Section 5.1). Otherwise, the process is reinitialized at the initial step by decomposing the activities more finely to detect new overlapping opportunities, so as to obtain a more compressed schedule.

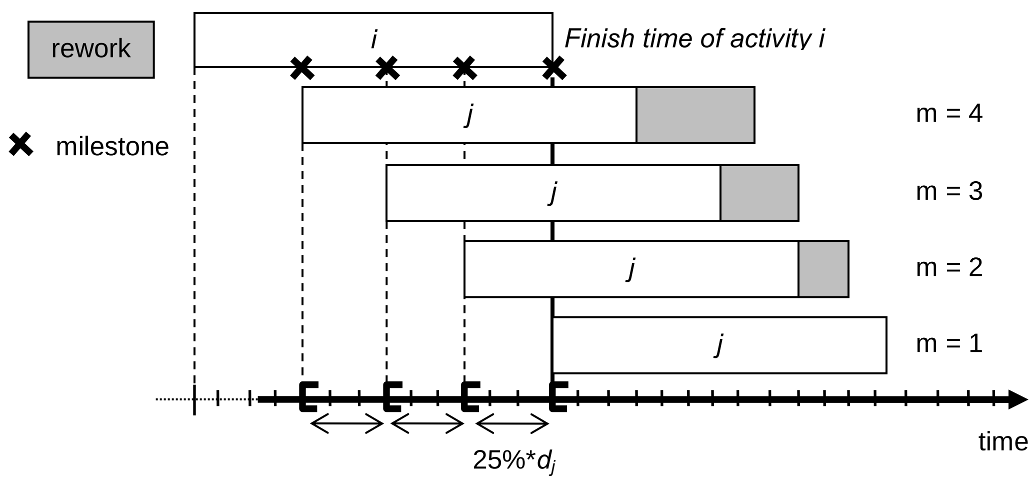

Figure 4. Possible execution modes for the couple of activities (*i, j*)

4. Heuristic method for project scheduling with activity overlapping

This section presents a heuristic method for the scheduling of projects with resource constraints and overlapping possibilities. This method is a constructive approach, i.e. which gradually builds a feasible solution while keeping an eye on solution cost, but it does not contain an improvement phase per se. This approach is commonly used in Time-Cost Tradeoff (TCT) analysis (Hegazy 2002). The proposed method corresponds to step 6 of the project acceleration process presented in Figure 2 (Section 3). An extension of the method is also proposed to consider the cost induced by the overlapping of activities.

4.1 Heuristic method

Figure 5 presents the heuristic method for the acceleration of projects via activity overlapping.

Step 1. Determine a schedule of the activities by solving a RCPSP. Let T be the so obtained scheduling plan. Let $D(T)$ be the duration of execution of plan T.

Step 2. Let A be the set of couples of overlappable activities and define $A' = \{j \in S : (i, j) \in A\}$ as the set of overlappable downstream activities. Let $m \in [1, m_i]$ be the overlapping mode of activity i, where m_i denotes the number of overlapping modes for activity i. For each overlappable activity $i \in A'$, set $m = 1$. Also, for each overlappable activity $i \in A'$, calculate the margin of activity i as follows: $M_i = LF_i - EF_i$, where EF_i and LF_i represent the earliest and latest possible finish times of activity i, respectively.

i) Order the overlappable activities according to the increasing order of the margins to obtain the list L.

ii) Let $\Sigma_{k \in R} R_{ik}$ be the total number of units of resources required by activity i per period. If certain overlappable activities have the same margin, order these activities according to the increasing order of the number of units of resources necessary for their realization.

iii) If certain overlappable activities require the same amount of resources, order these activities according to the decreasing number of successors.

Step 3. Choose the first activity i of the list L and consider the possibility to overlap activity i with the overlapping mode m.

Step 4. Determine a schedule of the activities by solving a RCPSP-OM. Let T' be the new so obtained scheduling plan. Let $D(T')$ be the duration of plan T'.

Step 5. If $D(T') < D(T)$, then T' becomes the new active plan. Set $T = T'$ and $m = m + 1$. If $m = m_i + 1$, remove activity i from the list L. If $D(T') \geq D(T)$, keep T as the active plan and remove activity i from the list L.

Step 6. If $L = \varnothing$, STOP. The plan T is the plan produced by the algorithm. Otherwise, return to step 3.

Figure 5. Heuristic for project acceleration via activity overlapping

This method is inspired by the methods for the compression of project duration (Gray et al. 2007). Indeed, these methods refer to the possibility of modifying the dependencies between activities such that the critical activities are executed in parallel (simultaneously) rather than in a sequential way. However, rework tasks are not considered. The method proposed in Figure 5 implies that rework tasks related to overlapping are added to downstream activities and that the consumption of the resources is constant throughout the execution of the project (including rework). The method also considers that the possible overlapping modes for every couple of activities and the duration of rework tasks associated with every mode are known in advance.

The method starts with the solution of a RCPSP by means of the management project software package MS Project 2007 to obtain an initial plan. The activities are then overlapped in an incremental way according to a priority order. For every overlappable activity considered, a new plan is obtained by starting earlier the downstream activity of the couple of overlappable activities considered (if the overlapping mode is greater than 0 and precedence constraints are verified for this activity) and, if necessary, by updating the earliest (and latest) possible finish times of each activity succeeding this one according to the constraints of a RCPSP-OM. The so obtained new plan *T'* is measured against the active plan *T*. If the duration of execution of the new plan is lower than the duration of the active plan, the new plan then becomes the active plan. Otherwise, the overlappable activity is removed from the list. The algorithm terminates after having considered all the overlapping possibilities.

In step 2 of the algorithm, overlappable activities are ordered according to three priority rules. Overlappable activities are first classified according to the increasing order of their margin. The margin M_i of activity i is the difference between the latest possible finish time LF_i and the earliest possible finish time EF_i of the activity. These times are determined a priori by means of a forward recursion and backward recursion algorithm that takes into account the starting up time of the project (time 0) and the maximal duration of the project (time T, i.e. the sum of processing times of all activities), while ignoring the resource constraints (Hartmann 1999). Overlappable activities of equal margin are then ordered according to the increasing order of the number of units of resources necessary for their realization. Indeed, the activities mobilizing few resources have less impact on the resource constraints. Finally, overlappable activities requiring the same quantity of resources are ordered according to the decreasing order of the number of successors. The ordered list of overlappable activities is updated every iteration. The algorithm was coded in Visual Basic linking Microsoft Excel and MS Project 2007. We also tested a modified version of the above heuristic by swapping the two priority rules ii) and iii) in step 2 (Figure 5). With respect to plan durations, there was no significant difference between the two procedures. However, the modified version increases the number of iterations necessary for obtaining a solution.

The heuristic is similar to the basic decision algorithm of Bogus et al. (2005b, 2006). The similarity arises from the fact that as in the aforementioned papers, a tradeoff is sought between time savings and increased cost of rework. However, there are important differences between these contributions and the present study. Bogus et al. (2005b, 2006) aimed to determine the optimal overlapping strategy for two activities without resource constraints. Furthermore, no experimentation is provided. In contrast, in the present study, we seek to study the interaction between overlapping and resource constraints in project scheduling with several overlappable couples of activities. This is achieved by the development of a constructive heuristic that integrates the scheduling of projects with resource constraints and overlapping possibilities. Computational experiments are presented in Section 5.

4.2 Trade-off between acceleration and increase of project cost

The objective-function used at step 5 of the algorithm (Figure 5) corresponds to the duration of the project. Now, in practice, acceleration through activity overlapping can appear, from a certain threshold, non efficient because of the important rework costs incurred. In this section, we propose to consider the trade-off between the decrease of the duration of a project via activity overlapping and the increase of the cost incurred by rework tasks by adapting the objective-function of the algorithm.

The search for the best trade-off between acceleration and increase of project cost involves the maximization of the gains associated with the execution of a project. The new evaluation function so corresponds to the gains associated with the substitution of the active plan *T* by the new plan *T'* and can be expressed as follows:

$$G = \quad * [D(T) - D(T\ )] + C(T) - C(T\ ) \qquad (1)$$

where is the opportunity cost corresponding to the bonuses (penalties) associated with an early (late) completion of a project. *C*(*T*) and *C*(*T'*) represent the costs of the rework tasks associated with the execution of the plans *T* and *T'*, respectively. The evaluation function (1) determines the gain associated with the decrease of the project duration from which we deduct the costs associated with rework tasks incurred by the activity overlapping strategy. If $G > 0$, then the plan *T'* becomes the active plan. Otherwise, the plan *T* is kept. We assume that gains and losses are linearly dependent of time.

5. Computational experiments

5.1 Experimental data

Figure 6 illustrates the generation process of the random instances. This process corresponds to the steps 3, 4 and 5 of the project acceleration process presented in Figure 2 (Section 3). Figure 6 indicates, for each step of the process, the sets and the input and output parameters.

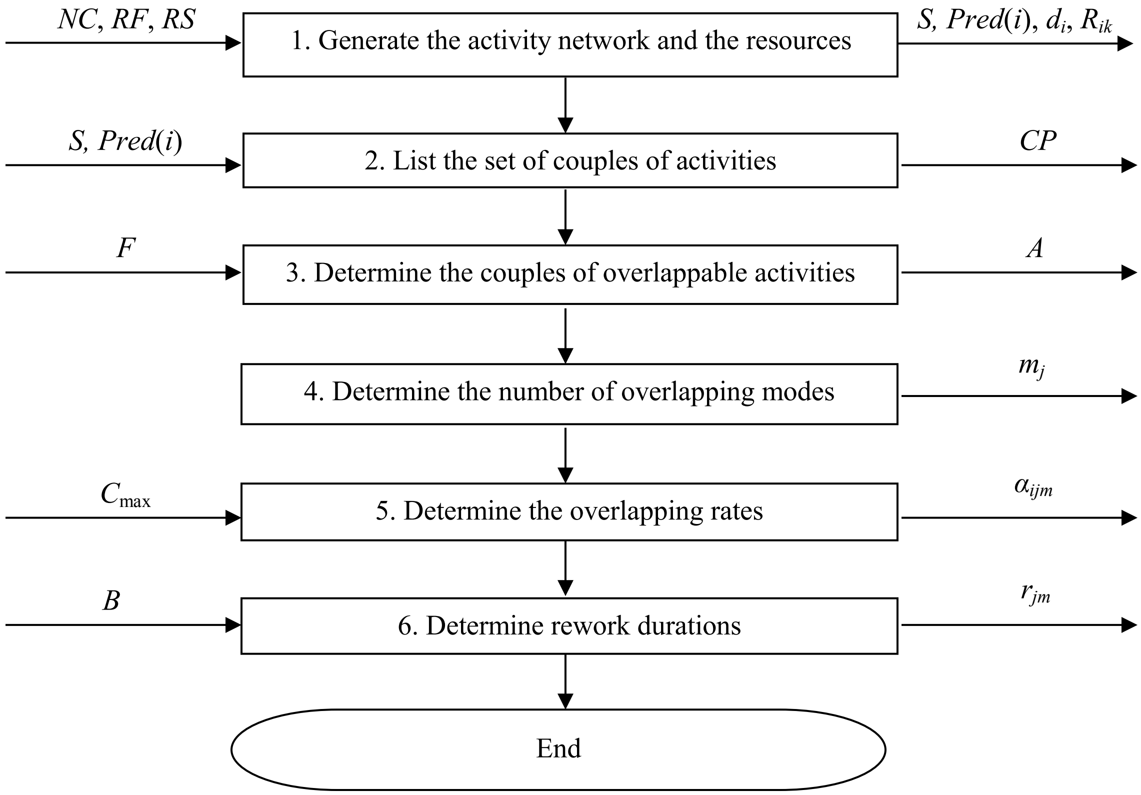

Figure 6. Data generation process

The PROGEN project generator, developed by Kolisch and Sprecher (1997), was used to generate three networks, each including 30 activities. Four types of resources are defined, each type representing a different competence. First, the network of activities (S), the precedence relations ($Pred(i)$), the durations of the activities (d_i) and the renewable resources (R_{ik}) are generated. The default parameters are kept, except for NC, RF and RS. The parameter NC (*Network Complexity*) represents the average number of relations between the activities. In the project library developed by Kolisch and Sprecher (1997), NC varies between 1.5 and 2.1. The parameter RF (*Resource Factor*) denotes, in percentage, the average number of required competencies per activity. $RF = 1$ when all four competencies are needed for the execution of each activity. $RF = 0.5$ when, on average, each activity uses two different competencies for its realization. For the tests, we set $NC = 1.8$ and $RF = 0.5$. The parameter RS (*Resource Strength*) defines the level of the resource constraints (Kolisch and Sprecher 1997). For every project, three levels of the resource constraints are considered: severe level ($RS = 0.5$), average level ($RS = 0.75$) and no resource constraints when the project is executed without overlapping ($RS = 1$). In this last case, the available resources are equal to the resource demand when executing the project without overlapping. For every competence k, the number of available resources is defined by the following equation:

$$R_k = Q_{k,\min} + Round\left(RS * (Q_{k,\max} - Q_{k,\min})\right) \quad (2)$$

where $Q_{k,\min}$ is equal to the minimal demand among the activities for resource k and $Q_{k,max}$ is equal to the maximal demand observed for resource k when the project is executed without considering the resource constraints. The parameter RS is the parameter having the most influence on the computation time for the solution of the RCPSP. The more RS is small, the more the problem is complex. Kolisch et al. (1995) showed that when RS varies from 1 to 0.2, the computation time increases by a factor of 1000.

Once the activity network and the renewable resources are generated, the necessary parameters for the consideration of the overlapping opportunities are defined: the set of couples of overlappable activities (A), the number of overlapping modes for every activity (m_j), the overlapping rate for every couple and every mode (α_{ijm}) and, for every activity, the rework duration associated with every mode (r_{jm}). The overlapping possibilities are randomly determined on the basis of three factors: F, $C_{\max}$ and B. The parameter F denotes the percentage of the couples of overlappable activities among the set of couples of activities CP. This parameter reflects the results of the analysis of relations and information flows between activities which allowed identifying the overlapping opportunities. The parameter $C_{\max}$ represents the maximum percentage of overlapping for a couple of activities. Finally, B denotes the percentage of necessary rework following activity overlapping. We suppose that the relation between the duration of overlapping and the duration of rework is linear. For the tests, we set $F = 40\%$, $C_{\max} = 75\%$ and $B = 40\%$. The value of B corresponds to the median value of the values used in the literature. The choice of the values of F and $C_{\max}$ is based on the study of Grèze et al. (2011). The couples of overlappable activities are randomly determined by means of the parameter F: $card(CP)*F = card(A)$. For the activities that are not overlappable, a single mode is practicable (m = 1, Figure 4). The overlapping rates of the activities that are not overlappable and the associated amounts of rework are equal to zero. For every couple of overlappable activities, four precedence modes are generated (Figure 4), corresponding to fractions of 0%, 25%, 50% and 75% of the duration of the downstream activity. This splitting is arbitrary. In practice, the precedence modes must correspond to the various milestones delivered by the upstream activity (see Section 3). So, for every couple of overlappable activities, the overlapping percentages and the rework durations associated to the modes can be determined by equations (3) and (4), respectively:

$$\alpha_{ijm} = \frac{C_{\max} * (m-1)}{3} \qquad \forall (i,j) \in A, \forall m \in [1, m_j] \tag{3}$$

$$r_{jm} = \sum_{i \in Pred(j)} round(B * \alpha_{ijm} * d_i) \qquad \forall j \in S : (i,j) \in A, \forall m \in [1, m_j] \tag{4}$$

Recall that when an activity overlaps several predecessors, the total duration of rework is then equal to the sum of rework durations associated with the overlapping with each predecessor. If an activity overlaps only a single predecessor, then the total duration of rework is equal to the rework duration associated with the overlapping with this predecessor.

In order to evaluate the quality of the solutions produced by the heuristic, we compared the deviations with respect to the optimal solutions obtained by solving the scheduling model with resource constraints and overlapping possibilities proposed by Berthaut et al. (2011). Let X_{jtm} be a binary variable equal to 1 if and only if activity j is executed in mode m and finishes at time t. Let n be the total number of activities. The fictitious activities 0 and $n + 1$, with zero processing time, correspond to the project start and project end, respectively. The objective-function (5), of the model of Berthaut et al. (2011) for the RCPSP-OM, minimizes the total duration of the project:

Minimize $$D = \sum_{t=EF_{n+1}}^{LF_{n+1}} t \cdot X_{n+1,t,1} \qquad (5)$$

The total cost induced by overlapping is equal to the sum of the additional coordination costs incurred by the overlapping of activities and the additional costs associated with rework tasks. The execution of activities in parallel requires a supplementary effort of coordination and communication between the resources. In this article, we suppose that the duration of coordination meetings and the cost-in-use of communication technologies are negligible. Consequently, coordination and communication costs are equal to zero. However, we consider that the resources involved in rework tasks added to an activity correspond, in quantity and in qualification, to the resources used in the realization of the activity. The cost of rework is thus equal to the cost-in-use of the resources for the additional work:

$$C = \sum_{j \in S} \sum_{t=EF_j}^{LF_j} \sum_{m=1}^{m_j} \sum_{k \in R} X_{jtm} * r_{jm} * R_{jk} * \theta_k \qquad (6)$$

where R represents the set of resources. The parameters R_{jk} and θ_k denote the number of units of resource k required by period for the execution of activity j, and the hourly cost of resource k, respectively. In order to take into account the cost associated with activity overlapping, the objective-function (5) is modified as follows:

Minimize $$P = \sum_{t=EF_{n+1}}^{LF_{n+1}} t \cdot X_{n+1,t,1} + \delta * C \qquad (7)$$

where δ is chosen such that $\delta * C < 1$. The objective-function (7) minimizes first the duration of the project, then the cost of rework tasks. Also, to look for the best trade-off between decrease of the duration of a project and increase of the cost of realization, the objective-function (5) is modified as follows:

Maximize $$G = \quad * (T_{ref} - \sum_{t=EF_{n+1}}^{LF_{n+1}} t \cdot X_{n+1,t,1}) - C \quad (8)$$

where T_{ref} is the reference time obtained when executing the project without overlapping. The objective-function (8) maximizes the real additional gain obtained when executing the project in accelerated regime. The opportunity cost is fixed to the value of $1000 per period. This cost models the bonuses (penalties) related to an advance (delay) in the delivery of the project. We assume that the bonuses (penalties) are proportional to the time saved (lost). In practice, the values of the bonuses (penalties) are established according to the terms of the contract between the firm and the customer.

All models were programmed using AMPL Studio v1.6.j running with CPLEX 12.2 (after tuning the parameters). All experiments were performed on a personal computer (2.22GHz and 3.00Go of RAM).

5.2 Analysis of results

For the computational experiments, four scenarios are studied. In the first scenario, the overlapping possibilities are forbidden and the objective consists in minimizing the execution duration of the project. The second scenario also permits to obtain a minimal duration project scheduling, but the overlapping possibilities are now allowed. In the third scenario, the costs of rework tasks induced by activity overlapping are taken into account in a hierarchical objective that minimizes first the execution duration of the project, then the rework cost. Finally, the fourth scenario considers the trade-off between the decrease of the duration of the project and the increase of the costs incurred by the overlapping of activities. We note that scenarios 2 and 3 are equivalent for the heuristic method because this method is a constructive approach, i.e. which builds the solution of minimal duration one element at each stage, without ever questioning the past choices. Each scenario is thus associated with a problem to solve and an objective to optimize:

- Scenario 1: RCPSP, Min *D* (Section 5.1, equation (5));
- Scenario 2: RCPSP-OM, Min *D* (Section 5.1, equation (5));
- Scenario 3: RCPSP-OM, Min *P* (Section 5.1, equation (7));
- Scenario 4: RCPSP-OM, Max *G* (Section 5.1, equation (8)).

Table 2 presents the durations of execution (*D*) and the rework costs (*C*) associated with the solutions obtained by means of the exact method (OPT), the MS Project tool (MSP) and the heuristic method (H) for the various scenarios. For example, C_{OPT3} corresponds to the rework cost of the solution produced by the exact method when considering the objective-function (7).

Table 2. Computational comparison of the four scenarios

Project	RS	Scenario 1		Scenario 2		Scenario 3				Scenario 4			
		D_{OPT1}	D_{MSP}	D_{OPT2}	C_{OPT2}	D_{OPT3}	C_{OPT3}	D_{H3}	C_{H3}	D_{OPT4}	C_{OPT4}	D_{H4}	C_{H4}
1	1	121	121	104	15 300	104	12 300	107	4 100	107	4 100	107	4 100
	0.75	121	125	104	15 300	104	13 100	107	4 100	107	4 100	107	4 100
	0.5	121	121	107	8 500	107	6 300	111	3 700	110	2 700	111	3 700
2	1	98	98	83	21 200	83	10 800	89	6 600	85	6 400	91	3 000
	0.75	98	100	85	15 000	85	9 200	92	4 000	85	6 800	92	4 000
	0.5	106	127	99	12 200	99	10 200	105	6 800	105	300	110	2 700
3	1	112	112	95	20 500	95	11 400	104	4 800	98	6 900	105	1 400
	0.75	112	121	98	14 700	98	9 900	102	11 100	100	6 500	107	5 600
	0.5	112	128	107	11 000	107	6 700	118	3 600	112	0	114	3 900

We note that the optimal execution durations are the same for the scenarios 2 and 3. Indeed, for Scenario 3, the exact method finds, among the solutions of minimal duration (Scenario 2), the minimum cost solution. The results of Table 2 reveal that the overlapping of activities (Scenarios 2, 3 and 4) enables to reduce the execution duration of the projects at the expense of an increase of the execution cost caused by reworks. However, the consideration of the costs related to the overlapping of activities in the evaluation function (Scenarios 3 and 4) allows limiting additional rework.

Table 3 compares the values of profits (or losses) associated with Scenarios 3 and 4. These values are calculated according to equation (7). For each method, profits stemming from the trade-off between acceleration and increase of the costs related to the project (Scenario 4) are more important than profits associated with Scenario 3 in most cases. Furthermore, for the exact method, Scenario 4 allows avoiding losses incurred by the project acceleration through activity overlapping. However, the increase of profits associated with Scenario 4 is obtained in consideration of an increase of project execution durations. Indeed, for Scenario 4, the execution durations generated by the exact method and the heuristic method are often longer than those of Scenario 3 (see Table 2). In some cases, profits are more important with the heuristic method thant with the exact method. This is due to the fact that the duration of the project scheduled without overlapping mode (D_{msp}) is longer with the heuristic, which can lead to a bigger benefits with overlapping modes.

Table 3. Comparison of the profits ($) associated with Scenarios 3 and 4

Project	RS	Exact method		Heuristic	
		Scenario 3	Scenario 4	Scenario 3	Scenario 4
1	1	4 700	9 900	9 900	9 900
	0.75	3 900	9 900	13 900	13 900
	0.5	7 700	8 300	6 300	6 300
2	1	4 200	6 600	2 400	4 000
	0.75	3 800	6 200	4 000	4 000
	0.5	-3 200	700	15 200	14 300
3	1	5 600	7 100	3 200	5 600
	0.75	4 100	5 500	7 900	8 400
	0.5	-1 700	0	6 400	10 100

Table 4 presents, for the exact method, the obtained deviations (%) when overlapping possibilities are allowed (Scenarios 2, 3 and 4). In the column 'Scenario 2', we observe that the impact of overlapping on project execution duration decreases when the level of the resource constraints is severe. For example, the reduction of the execution duration of project 3 through activity overlapping varies from 15.18% when RS = 1 to 4.46% when RS = 0.5. The column 'Scenario 3' gives the cost reductions associated with the consideration of the rework cost in the objective-function. In all cases, the costs of the optimal solutions are reduced in comparison with the costs of the solutions found with Scenario 2. The taking into consideration of the costs in the objective-function (Scenario 3) allows limiting unnecessary overlapping of activities, and thus the addition of supplementary rework tasks. However, project acceleration through activity overlapping can incur losses (see Table 3). Scenario 4 permits to avoid this possibility by considering the trade-off between the acceleration of projects and the increase of associated costs. The column 'Scenario 4' shows that an increase of the durations in comparison with Scenario 3 allows reducing rework costs.

Table 4. Comparison of the successive improvements of scenarios 2, 3 and 4

Project	RS	Scenario 2	Scenario 3	Scenario 4	
		$(D_{OPT2}-D_{OPT1})/D_{OPT1}$	$(C_{OPT3}-C_{OPT2})/C_{OPT2}$	$(D_{OPT4}-D_{OPT3})/D_{OPT3}$	$(C_{OPT4}-C_{OPT3})/C_{OPT3}$
1	1	-14.05	-19.61	2.88	-66.67
	0.75	-14.05	-14.38	2.88	-68.70
	0.5	-11.57	-25.88	2.80	-57.14
2	1	-15.31	-49.06	2.41	-40.74
	0.75	-13.27	-38.67	0.00	-26.09
	0.5	-6.60	-16.39	6.06	-97.06
3	1	-15.18	-44.39	3.16	-39.47
	0.75	-12.50	-32.65	2.04	-34.34
	0.5	-4.46	-39.09	4.67	-100.00

Table 5 presents the computation times for the solution of the models associated with Scenarios 2, 3 and 4. We observe that the computation times depend on the network of activities considered, on the level of the resource constraints and on the scenario studied. In particular, computation times for the solution of the minimum duration model with consideration of the costs (Scenario 3) are higher than times required for solving the basic model (Scenario 2). In fact, solving the model related to Scenario 3 requires the exploration of all the solutions of minimal duration to find the solution of minimal cost. Furthermore, the search for the best trade-off between reduction of the duration and increase of the costs (Scenario 4) is faster than the search for optimal solutions for the models of Scenarios 2 and 3.

Table 5. Computation times (seconds) for the exact method

Project	RS	Scenario 2	Scenario 3	Scenario 4
1	1	8	7	7
	0.75	22	14	12
	0.5	60	> 5000	20
2	1	12	84	8
	0.75	26	86	20
	0.5	> 5000	> 5000	2110
3	1	9	4	7
	0.75	13	99	13
	0.5	120	> 5000	20

Table 6 presents an evaluation of the heuristic methods. We first observe that, for the MS Project scheduling tool, the deviations (%) with respect to the optimal execution durations (column (1)) are more significant when the level of resource constraints is severe. This result, comparable to the results obtained in the literature (Kolisch 1999), can be explained by the poor performance of the MS Project tool which uses simple heuristic rules allowing to quickly elaborate a project schedule by taking into account given constraints.

Table 6. Comparison of the heuristics MSP, H3 and H4

Project	RS	MSP	H3		H4			
		(1)	(2)	(3)	(4)	(5)	(6)	(7)
		$(D_{MSP}-D_{OPT1})/D_{OPT1}$	$(D_{H3}-D_{OPT3})/D_{OPT3}$	$(D_{H3}-D_{MSP})/D_{MSP}$	$(D_{H4}-D_{OPT4})/D_{OPT4}$	$(D_{H4}-D_{MSP})/D_{MSP}$	$(D_{H4}-D_{H3})/D_{H3}$	$(C_{H4}-C_{H3})/C_{H3}$
1	1	0.00	2.88	-11.57	0.00	-11.57	0.00	0.00
	0.75	3.31	2.88	-14.40	0.00	-14.40	0.00	0.00
	0.5	0.00	3.74	-8.26	0.91	-8.26	0.00	0.00
2	1	0.00	7.23	-9.18	7.06	-7.14	2.25	-54.55
	0.75	2.04	8.24	-8.00	8.24	-8.00	0.00	0.00
	0.5	19.81	6.06	-17.32	4.76	-13.39	4.76	-60.29
3	1	0.00	9.47	-7.14	7.14	-6.25	0.96	-70.83
	0.75	8.04	4.08	-15.70	7.00	-11.57	4.90	-49.55
	0.5	14.29	10.28	-7.81	1.79	-10.94	-3.39	8.33

The deviations in column (2) show that the quality of the solutions obtained by using the heuristic method to generate an accelerated schedule taking into account the overlapping modes and the associated rework costs (Scenario 3) is variable. Indeed, for Scenario 3, the use of the heuristic method induces deviations within 2.88 % and 10.28 % from the optimal solutions. We also notice that the use of the heuristic method to generate a schedule with overlapping modes and associated costs (Scenario 3) allows, in certain cases, to compensate the poor performance of the MS Project tool. For example, for project 2 with RS = 0.5, the deviation observed in comparison with the optimal solution is 19.81% for the tool and 6.06% for the heuristic method. Furthermore, the deviations in columns (3) and (5) show that the heuristic method enables to significantly reduce the durations of the projects in comparison with the durations obtained with the MS Project tool. Contrary to column (2), column (4) shows that the use of the heuristic method to generate a schedule with overlapping modes, associated costs and penalties related to the execution of the project (Scenario 4) allows, in certain cases, to generate schedules with near-optimal durations. Finally, columns (6) and (7) show that an increase in project durations, in comparison with Scenario 3, enables to reduce the rework costs.

Table 7 gives the number of iterations for obtaining a good solution by means of the heuristic method for Scenario 3. Every iteration corresponds to a project schedule. We note that the number of iterations is rather low and variable.

Table 7. Number of iterations for the heuristic method (Scenario 3)

		Network		
		1	2	3
RS	1	13	67	94
	0.75	14	5	135
	0.5	17	22	3

6. Conclusion

In this article, we proposed a heuristic method for the RCPSP-OM. In order to evaluate the quality of the solutions generated by the heuristic, we calculated the deviations in comparison with the optimal solutions obtained by solving the model proposed by Berthaut et al. (2011). When the objective consists in minimizing the project duration, the results show that the consideration of the costs associated to the overlapping of activities allows to significantly reducing the cost of reworks. On the other hand, when the objective consists in maximizing the gains related to the project execution, the search for the best trade-off between acceleration and increase of the costs of the project enables to avoid losses.

The exact solution of the model of Berthaut et al. (2011) by means of a linear integer programming solver (such as CPLEX) is inconvenient to apply to large-sized projects because of the fast increase in computation times. The heuristic method developed in this article is a constructive approach which, in certain cases, allows producing good solutions in reasonable computation times. However, the quality of the solutions is variable and should be confirmed by an extended statistical study to confirm quality of this heuristic. However, it confirms that heuristic can be a good solution to implement overlapping modes in an industrial context with computational time constraints.

Furthermore, some assumptions defined in Section 3 may appear too reducing. For example, we assume that, for each couple of overlappable activities and for each overlapping mode, the total rework duration and cost are constant and preliminary known. However, in a practical industrial context, these parameters are difficult to estimate. Real projects also include numerous information feedbacks and interdependent engineering activities, and information exchanges between overlapped activities require non-negligible time and cost (Loch and Terwiesch 1998; Lin et al. 2010) in practice. We thus see the development of more powerful and realistic project scheduling methods, as tabu search, genetic algorithms or stochastic approaches, taking into consideration the characteristics of industrial applications arising from practice. Stochastic approaches would allow to take into account the foreseeable variations between estimates and actual values.

Acknowledgments

This work has been supported by the Natural Sciences and Engineering Research Council of Canada and the Jarislowsky/SNC-Lavalin Research Chair in the Management of International Projects . This support is gratefully acknowledged.

References

Banaszak, Z.A., & Zaremba, M.B. (2006). Project-driven planning and scheduling support for virtual manufacturing. Journal of Intelligent Manufacturing, 17(6), 641-651.

Bartusch, M., Möhring, R.H., & Radermacher, F.J. (1988). Scheduling project networks with resource constraints and time windows. Annals of Operations Research, 16(1), 199–240.

Berthaut, F., Grèze, L., Pellerin, R., Perrier, N., & Hajji, A. (2011). Optimal resource-constrained project scheduling with overlapping modes. Proceedings of International Conference on Industrial Engineering and Systems Management. Metz, France: International Institute for Innovation, Industrial Engineering and Entrepreneurship, 299-308.

Berthaut, F., Pellerin, R., Perrier, N., & Hajji, A. (2012). Time-cost trade-offs in resource-constrained project scheduling problems with overlapping modes. To appear in International Journal of Project Organisation and Management.

Blazewicz, J., Lenstra, J. K., & Rinnooy Kan, A. H. G. (1983). Scheduling subject to resource constraints: classification and complexity. Discrete Applied Mathematics, 5(1), 11−24.

Bogus, S. M., Molenaar, K. R., & Diekmann, J. E. (2005a). Concurrent engineering approach to reducing design delivery time. Journal of Construction Engineering and Management, 131(11), 1179−1185.

Bogus, S. M., Diekmann, J. E., & Molenaar, K. R. (2005b). Evaluating the consequences of overlapping dependent activities. Construction Research Congress 2005, No 7539, 1−10.

Bogus, S. M., Molenaar, K. R., & Diekmann, J. E. (2006). Strategies for overlapping dependent design activities, Construction Management and Economics, 24, 829–837.

Browning, T. R. (2001). Applying the design structure matrix to system decomposition and integration problems: A review and new directions. IEEE Transactions on Engineering Management, 48(3), 292−306.

Browning, T. R., & Eppinger, S. D. (2002). Modeling impacts of process architecture on cost and schedule risk in product development. IEEE Transactions on Engineering Management, 49(4), 428−442.

Brucker, P., Drexl, A., Möhring, R., Neumann, K., & Pesch, E. (1999). Resource-constrained project scheduling: Notation, classification, models, and methods. European Journal of Operational Research, 112(1), 3−41.

Chen, S. J., & Li, L. (2003). Decomposition of interdependent task group for concurrent engineering. Computers & Industrial Engineering, 44(3), 435−459.

Cho, S. H., & Eppinger, S. D. (2005). A simulation-based process model for managing complex design projects. IEEE Transactions on Engineering Management, 52(3), 316–328.

De Reyck, B., & Herroelen, W. (1998). Branch-and-bound procedure for the resource-constrained project scheduling problem with generalized precedence relations. European Journal of Operational research, 111 (1),152−174.

Eppinger, S. D., Whitney, D. E., Smith, R. P., & Gebala, D. A. (1994). A Model-Based Method for Organizing Tasks in Product Development. Research in Engineering Design-Theory Applications and Concurrent Engineering, 6(1), 1−13.

Fayez, M., Axelsson, P., Oloufa, A. A., & Hosni, Y. (2003). DSM Versus CPM: Issues for Planning Design & Construction Activities. Proceedings of the Construction Research Congress, Winds of Change: Integration and Innovation in Construction. Honolulu, HI: American Society of Civil Engineers, 181−188.

Gerk, J. E. V., & Qassim, R. Y. (2008). Project Acceleration via Activity Crashing, Overlapping, and Substitution. IEEE Transactions on Engineering Management, 55(4), 590–601.

Globerson, S. (1994). Impact of various work-breakdown structures on project conceptualization. International Journal of Project Management, 12(3), 165-171.

Gray, C. F., Langevin, Y., & Larson, E. W. (2007). Management de projet, Montréal: Chenelière McGraw-Hill.

Grèze, L., Pellerin, R., Leclaire, P., & Perrier, N. (2011). Evaluating the effectiveness of task overlapping as a risk response strategy in engineering projects. To appear in International Journal of Project Organisation and Management.

Hartmann, S. (1999). Project scheduling under limited resources. Berlin:Springer.

Hartmann, S., & Briskorn, D. (2010). A survey of variants and extensions of the resource-constrained project scheduling problem. European Journal of Operational Research, 207(1), 1−14.

Hasgül, S., Saricicek, I., Ozkan, M., & Parlaktuna, O. (2009). Project-oriented task scheduling for mobile robot team. Journal of Intelligent Manufacturing, 20(2), 151-158.

Hegazy, T. (2002). Computer-based construction project management. Upper Saddle River, NJ: Prentice Hall.

Herroelen, W., De Reyck, B., & Demeulemeester, E. (1998). Resource constrained project scheduling – A survey of recent developments. Computers and Operations Research, 25(4), 279–302.

Herroelen, W. (2005). Project scheduling - Theory and practice. Production and Operations Management, 14(4), 413–432.

Jung, Y., & Woo, S. (2004). Flexible work breakdown structure for integrated cost and schedule control. Journal of Construction Engineering and Management-Asce 130(5), 616–625.

Kolisch, R. (1999). Resource allocation capabilities of commercial project management software packages. Interfaces, 29(4), 19–31.

Kolisch, R., & Hartmann, S. (2006). Experimental investigation of heuristics for resource-constrained project scheduling: An update. European Journal of Operational Research, 174(1), 23−37.

Kolisch, R., & Sprecher, A. (1997). PSPLIB - A project scheduling problem library. European Journal of Operational Research, 96(1), 205−216.

Kolisch, R., Sprecher, A., & Drexl, A. (1995). Characterization and generation of a general class of resource-constrained project scheduling problems. Management Science, 41(10), 1693−1703.

Kolisch, R., & Padman, R. (2001). An integrated survey of deterministic project scheduling. Omega, 29(3), 249–272.

Krishnan, V. (1996). Managing the simultaneous execution of coupled phases in concurrent product development. IEEE Transactions on Engineering Management, 43(2), 210–217.

Krishnan, V., Eppinger, S. D., & Whitney, D. E. (1997). A model-based framework to overlap product development activities. Management Science, 43(4), 437–451.

Lee, Y.-H., Kumara, S.R.T., & Chatterjee, K. (2003). Multiagent based dynamic resource scheduling for distributed multiple projects using a market mechanism. Journal of Intelligent Manufacturing, 14(5), 471-484.

Liberatore, M.J., & Pollack-Johnson, B. (2006). Extending project time–cost analysis by removing precedence relationships and task streaming. International Journal of Project Management, 24(6), 529–535.

Lin, J., Chai, K. H., Brombacher, A. C., & Wong, Y. S. (2009). Optimal overlapping and functional interaction in product development. European Journal of Operational Research, 196(3), 1158–1169.

Lin, J., Qian, Y., Cui, W., & Miao, Z. (2010). Overlapping and communication policies in product development. European Journal of Operational Research, 201(3), 737–750.

Loch, C. H. & Terwiesch, C. (1998). Communication and uncertainty in concurrent engineering. Management Science, 44(8), 1032–1048.

Maheswari, J. U., & Varghese, K. (2005). Project Scheduling using Dependency Structure Matrix. International Journal of Project Management, 23(3), 223–230.

Neumann, K., & Zhan, J. (1995). Heuristics for the minimum project-duration problem with minimal and maximal time lags under fixed resource constraints. Journal of Intelligent Manufacturing, 6(2), 145–154.

Oddi, A., Cesta, A., Policella, N, & Smith, S.F. (2010). Iterative flattening search for resource constrained scheduling. Journal of Intelligent Manufacturing, 21(1), 17-30.

Pellerin, R. (1997). Modèle d'ordonnancement dynamique de projets de réfection. Ph.D. Thesis, Montréal, École Polytechnique de Montréal.

Pritsker, A. A. B., Watters, L. J., & Wolfe, P. M. (1969). Multi-project scheduling with limited resources: a zero-one programming approach. Management Science, 16(1), 93–108.

Roemer, T. A., Ahmadi, R., & Wang, R. H. (2000). Time-cost trade-offs in overlapped product development. Operations Research, 48(6), 858–865.

Shue, L.-Y., & Zamani, R. (1999). Intelligent search method for project scheduling problems. Journal of Intelligent Manufacturing, 10(3), 279–288.

Steward, D. V. (1981). The Design Structure System: A Method for Managing the Design of Complex Systems. IEEE Transactions on Engineering Management, 28(3), 71–74.

Terwiesch, C., & Loch, C. H. (1999). Measuring the effectiveness of overlapping development activities. Management Science, 45(4), 455–465.

Thiagarasu, V. & Devi, T. (2009) Multi-agent coordination in project scheduling: priority rules based resource allocation. International Journal of Recent Trends in Engineering, 1(2), 42–46.

Wang, J. T. & Lin, Y. I. (2009). An overlapping process model to assess schedule risk for new product development. Computers & Industrial Engineering, 57(2), 460–474.