THE IMPACT OF PROJECT CHARACTERISTICS ON THE EFFICIENCY OF ACTIVITY OVERLAPPING IN PROJECT SCHEDULING

FRANCOIS BERTHAUT
ROBERT PELLERIN
NATHALIE PERRIER
ÉCOLE POLYTECHNIQUE DE MONTRÉAL, CANADA

ADNÈNE HAJJI
LAVAL UNIVERSITY, CANADA

KEYWORDS: ACTIVITY OVERLAPPING, PROJECT MANAGEMENT, PROJECT SCHEDULING; LINEAR PROGRAMMING; SCATTER SEARCH; TWO-PART MODEL.

Abstract: This paper tackles the project scheduling problem in presence of complex networks of activities, resource constraints, overlapping and rework. The objective is to analyse the influence of project characteristics, such as project size, resource constraints, overlapping opportunities and rework, on the efficiency of overlapping in terms of reduction of the project makespan. An exact solution procedure and a metaheuristic are thus proposed to minimize the project makespan, while limiting the use of overlapping. A two-part model is used to conduct a statistical analysis of the influence of project characteristics on the makespan gain with overlapping. Results suggest that the best overlapping decision should consist in overlapping few pairs of overlappable activities with a large degree of overlapping. Furthermore, for complex projects, overlapping decisions should not rely solely on the criticality of the activities. These findings provide a better understanding of overlapping decisions and should guide planners in improving existing practices.

1. INTRODUCTION

Overlapping consists in relaxing the sequential execution of dependent activities by allowing downstream activities to begin before receiving all the final information required from upstream activities. Several strategies developed to accelerate project execution, such as concurrent engineering (Terwiesch and Loch, 1999) and fast-tracking (Dzeng, 2006 and Pena-Mora and Li, 2001), are based on the concept of overlapping. However, overlapping often causes additional reworks in downstream activities, as well as iterations of interdependent activities. Such reworks may outweigh the benefices of overlapping in terms of cost and time (Terwiesch and Loch, 1999). This raises the question of when and to which extent overlapping should be applied. Two groups of models have been developed in the literature to investigate this question. First, many authors have considered only one or few pairs of overlappable activities and no resource constraints. Such models were described, for example, by Krishnan et al. (1997), Roemer et al. (2000), Roemer and Ahmadi (2004), Khoueiry et al. (2013), Loch and Terwiesch (1998), Lin et al. (2010) and Tyagi et al. (2013). Other approaches, such as those proposed by Browning and Eppinger (2002), Wang and Lin (2009), Lim et al. (2014), Cho and Eppinger (2005) and Huang and Chen (2006), have considered overlapping in projects with complex networks of activities using scheduling techniques. These approaches can be distinguished according to the deterministic or stochastic nature of the problem and whether resource constraints, reworks and iterations are considered.

The standard deterministic Resource-Constrained Project Scheduling Problem (RCPSP) has also been extended to model overlapping without feedbacks and iterations. The objective is not only to find the final date of the project, but also the execution dates of the activities that will be used as baseline schedule. Such extensions were described, for example, by Gerk and Qassim (2008), Berthaut et al. (2014), Bartusch et al. (1988), De Reyck and Herroelen (1998), Bozejko et al. (2014) and Liberatore and Pollack-Johnson (2006).

In the aforementioned literature, only few papers have considered resource constraints and non-negligible reworks. Furthermore, all these papers assumed a simple linear relationship between the amount of rework and the amount of overlap, which is not realistic. In order to fill this gap, Berthaut et al. (2011) proposed a formulation based on overlapping modes. Preliminary information can be issued at predefined intermediate points corresponding to the completion of internal milestones of the upstream activity. An expected amount of rework in the downstream activity is associated with the release of preliminary information, which is not required to be linear with the amount of overlap. If the downstream activity starts at a given milestone or before the next milestone, the same expected amount of rework is considered. The expected total amount of rework is a piecewise constant function of the amount of overlap, where each step is called an overlapping mode. Linear integer programming models have also been proposed in the last few years for the RCPSP with activity overlapping. Such models were described, for example, by Berthaut et al. (2011), Grèze et al. (2014a, 2014b), Berthaut et al. (2014) and Grèze et al. (2011). Baydoun et al. (2016) also developed a rough-cut capacity planning model for overlapping work packages. In these models, the overlapping modes are converted for convenience into activity modes, which represent all the combinations of overlapping modes of an activity with the associated overlappable activities. As a consequence, the number of overlapping variable decisions is very large.

The main objective of this paper is to measure and analyse the effects of project characteristics, such as project size, resource constraints, overlapping opportunities and rework, on the efficiency of overlapping. The efficiency of overlapping is measured by the relative makespan gain from the project without overlapping, while limiting the use of overlapping. The makespan and the overlapping decisions are computed by solving the RCPSP with overlapping modes with two optimization techniques. First, a 0-1 integer linear program is introduced. The program directly models the overlapping modes, instead of modeling activity modes. The model is solved with the CPLEX solver. An advanced scatter search-based metaheuristic is also proposed to solve the RCPSP with overlapping modes. The mathematical model and the metaheuristic are tested on a set of instances generated with a full factorial experimental design of the project parameters. A statistical analysis of the results enables to capture the influence of the projects characteristics on the efficiency of overlapping and to identify the most efficient practical overlapping strategies to improve the makespan gain.

The remainder of the paper is organized as follows. Section 2 describes the problem statement and assumptions. The 0-1 integer linear programming model is presented in section 3. The metaheuristic is described in section 4. Section 5 presents the generator of projects with overlapping. The computational results are summarized and analysed in section 6 and the findings are compared to practical overlapping strategies in section 7. Finally, section 8 concludes the paper with recommendations for future work.

2 PROBLEM STATEMENT AND ASSUMPTIONS

A project is defined by a set of activities S including two fictitious activities 0 and n+1 which correspond to the project start and end, respectively. Let dj be the normal duration of activity j without overlapping. The symbols used throughout the paper are defined in **Table 1**.

The following assumptions are considered:

1) The information flow is unidirectional from upstream to downstream activities. Feedback information from downstream activities can lead to modifications in the upstream activities and cause iterations in the case of interdependent activities (Wang and Lin, 2009). Design structure matrix, block triangularization algorithms, and aggregation and decomposition of activities can be used to determine a sequence of activities without any feedback (Browning, 2001). We assume that such preliminary studies have been conducted.

Table 1: Symbols and definitions

Symbol	Definition
n	number of non-dummy activities
N_c	number of overlappable pairs of activities
S	Set of activities, $S = \{0, ..., n+1\}$
d_j	Normal duration of activity j without overlapping
A	Set of overlappable pairs of activities $(A_1,\ldots, A_{Nc})$
P	Set of classical finish-to-start precedence relations (no overlap is possible)
E	Set of temporal or precedence constraints $i \rightarrow k$ (i, k), $E = A \cup P$
Po_j	Set of immediate predecessors of activity j that are overlappable with activity j
Pn_j	Set of immediate predecessors of activity j that are not overlappable with activity j
P_j	Set of immediate predecessors of activity j, $P_j = Po_j \cup Pn_j$
Po	Set of activities that are overlappable with at least one direct successor
So_j	Set of immediate successors of activity j that are overlappable with activity j
Sn_j	Set of immediate successors of activity j that are not overlappable with activity j
S_j	Set of immediate successors of activity j, $S_j = So_j \cup Sn_j$
So	Set of activities that are overlappable with at least one direct predecessor
R	Set of renewable resources
R_k	Constant amount of available units of renewable resource k
R_{jk}	Per period resource requirement of activity j for renewable resource k
m_{ij}	Number of precedence/overlapping modes of pair (i, j); if not overlappable, $m_{ij} = 1$, else $m_{ij} > 1$
α_{ijm}	Maximum amount of overlap when pair (i, j) is executed in mode m; $\alpha_{ij1} = 0$ (no overlap)
r_{ijm}	Amount of rework in the downstream activity j when pair (i, j) is executed in mode m, $r_{ij1} = 0$
T	Upper bound of the project makespan
$t = \{0, ..., T\}$	Periods
ES_j, LS_j	Earliest and latest possible start time of activity j, respectively
EF_j, LF_j	Earliest and latest possible finish time of activity j, respectively
$milestone_j^a$	time of the milestone of activity j for a progress of a%

2) *Preliminary information exchange is allowed between overlappable activities and is instantaneous*. Dependent activities can be categorized into non-overlappable and overlappable activities. Non-overlappable activities are represented by classical finish-to-start precedence relations, where a downstream activity requires the completion of an upstream activity. Overlappable activities are represented by a finish-to-start plus-lead time precedence constraint, where the lead-time accounts for the amount of overlap, and thus depends on the overlapping decisions. Information exchange is assumed to require negligible time.

3) *Overlapping can be executed according to overlapping modes*. In practice, the activity progress is measured according to the completion of internal milestones (major events), such as design criteria frozen, detailed design completed, drawings finalized or any activity deliverable. This preliminary information is used as input for downstream activities. The rework caused by overlapping on the downstream activities is defined for each milestone of the upstream activity and remains constant between two consecutive milestones. These overlapping modes can be seen as different overlapping configurations from no overlapping to aggressive overlapping.

4) *The problem is formulated in a deterministic environment*. Scheduling is performed on a period-by-period basis: resource availabilities and allocations are estimated per period, while activity durations and the amounts of rework and overlap are discrete multiples of one period (Hartmann, 1999). The expected amount of rework is assumed to be preliminary known for each overlapping mode of each pair of overlappable activities.

The concept of overlapping modes introduced by Berthaut et al. (2011) is used in the present paper. The overlapping process of two dependent activities (i, j) in A with overlapping modes is depicted in Figure 1. The downstream activity j starts with preliminary inputs from activity i. The amount of overlap is defined as the difference between the finish time of activity i and the start time of activity j. As the upstream activity proceeds, its information evolves to its final form and is released at intermediate points to the downstream activity j. The expected amount of rework in the downstream activity is denoted by r_{ijm} in mode $m \in \{1, ..., m^{ij}\}$ with $r_{ijm} \geq 0$. The concept of overlapping modes can be generalized to precedence/overlapping modes in order to describe all precedence constraints $(i, j) \in E$. For each precedence constraint $(i, j) \in P$, there is only one mode ($m_{ij} = 1$) with $\alpha_{ij1} = 0$ and $r_{ij1} = 0$. If $(i, j) \in A$, there exist

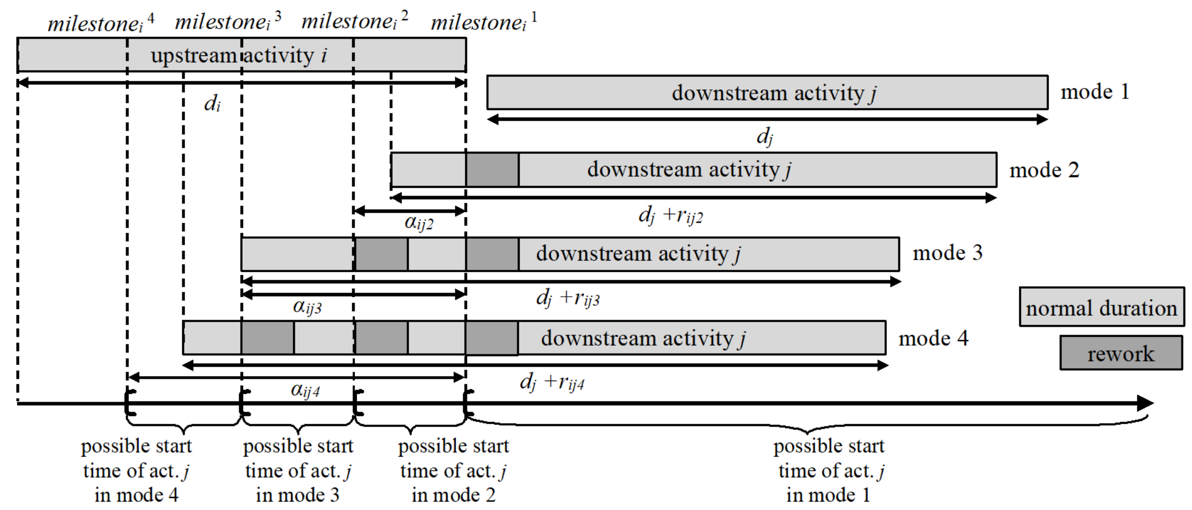


Fig. 1: Overlapping process with overlapping modes

(1) $$X_{jt} = \begin{cases} 1 & \text{if activity } j \text{ finishes at time } t \\ 0 & \text{otherwise} \end{cases} \quad \forall j \in S \quad \forall t \in \{0..T\}$$

(2) $$c_{ijm} = \begin{cases} 1 & \text{if } (i, j) \text{ is executed in mode } m \\ 0 & \text{otherwise} \end{cases} \quad \forall (i, j) \in A \quad \forall m \in \{1..m_{ij}\}$$

(3) $$U_{jt} = \begin{cases} 1 & \text{if time } t \text{ is greater th an the start date of activity } j \\ 0 & \text{otherwise} \end{cases} \quad \forall j \in So \quad \forall t \in \{0..T\}$$

(4) MINIMIZE $$\sum_{t=EF_{n+1}}^{t=LF_{n+1}} t \cdot X_{n+1t} + \frac{1}{2}\left(\frac{\sum_{(i,j)\in A} \sum_{m=1}^{m=m_{ij}} r_{ijm} \cdot c_{ijm}}{\sum_{(i,j)\in A} r_{ijm_{ij}}} + \frac{\sum_{(i,j)\in A} \sum_{m=1}^{m=m_{ij}} m \cdot c_{ijm}}{\sum_{(i,j)\in A} m_{ij}} \right)$$

Subject to:

(5) $$\sum_{t=EF_i}^{t=LF_i} t \cdot X_{it} \leq \sum_{t=EF_j}^{t=LF_j} t \cdot X_{jt} - d_j - \sum_{b \in Po_j} \sum_{m=2}^{m=m_{bj}} r_{bjm} \cdot c_{bjm} \quad \forall (i, j) \in P$$

(6) $$\sum_{t=EF_i}^{t=LF_i} t \cdot X_{it} - \sum_{m=2}^{m=m_{ij}} \alpha_{ijm} \cdot c_{ijm} \leq \sum_{t=EF_j}^{t=LF_j} t \cdot X_{jt} - d_j - \sum_{b \in Po_j} \sum_{m=2}^{m=m_{bj}} r_{bjm} \cdot c_{bjm} \quad \forall (i, j) \in A$$

$$\sum_{t=EF_i}^{t=LF_i} t \cdot X_{it} - 1 - \sum_{m=2}^{m=m_{ij}} \alpha_{ijm-1} \cdot c_{ijm} \geq \sum_{t=EF_j}^{t=LF_j} t \cdot X_{jt} - d_j - \sum_{b \in Po_j} \sum_{m=2}^{m=m_{bj}} r_{bjm} \cdot c_{bjm} - (LS_j - EF_i + 1) \cdot c_{ij1} \quad \forall (i,j) \in A \tag{7}$$

$$\sum_{t=EF_i}^{t=LF_i} t \cdot X_{it} \leq \sum_{t=EF_l}^{t=LF_l} t \cdot X_{lt} - d_l - \sum_{b \in Po_l} \sum_{m=2}^{m=m_{bl}} r_{blm} \cdot c_{blm} \quad \forall j \in Po \cap So \quad \forall i \in Po_j \tag{8}$$

$$t - \sum_{b=EF_j}^{b=LF_j} b \cdot X_{jb} + d_j + \sum_{i \in Po_j} \sum_{m=1}^{m=m_{ij}} r_{ijm} \cdot c_{ijm} \leq (T - ES_j + 1) \cdot U_{jt} \quad \forall l \in So_j \quad \forall t \in \{0..T\} \tag{9}$$

$$\sum_{j \in S \setminus So} R_{jk} \cdot \sum_{b=\max(t, EF_j)}^{b=\min(t+d_j-1, LF_j)} X_{jb} + \sum_{i \in So} R_{jk} \cdot \left(U_{it} - \sum_{b=EF_i+1}^{b=t} X_{i,b-1} \right) \leq R_k \quad \forall k \in R \quad \forall t \in \{0..T\} \tag{10}$$

$$\sum_{t=EF_j}^{t=LF_j} X_{jt} = 1 \quad \forall j \in S \tag{11}$$

$$\sum_{m=1}^{m=m_{ij}} c_{ijm} = 1 \quad \forall (i,j) \in A \tag{12}$$

$$X_{jt} = \{0,1\} \quad \forall j \in S \quad \forall t \in \{0..T\} \tag{13}$$

$$c_{ij} = \{0,1\} \quad \forall (i,j) \in A \tag{14}$$

$$U_{jt} = \{0,1\} \quad \forall l \in So_j \quad \forall t \in \{0..T\} \tag{15}$$

additional overlapping modes associated with the milestones of the upstream activity ($m_{ij} > 1$). The precedence constraint of two overlappable activities (i, j) depends on the overlapping mode: when not overlapped ($m = 1$), the downstream activity start time is superior or equal to the upstream activity finish time, whereas the downstream activity start time belongs to the interval $\{milestone_i^{m}, milestone_i^{m-1} - 1\}$ in the case of overlapping in mode $m \in \{2, ..., m^{ij}\}$.

There is no restriction concerning the number of overlappable predecessors. If a downstream activity can be overlapped by several upstream activities, the amount of rework in downstream activity is assumed to be the sum of the amounts of rework caused by each upstream activity (Cho and Eppinger, 2005). If an activity j is both the upstream activity for a pair (j, l) and the downstream activity for another pair (i, j), then activity l must start after the end of activity i in order to eliminate the influence of information change in activity i on activity l. This assumption is referred to as "sashimi-style" overlapping (Imai et al., 1985 and Roemer and Ahmadi, 2004).

3. Mathematical model for the RCPSP with overlapping modes

In this section, a 0-1 integer linear programming model of the project makespan minimization problem with overlapping modes is presented. The model makes directly use of the overlapping modes.

3.1 *0-1 integer linear programming model*

Each activity j in S must finish within the time window $\{EF^j, ..., LF^j\}$. The proposed model is derived from the well-known model introduced by Pritsker et al. (1969). The decision variables are defined as (1) to (3) expressions.

The X_{jt} and c_{ijm} variables determine the execution dates and the overlapping decisions, respectively. U_{jt} is a binary variable introduced to linearize the optimization model. The main objective is to find a precedence, overlapping and resource-feasible schedule that minimizes the project makespan. A second objective is added to limit overlapping in case of tie. The problem is formulated as (4) to (15) expressions.

The first part of the objective function (4) represents the project makespan. The second part is a global measure of the degree of overlapping, composed of the ratio between the sum of the amounts of rework and the sum of the maximum possible amounts of rework, and the ratio between the sum of the modes and the sum of the maximum modes. This measure of overlapping belongs to the interval [0,1[while the project makespan is an integer value. Constraints (5) represent the finish-to-start precedence constraints for the activities that cannot overlap. Constraints (6) and (7) reflect the constraints depicted in Figure 1. If (i, j) is overlapped in mode $m > 2$, then constraints (6) and (7) guarantee that the start time of activity j belongs to the interval [$milestone_i^{\,m}$, $milestone_i^{\,m-1}$[. If (i, j) is not overlapped ($c_{ij1} = 1$), then constraints (7) are not restrictive and constraints (6) represent classical finish-to-start precedence constraints. Constraints (8) represent the "sashimi-style" overlapping assumption. Constraints (9) state that the additional variable U_{jt} is equal to 1 when t is superior to the start date of activity j in So. Constraints (10) guarantee that the resource consumption does not exceed the resource capacity. For any period t in $\{0,\ldots,T\}$, the resource consumption is obtained by summing the resource consumptions of all activity executed in period t. For any activity i in So, the expression with parentheses in Constraints (10) is equal to 1 for all the periods where activity i is executed. Constraints (11) and (12) ensure that each activity and each pair of overlappable activities is assigned only one finish time and one overlapping mode, respectively. Finally, constraints (13) to (15) define the binary decision variables. The model is solved using CPLEX.

3.2) *Constraint propagation as preprocessing*

The most basic way to derive the time windows $\{EF^j, \ldots, LF^j\}$ for a given upper bound is to apply a modified forward and backward recursion algorithms that include overlapping. However, this method overlooks the resource constraints. To illustrate how resource constraints can be involved, consider two independent activities $(i, j) \notin E$ such that the sum of the required resources exceeds the resource capacities. Obviously, these activities cannot be executed in parallel. In addition, if their time windows are such that $LF_i - ES_j < d_i + d_j$, then one can deduce that activity i must be executed before activity j. By applying again the forward and backward recursion algorithms, the time windows may be tightened. To include the resource constraints, constraint propagation can be used as pre-processing technique. In this paper, a constraint propagation algorithm is adapted to the RCPSP with overlapping. The proposed algorithm is an iterative process involving path consistency, the immediate selection algorithm (Carlier and Pinson, 1989), the symmetric triples rules (Brucker et al., 1998), edge-finding rules (Demassey et al., 2005) and shaving techniques to deduce new relations. These rules already exist in the literature, although they have been modified to take into account overlapping. Additional rules specific to overlapping are also introduced. The constraint propagation algorithm stops when no more adjustments can be performed or when infeasibility is detected. The latter means that $T + 1$ constitutes a lower bound of the problem. If infeasibility has not been detected, the model presented in section 3.1 can be enhanced to include the information derived from the constraint propagation algorithm. The algorithm will not be presented here but we instead refer the interested reader to the works of Brucker and Knust (2000, 2003, 2012), Sprecher et al. (1997), Demassey et al. (2005), Brucker et al. (1998) and Carlier and Pinson (1989).

4. A path relinking-based scatter search for the RCPSP with overlapping

Scatter search (SS) is an evolutionary algorithm that has been successfully applied to the RCPSP by Valls et al. (2004), Debels et al. (2006), Ranjbar et al. (2009), Mobini et al. (2009), Chen et al. (2010), Paraskevopoulos et al. (2012) and Berthaut et al. (2018). The algorithm proposed in this paper for the RCPSP with overlapping is depicted in Figure 2. This algorithm is derived from the high-quality metaheuristic developed by Berthaut et al. (2018) to solve the standard RCPSP.

The scatter search algorithm first generates an initial population of size *InitPop*. An activity list and an overlapping list must be generated for each member of the population. Each overlapping list is generated by randomly selecting an overlapping mode between 1 and m_{ij} for each pair $(i, j) \in A$. Then, a reference set *RefSet* composed of two distinct sets $RefSet_1$ and $RefSet_2$ of high-quality and diversified solutions is built from the population of solutions and will be evolved to form a new population. This new population is initialized with the best current solution. Note that the construction of $RefSet_1$ and $RefSet_2$ starts by sorting the new population according to the lowest makespan. In case of tie, a second measure is used to choose the solution that overlap the least, depicted in the second part of the objective presented in section 3.1. As a third measure to break the tie, a function used by Paraskevopoulos et al. (2012) and Berthaut et al. (2018) is applied to measure the deviation of the schedule from the earliest finish times. The deviation is weighted with the activity durations and the resource consumptions. The *RefSet* update mechanism maintains two matrices to track the previously generated activity and overlapping lists.

The reference solutions are afterwards paired to form subsets that are combined with a path relinking method (PR) to generate new solutions. These solutions are evaluated and either directly added to the new population or first improved by forward-backward improvement (FBI) depending on their quality. The PR method generates a new solution from a high-quality solution by making alternatively one move on the activity list and one move on the overlapping list towards another high-quality solution. The scatter search algorithm stops when the number of generated schedules *NSched* reaches *NSched_limit*.

It is worth noting that the scheduling direction (forward or backward) and the project network are reversed at each iteration of the algorithm. Two different modified versions of the

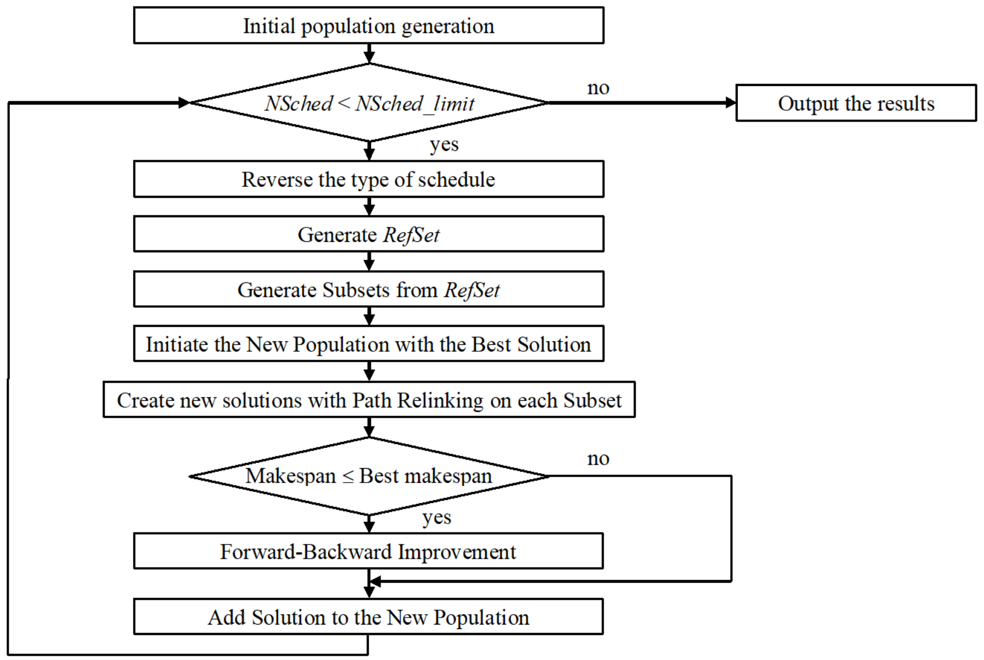


Fig. 2: Flow chart of the proposed scatter search algorithm

serial generation scheme have thus been developed for the forward and backward scheduling. The overlapping list remains the same when the scheduling direction is reversed. Also, the modified version of the topological order (TO) representation used by Ranjbar et al. (2009) and Berthaut et al. (2018) is adopted in order to embed the TO representation into the reversing of the project network. For further details on each step of the scatter search algorithm, we refer the interested reader to recent work by Berthaut et al. (2018).

5. Instance generator with overlapping

To evaluate the performance of the two methods presented in section 3 (mathematical model solved with CPLEX) and section 4 (scatter search) to solve the RCPSP with overlapping, three sets of instances composed of projects with 30, 60 and 120 activities are generated. The instances are derived from the PSPLIB benchmark designed for the standard RCPSP (Kolisch and Sprecher, 1997) and additional overlapping data are generated. The levels defined in Kolisch et al. (1999) are used, namely $NC \in \{1.5, 1.8, 2.1\}$, $RF \in \{0.25, 0.5, 0.75, 1\}$ and $RS \in \{0.2, 0.5, 0.7, 1\}$, where NC corresponds to the average number of non-redundant precedence relations per activity, RF gives the average percent of different resource types required by an activity and RS is a normalized indicator that defines how scarce the resources are.

Therefore, 48 instances are considered for each set of 30, 60 and 120 activities. For each instance, 27 instances with overlapping data are generated. Three additional parameters are introduced to generate the overlapping data: the proportion of pairs of overlappable activities among all the precedence relations (OC), the rework rate (RR) and the maximum amount of overlap (MO). OC indicates how many precedence constraints can be relaxed to authorize overlapping, due to the nature of the project and the content of the activities. In order to characterize the rework with a single parameter RR, a linear relation is assumed between the total amount of rework and the amount of overlap. Roughly speaking, RR represents the global sensitivity of the downstream activities to changes in the input information provided by the upstream activities. Also, the completion dates of the internal milestones of each upstream activity is assumed to correspond to 0%, 25%, 50% and 75% of the duration of the activity. The maximum number of modes is 4. If MO = 75%, it means that the downstream activity could start once the upstream progress reaches the internal milestone associated with 1 - MO = 25%, and that there are four modes (i.e., 0%, 25%, 50% and 75%). The maximum amount of overlap is also constrained by the durations of the activities in such a way that the downstream activity must start and end after the upstream activity. As the parameters of the projects are assumed to be integer values, the amounts of overlap and rework must be rounded. The overlapping data generation proceeds as follows for each PSPLIB instance:

1. Generate the set A of pairs of overlappable activities:
   a. the total number of pairs of overlappable activities is given by
   $$|A| = \text{round}(OC \cdot |E|)$$
   ;
   b. A is built by randomly selecting $|A|$ pairs of activities among the set E.
2. Generate the modes and the amounts of overlap for each pair $(i, j) \in A$. The maximum overlap MO and the durations of the upstream and downstream activities d_i and d_j provide the amount of overlap associated with each mode as follows:
   $$\alpha_{ij,m_{ij}} = \text{round}\left(d_i \cdot \min\left(\frac{d_j}{d_i}; MO\right)\right)$$
   ;
   $$m_{ij} = \text{ceil}\left(\frac{\alpha_{ij,m_{ij}}}{0.25}\right) + 1$$
   ,
   $$\alpha_{ijm} = \text{round}\left(0.25 \cdot (m-1)\right)$$
   ,
   $$\forall m \in \{2, .., m_{ij} - 1\}$$
   .

The modes with the same amount of overlap after rounding are merged into a single mode.

3. Generate the amounts of rework for each pair $(i, j) \in A$. The amount of rework is computed by

$$r_{ijm} = \text{round}\left(\alpha_{ijm} \cdot RR\right),$$
$$\forall m \in \left\{2,..,m_{ij}-1\right\}.$$

The levels used to generate the whole benchmark set are summarized in Table 2. The benchmark set is a full factorial design of the seven parameters, thus composed of 3888 instances.

Size	*NC*	*RF*	*RC*	*OC*	*MO*	*RR*
30	1.5	0.25	0.2	0.2	0.25	0.25
60	1.8	0.5	0.5	0.4	0.5	0.5
120	2.1	0.75	0.7	0.6	0.75	0.75
		1	1			

Table 2: Parameter levels of the benchmark set

6. Computational experiments

The model defined in section 3 is first solved using CPLEX to find the optimal solutions of the 1296 project instances generated for 30 activities. The scatter search algorithm proposed in section 4 is then applied to the whole benchmark of 3888 project instances with 30, 60 and 120 activities. A statistical analysis is also presented to model the influence of project characteristics on the makespan gain with overlapping.

6.1 Results with the exact procedure for 30 activities

The upper bound T is initially set at M_{no} - 1, where M_{no} is the optimal makespan of the RCPSP without overlapping available on the PSPLIB website (http://www.om-db.wi.tum.de/psplib/files/j30opt.sm). The constraint propagation (CP) algorithm was implemented in MATLAB R2011b, while ILOG-CPLEX 12.5 was used as solver for the mathematical programming (MP) model. The tests were carried out on a personal computer with an AMD FX-6100 six-core clocked at 3.30 GHz with 16 GB RAM using Windows 7 Enterprise. The average time to perform the constraint propagation algorithm is 0.39 sec. Infeasibility was detected for 136 out of the 1296 instances, which means that the optimal solution is to not overlap. For the 1160 remaining instances, an average reduction of 7.26% of the length of the time windows is observed. Within a time limit of 20000 sec., an optimal solution for the modified 0-1 integer linear programming model is found in 1105 instances. The 55 remaining instances that cannot be solved to optimality are mostly derived from instances known to be the hardest of PSPLIB. Nevertheless, a lower bound LB of the makespan can be found for each instance and possibly a best integer solution to be used as new upper bound T. Since the optimal solution is comprised between LB and T, a destructive procedure is introduced to improve the results. This procedures consists in restricting the problem by setting a maximal objective function value F and trying to contradict (destruct) the feasibility of this reduced problem. In case of success, $F + 1$ is a valid new lower bound. For further details on destructive procedures, see Klein and Scholl (1999). Out of the 55 instances tested with the destructive procedure, the optimal schedule and overlapping decisions have been found for 20 instances. For 8 other instances, an integer solution is found for the optimal makespan, but the overlapping modes are not proved to be optimal. Finally, 27 instances cannot be solved, but the interval $\{LB, \ldots, T\}$ has been reduced. The results are summarized in Table 3. Almost 98% of the instances have been solved to optimality with respect to the makespan, while almost 97% have been solved to optimality with respect to both the makespan and the overlapping decisions. The average CPU time required to solve the instances is 7184.81 sec. and about 70% of the instances can be solved in less than 10 sec. To measure the global effect of overlapping on the project duration, the makespans obtained for the RCPSP with and without overlapping are compared. The average makespan gain is 4.11% over the instances that are solved to optimality.

6.2 Performances of the scatter search algorithm

In the last section, an important computation effort is observed for solving the RCPSP with overlapping with an exact procedure for projects of 30 activities. The objective of this section is to show the efficiency of the scatter search (SS) algorithm to tackle this issue. The SS algorithm was implemented in MATLAB R2011b. The experiments were conducted on a personal computer with an Intel Core I5 2.53 GHz processor and 4 GB RAM using Windows 7 Professional. As the proposed SS is based on random devices, each instance is solved ten times and the average results of the ten replications, as well as the 95% confidence interval are provided. The algorithm is tested with 1000, 5000 and 50000 generated schedules as stopping criteria (Hartmann and Kolisch, 2000; Kolisch and Hartmann, 2006). Parameter tuning was performed by applying a local search process based on the idea that determining the values of the parameters can be formulated as an optimization problem (Talbi, 2009). Table 4 presents the resulting combinations of parameters.
Detailed results are presented in Table 5. The row "Avg. dev. CPM" gives the average deviation of the best makespan from the critical-path lower bound with overlapping. The rows "Avg. dev. LB" and "Avg. dev. UB" represent the average deviation from the best lower and upper bounds found in section 6.1, respectively. The row "Avg. dev. Opt." is the average deviation from the optimal makespan for the 1261 instances solved to

No. of instances	*Prop. of instances*	*Optimal makespan reached*	*Optimal schedule known*	*Optimal overlapping modes known*	*Method*
136	10.50%	yes	yes	yes	CP
1105	85.26%	yes	yes	yes	CP + MP
20	1.54%	yes	yes	yes	CP + MP, destructive procedure
8	0.62%	yes	yes	no	CP + MP, destructive procedure
27	2.08%	no	no	no	CP + MP, destructive procedure

Table 3: Parameter levels of the benchmark set

Size	30			60			120		
Schedule limit	1000	5000	50000	1000	5000	50000	1000	5000	50000
InitPop	300	800	1000	200	600	1500	200	300	1300
b_1	11	22	75	8	15	38	6	11	17
b_2	6	12	24	4	8	22	4	7	16
t_1	0.6	0.5	0.5	0.9	1.3	1.4	1	1.8	1.8
t_2	1.5	1.4	1.5	1.6	1.8	2.1	2.3	2.7	3.2
n_{pr}	1	1	2	1	1	1	1	1	1

Table 4: Parameter tuning for the proposed scatter search for the RCPSP with overlapping

Size	30			60			120		
Schedule limit	1000	5000	50000	1000	5000	50000	1000	5000	50000
Avg. dev. CPM	17.97%	17.53%	17.33%	14.61%	13.88%	13.31%	16.59%	15.69%	14.92%
Avg. dev. LB	0.65%	0.34%	0.18%	-	-	-	-	-	-
Avg. dev. UB	0.57%	0.26%	0.10%						
Avg. dev. Opt.	0.52%	0.25%	0.10%	-	-	-	-	-	-
Avg. No. of LB	959	1092	1191	-	-	-	-	-	-
Avg. Prop. of LB	74%	84%	92%	-	-	-	-	-	-
Avg. CPU (s)	2.12	7.06	82.02	4.86	17.68	142.04	35.18	43.74	273.02
Max. CPU (s)	10.85	57.44	1689.38	22.43	116.01	1583.15	47.70	300.43	2368.91
Avg. No. sched.	219	657	3268	259	930	5748	308	1083	6791

Table 5: Detailed performance for the proposed Scatter search

optimality in section 6.1. “Avg. No. of LB” and “Avg. Prop. of LB” are respectively the average number and the proportion of instances for which the best makespan reaches the best LB. “Avg. CPU”, “Max. CPU” and “Avg. No. sched.” represent the average and maximum computation times and the average number of generated schedules to reach the best solution, respectively. Table 5 shows that the average deviation of the best makespan from the best *LB* is 0.18% for the instances of 30 activities with a limit of 50000 schedules. The best *LB* is reached for a large proportion of the instances (92%). If we only consider the 1261 instances solved to optimality in section 6.1, the average deviation of the best makespan from the optimal makespan is 0.10%. These results are obtained with an average computation time of 82.02 sec., far below the one observed with the exact procedure. In addition, the metaheuristic is able to reach the optimal makespan and the optimal overlapping decisions more quickly than the exact procedure for 58% of the instances. This shows the efficiency of the proposed metaheuristic. The method rapidly provides good quality solutions even for lower schedule limits. Also, high quality schedules and overlapping decisions are obtained for the hard instances for which no integer solution was found with the exact method. Table 6 compares the average makespan gain obtained with the two methods for 30 activities. In addition, two overlapping measures are introduced for the overlap decisions: the proportion of overlapped pairs among the set of pairs of overlappable activities and the average amount of overlap for the overlapped pairs expressed as a proportion of the maximum possible amount of overlap. The overlapping measures for the two methods are quite similar, but overlapping appears to be slightly less used in the solutions found with the SS algorithm.

6.3 Efficiency of overlapping for 30, 60 and 120 activities

The efficiency of overlapping is analysed by measuring the makespan gain obtained by the metaheuristic for the whole benchmark with 30, 60 and 120 activities. Ten repetitions are again conducted for each instance in order to reduce the influence of the random devices. Table 7 presents the sensitivity of the makespan gain with respect to the stopping criterion. Not surprisingly, the makespan gain increases with the schedule limit. The histogram of the makespan gain for the whole benchmark set is presented in Figure 3 with the limit of 50000 schedules. The histogram shows that the makespan gain is widespread, with a minimum value of 0 and a maximum value of 28%. The overall mean is 4.43% and the standard deviation is 4.68%. For 25.08% of the instances, overlapping

	Makespan gain				Overlapping measures*	
	Interval (%) from Upper and Lower bounds	Avg. gain (%)	95% Conf. Interval (%)		Avg. Proportion of overlapped pairs (%)	Avg. amount of overlap (%)
Exact procedure	[4.04, 4.11]				8.89	61.87
SS algorithm (50000 sched.)	-	3.96	[3.96, 3.96]		8.61	61.50

*for the subset of 1261 instances for which an optimal schedule is found with the exact procedure.

Table 6: Comparison of the exact procedure and the SS for the RCPSP with overlapping for 30 activities

Size	30			60			120		
Schedule limit	1000	5000	50000	1000	5000	50000	1000	5000	50000
Avg. makespan gain	3.72%	3.82%	3.96%	4.44%	4.58%	4.68%	4.37%	4.51%	4.64%
95% confidence interval (%)	[3.69,3.74]	[3.82,3.83]	[3.96,3.96]	[4.42,4.47]	[4.56,4.60]	[4.67,4.69]	[4.35,4.40]	[4.49,4.53]	[4.63,4.65]

Table 7: Sensitivity of the makespan gain with respect to the stopping criterion and the project size

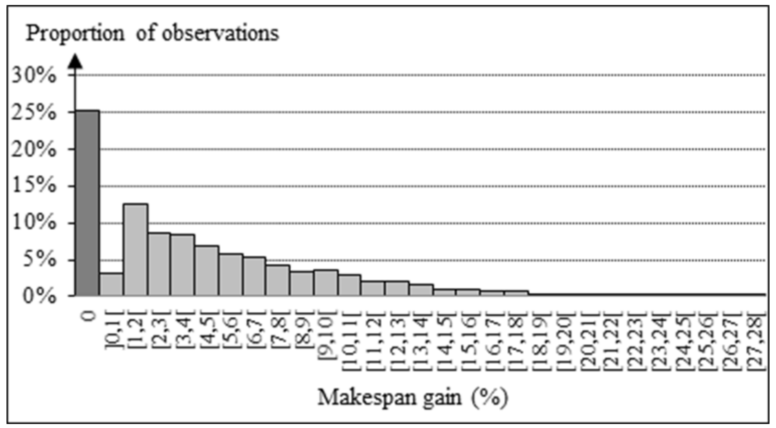


Fig. 3: Histogram of the makespan gain with overlapping

(16) $$\text{Prob}\left(Gain > 0 \mid (x_1,..,x_8, Rep)\right) = \frac{e^{g(x_1,...x_8,Rep)}}{1+e^{g(x_1,...x_8,Rep)}}$$

with $$g(x_1,...,x_8,Rep) = \beta_0 + \sum_{i=1}^{i=8} \beta_i \cdot x_i + \sum\sum_{i<j} \beta_{ij} \cdot x_i \cdot x_j + \sum_{i=1}^{i=8} \beta_{ii} \cdot x_i^2 + \sum_{k=1}^{k=10} \gamma_k \cdot Rep_k$$

(17) $$f = \alpha_0 + \sum_{i=1}^{i=8} \alpha_i \cdot x_i + \sum\sum_{i<j} \alpha_{ij} \cdot x_i \cdot x_j + \sum_{i=1}^{i=8} \alpha_{ii} \cdot x_i^2 + \sum_{k=1}^{k=10} \delta_k \cdot Rep_k$$

(18) $$f = \ln(\mu)$$

does not yield to a makespan gain. For these instances, any overlapping decisions would be inefficient. This raises the question of the influence of each project characteristic on the makespan gain. This issue is tackled in the next section with a statistical analysis.

6.4 *Influence of project characteristics on the efficiency of overlapping*

Figure 3 is a histogram of the makespan gain. The makespan gain is characterized by most of the observations being zero and right skewed continuous distribution for the positive values. In presence of such so-called semi-continuous, two-part models can be used (Wooldridge, 2002), such as generalized linear models with gamma distribution and log link. A logistic regression is used for the first part of the model. Let the variable *Gain* represents the makespan gain. The probability of the makespan gain to be positive is modeled as (16). where $(x_1, \ldots, x_8) = (Size, NC, RF, RS, OC, RR, MO, PC)$ represents the seven project characteristics and an additional continuous variable *PC* that measures the proportion of pair of overlappable activities on the critical path of the project without overlapping. The dummy variables Rep_k represent the effect of repeating on the results (i.e, $Rep_k = 1$ if $Rep = k$, 0 otherwise). The second part of the model predicts how much the makespan gain is conditional on its being positive. The model is composed of the following elements.

The gain is assumed to be generated from a gamma distribution, with a mean μ and a variance $V(\mu)$ proportional to the square of μ such that $V(\mu) = \mu^2 / v = \sigma^2 \cdot \mu^2$, where v and σ represent the shape parameter and the coefficient of variation, respectively. The influence of the covariates is modeled by the linear predictor f in (17).

The log link function relates the linear predictor to the mean μ in (18).

The statistical analysis was conducted using STATISTICA 12. The coefficients of the normalized variables for the two parts of the model are presented in Table 8. Only the significant effects evaluated using the Wald's test with a threshold p-value of 0.05 are presented. For instance, repeating the metaheuristic procedure has a non-significant impact on the makespan gain.

The ability of the logistic regression model to match the predicted and observed makespan gains is measured with the Hosmer and Lemeshow's goodness of fit test with grouped data. The model was assessed on subsamples of approximately 1000 observations (Paul and Pennell, 2013). The model shows no evidence of lack of fit (all p-value are above 0.05). With a cut-off point value of 0.5, the overall rate of correct classification between observed and predicted makespan gains is 85.61%, with a rate of correct 1 (sensitivity) of 92.10% and a rate of correct 0 (*specificity*) of 66.23%. A more complete description of classification accuracy is given by the ROC curve (Receiver Operating Characteristic) which plots the sensitivity and *1 - specificity* for the entire range of possible cut-off points. The area under the ROC curve provides a common measure of discrimination. This measure for the proposed logistic regression is 91.15%, which can be qualified as excellent (Hosmer and Lemeshow, 2000). The Nagelkerke's pseudo R-squared, which measures the overall performance of the logistic regression, is 56.99%. The estimated coefficient of variation, $\hat{\sigma} = 0.44$, for generalized linear model with gamma distribution and log link is derived from the method of moments. Finally, the overall performance of the generalized linear model is measured with a Nagelkerke's pseudo R-squared of 71.65%. The residual analysis based on the procedure of McCullagh and Nelder (1989) was carried out to verify the adequacy of the model.

Figure 4 and Figure 5 show the predicted probability of makespan gain and the predicted makespan gain (conditional on its being positive) as a function of each independent variable, respectively. Each normalized variable is varied from -1 to 1 with all other variables held constant at their mean value. As expected, the probability of makespan gain and the predicted makespan gain (conditional on its being positive) increase when either the size of the project (*Size*), the average number of precedence relations per activity (*NC*), the resource capacity (*RS*), the proportion of pairs of overlappable activities among all the precedence relations (*OC*), the maximum amount of overlap (*MO*), or the proportion of overlappable pairs of activities on the critical path (*PC*) increases. It decreases when either the average number of different resource types required per activity (*RF*) or the rework rate (*RR*) increases. In addition, Figure 4 shows that both *PC* and *RS* have a higher impact on the probability of makespan gain, while the probability of gain is above 90% for any value of the other variables. Figure 5 highlights that *PC* has a higher influence on the makespan gain (conditional on its being positive) than the other parameters, while *NC* has the lower influence. However, the makespan gain (conditional on its being positive) is below 10% when varying the parameters, with the exception of *PC*. This means that the relative influence of *RS* on the makespan gain (conditional on its being positive) is more moderate than its influence on the probability of makespan gain.

As last step of the two-part model, the predicted makespan gain is obtained by combining the two models in (19).

Effect	Logistic regression model			Gamma log link model (positive part)		
	Coefficient	Wald Statistic	p-value	Coefficient	Wald Statistic	p-value
Intercept	-5.582	1923.15	0.0000	-1.999	19082.73	0.0000
Size	-1.096	97.67	0.0000	0.543	1250.04	0.0000
NC	-0.279	129.91	0.0000	0.020	33.5	0.0000
RF	2.125	660.34	0.0000	-0.212	439.21	0.0000
RS	-3.925	1567.33	0.0000	0.820	5849.91	0.0000
OC	-0.625	76.73	0.0000	0.394	12715.39	0.0000
MO	-0.668	617.7	0.0000	0.315	8454.49	0.0000
RR	1.163	275.14	0.0000	-0.355	1563.25	0.0000
PC	-3.001	148.79	0.0000	1.151	1253.13	0.0000
*Size*NC*	0.155	38.76	0.0000	0.017	17.88	0.0000
*Size*RF*	-0.403	157.71	0.0000	0.088	284.85	0.0000
*NC*RF*	0.202	50.91	0.0000	—	—	—
*Size*RS*	-0.840	448.77	0.0000	0.109	361.56	0.0000
*NC*RS*	-0.155	24.43	0.0000	0.035	51.56	0.0000
*RF*RS*	-0.338	75.85	0.0000	0.264	2441.34	0.0000
*Size*OC*	0.254	71.38	0.0000	0.037	86.25	0.0000
*RF*OC*	0.268	80.15	0.0000	-0.071	255.96	0.0000
*RS*OC*	-0.488	188.06	0.0000	0.170	1217.25	0.0000
*Size*MO*	0.165	44.12	0.0000	0.026	46.26	0.0000
*NC*MO*	-0.070	8.06	0.0050	0.011	7.65	0.0060
*RF*MO*	0.209	53.44	0.0000	-0.021	23.67	0.0000
*RS*MO*	-0.358	109.1	0.0000	0.101	434.62	0.0000
*OC*MO*	-0.054	4.28	0.0390	0.022	30.17	0.0000
*Size*RR*	-0.158	30.78	0.0000	-0.022	22.72	0.0000
*NC*RR*	-0.149	34.47	0.0000	-0.019	22.41	0.0000
*RF*RR*	-0.162	29.87	0.0000	0.026	34.68	0.0000
*RS*RR*	0.406	136.84	0.0000	-0.056	134.16	0.0000
*OC*RR*	-0.291	131.6	0.0000	-0.040	97.98	0.0000
*MO*RR*	—	—	—	0.066	266.79	0.0000
*Size*PC*	-0.412	7.38	0.0070	0.555	503.1	0.0000
*RF*PC*	1.996	302.65	0.0000	-0.052	8.58	0.0030
*RS*PC*	-2.714	423.77	0.0000	0.411	491.95	0.0000
*OC*PC*	0.323	9.37	0.0020	—	—	—
*RR*PC*	0.496	26.12	0.0000	-0.046	8.41	0.0040
$Size^2$	0.320	55.24	0.0000	-0.046	39.31	0.0000
RS^2	0.536	163.8	0.0000	-0.419	5111.01	0.0000
OC^2	0.385	124.1	0.0000	-0.099	325.96	0.0000
MO^2	—	—	—	-0.127	531.27	0.0000
RR^2	0.141	16.1	0.0000	-0.091	277.47	0.0000
PC^2	1.861	144.82	0.0000	-0.363	155.64	0.0000

Table 8: Results of fitting the logistic regression model and the gamma log link model

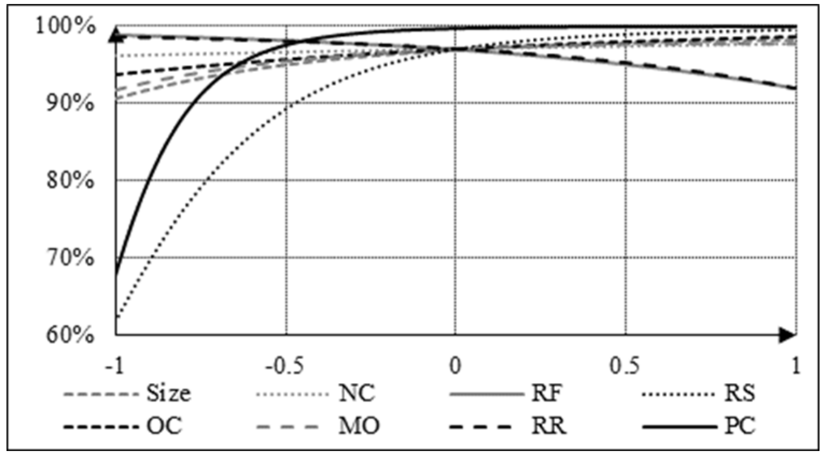


Fig. 4: Predicted probability of makespan gain as a function of the parameters

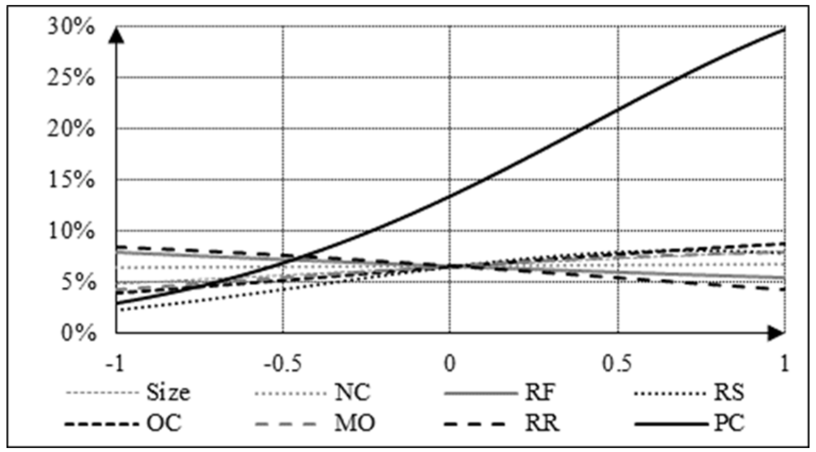


Fig. 5: Predicted makespan gain (conditional on its being positive) as a function of the parameters

(19)

$$\mathrm{E}\left(Gain \mid (x_1,\ldots,x_8)\right) = \mathrm{Prob}\left(Gain > 0 \mid (x_1,\ldots,x_8)\right) \cdot \mathrm{E}\left(Gain \mid (x_1,\ldots,x_8), Gain > 0\right)$$

$$\mathrm{E}\left(Gain \mid (x_1,\ldots,x_8)\right) = \frac{e^{g(x_1,\ldots,x_8)}}{1+e^{g(x_1,\ldots,x_8)}} \cdot e^{f(x_1,\ldots,x_8)}$$

This two-part model applied to the whole set of instances exhibits a mean absolute error of 1.64% between the predicted and observed makespan gain and a coefficient of determination (R squared) of 74.96%.

6.5 *The best overlapping decisions*

Table 9 shows the frequency of distribution of the proportion of overlapped pairs among all the pairs of overlappable activities for the whole benchmark set. The table highlights that the proportion of overlapped pairs is generally small, with a mean of 6.94%, a minimal value of 0% (i.e., no overlapping) and a maximal value of 45%. Almost 95% of the instances have a proportion of overlapped pairs below 20%. If only the instances with an observed strictly positive makespan gain are considered, the mean proportion of overlapped pairs is 9.23%. In addition, Table 10 highlights that the average amount of overlap among the overlapped pairs (i.e., the ratio of the amount of overlap by the maximum possible amount of overlap) is quite large. The average value is 62.65%, with 0% as minimal value and 100% as maximum value. If only the instances with a strictly positive makespan gain are considered, the mean value is 81.97% and more than 95% of the instances have an average amount of overlap above 40%. These results suggest that the best strategy would consist in overlapping only few pairs of overlappable activities with a large degree of overlapping. This conclusion has several practical consequences for planning and controlling a project. Effective overlapping can be applied by targeting only few pairs of activities. Also, if overlapping requires specific communication and control processes in practice, this work only needs to be focused on few activities. The distribution of the observed proportion of overlapped pairs on the critical path is given in Table 11. While the average proportion of pairs of overlappable activities which are on the critical path without overlapping is 9.86% in the whole benchmark set, the average proportion of overlapped pairs on the critical path among all the overlapped pairs is 56.64%. This means that the pairs of overlappable activities on the critical path are more likely to be overlapped. Also, for projects with a complex network and resource constraints, many pairs of overlappable activities that are not critical are overlapped and thus the overlapping decision should not rely solely on the criticality of the activities.

7 Strategies for improving the efficiency of overlapping

Practical strategies have been proposed in the literature to improve the efficiency of overlapping, based on the concepts of evolution of the upstream information and the sensitivity of the downstream activity,

Proportion of overlapped pairs among all the pairs of overlappable activities	0%	]0%,5%[	[5%,10%[	[10%,15%[	[15%,20%[	]20%,45%]
Proportion of instances	26.09%	19.14%	25.22%	17.53%	6.97%	5.05%

Table 9: Frequency of distribution of the proportion of overlapped pairs

Avg. amount of overlap	0%	]0%,40%[	[40%,60%[	[60%,80%[	]80%,100%]	100%
Proportion of instances	26.09%	1.31%	6.34%	18.18%	14.33%	33.75%

Table 10: Frequency of distribution of the average amount of overlap for the overlapped pairs

Avg. proportion of overlapped pairs which are on the critical path	0%	]0%,40%[	[40%,60%[	[60%,80%[	]80%,100%]	100%
Proportion of instances	31.25%	3.24%	10.25%	12.26%	4.73%	38.27%

Table 11: Frequency of distribution of the average proportion of overlapped pairs on the critical path

Overlapping strategies	Influence on *PC*	Influence on *OC*	Influence on *MO*	Influence on *RR*	Influence on *RS*
1) Strategies to begin the downstream activities earlier (Wang and Lin 2009; Bogus et al. 2006)	x	x	x	x	
1) Strategies to add resource capacities					x
2) Strategies to reduce the sensitivity of downstream activities (Wang and Lin 2009; Bogus et al. 2006; Blacud et al. 2009)				x	

Table 12. Summary of overlapping strategies to improve the efficiency of overlapping

introduced by Krishnan et al. (1997). The upstream information evolution characterizes the refinement of information from its preliminary form to a final value, whereas the downstream sensitivity represents the amount of rework required to incorporate upstream changes. Table 12 summarizes these strategies and their relationship to the most influent project characteristics observed in section 6.4. The strategies that allow downstream activities to begin earlier could be applied, on one hand, to increase the value of *OC* by identifying additional pairs of overlappable activities, and, on the other hand, to increase the value of *MO* by allowing earlier transfer of information. In addition, these strategies can be specifically targeted towards the identification of additional overlapping pairs on the critical path in order to increase the value of *PC*. Also, applying strategies to accelerate the evolution of the information and to reduce the sensitivity of the downstream activity could be useful to reduce the value of *RR*. Even though these strategies do not include any action concerning the resource constraints, the results in section 6.4 show that the scarcity of the resource has a major influence on the makespan gain. For this reason, we recommend the allocation of additional resource capacities (i.e., increase *RS*) in order to allow more tasks to be performed in parallel and to allow the execution of reworks.

8 Concluding remarks

The main contribution of the paper is to quantify and analyse the influence of project characteristics on the reduction of the project makespan in projects with complex networks, resource constraints, overlapping, and rework. The reduction of the project makespan is obtained by solving the project scheduling problem with and without overlapping. Two methods have been developed for solving the problem with overlapping. A 0-1 integer linear programming model with overlapping modes, which is solved using CPLEX, and a metaheuristic based on a scatter search algorithm, initially developed for the standard RCPSP. The first procedure is able to find the optimal makespan gain for 98% of the project instances with 30 activities. Comparison of the two methods shows that the metaheuristic produces high-quality solutions in reasonable computational time.

The first finding is that no reduction of the makespan is observed in about 25% of the projects of the benchmark. Overlapping in these projects is not only useless, but it will also cause additional workload and costs. A two-part model is used to conduct a statistical analysis to measure the effect of project characteristics on the makespan gain. The results reveal that the proportion of pairs of overlappable activities on the critical path and the scarcity of the resource constraints have the highest influence on the makespan gain. The two-part model could be used as a predictive model to evaluate the need for overlapping and to quantify the makespan reduction of a project.

The results also suggest strategies adding resource capacities and begin the downstream activities earlier should be emphasized. Furthermore, the best overlapping decision should consist in overlapping only few pairs of overlappable activities with a large degree of overlapping. Even though the activities on the critical path are more likely to be overlapped, overlapping decisions should not rely solely on the criticality of the activities. Therefore, a first extension of this study should be to examine other characteristics, such as the consumption of resources, in order to assist project managers and planners to choose the most appropriate activities to be overlapped.

Another direction worth pursuing involves the estimation of the overlapping data. Some methods exist in the literature when organizations have experience with similar projects (Lin et al., 2010; Krishnan et al., 1997; Roemer et al., 2000; Loch and Terwiesch, 1998; Grèze et al., 2011, 2014; Lin et al., 2009), but the problem of how to reliably estimate these data for new projects should also be investigated. In this regard, our predictive model could be used to evaluate the robustness of the makespan gain and the stability of the activity execution times in response to an unreliable estimation of the overlapping data. Finally, the scheduling problem is formulated in a deterministic environment and does not directly address schedule risks. The mathematical model may thus be extended to introduce randomness, feedbacks and iterations. As some of the most advanced approaches in the literature for stochastic scheduling, such as proactive and reactive techniques, involve determining a baseline schedule without anticipation of uncertainty (Demeulemeester et al., 2008; Guéret and Jussien, 2008; Herroelen and Leus, 2004, 2005), the formulation proposed in this paper constitutes a first step towards the development of approaches for stochastic scheduling of projects with overlapping.

References

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

Baydoun, G., Haït, A., Pellerin, R., Clément, B., & Bouvignies, G. (2016). A rough-cut capacity planning model with overlapping. OR Spectrum, 38(2): 335-364.

Berthaut, F., Grèze, L., Pellerin, R., Perrier, N., & Hajji, A. (2011). Optimal resource-constraint project scheduling with overlapping modes. Proceedings of the 4th International Conference on Industrial Engineering and Systems Management, Metz, France, May 25-27, I[4]e[2] Institute, Belgium, 299-308.

Berthaut, F., Pellerin, R., Perrier, N., & Hajji, A. (2014). Time-cost trade-offs in resource-constraint project scheduling problems with overlapping modes. International Journal of Project Organisation and Management, 6(3): 215-236.

Berthaut, F., Pellerin, R., Hajji, A., & Perrier, N. (2018). A path relinking-based scatter search for the resource-constrained project scheduling problem. International Journal of Project Organisation and Management, 10(1): 1-36.

Blacud, N.A., Bogus, S.M., Diekmann, J.E., & Molenaar, K.R (2009). Sensitivity of Construction Activities under Design Uncertainty. Journal of Construction Engineering and Management, 135(3): 199-206.

Bogus, S.M., Molenaar, K.R., & Diekmann, J.E. (2006). Strategies for overlapping dependent design activities. Construction Management and Economics, 24(8): 829-837.

Bozejko, W., Hejducki, Z., Uchroski, M., & Wodecki, M. (2014). Solving resource-constrained construction scheduling problems with overlaps by metaheuristic. Journal of Civil Engineering and Management, 20(5): 649-659.

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., & Knust, S. (2000). A linear programming and constraint propagation-based lower bound for the RCPSP. European Journal of Operational Research, 127(2): 355-362.

Brucker, P., & Knust, S. (2012). Complex Scheduling. Berlin, Germany: Springer.

Brucker, P., & Knust, S. (2003). Lower bounds for resource-constrained project scheduling problems. European Journal of Operational Research, 149(2): 302-313.

Brucker, P., Knust, S., Schoo, A., & Thiele, O. (1998). Branch and bound algorithm for the resource-constrained project scheduling problem. European Journal of Operational Research, 107(2): 272-288.

Carlier, J., & Pinson, E. (1989). An algorithm for solving the job-shop problem. Management Science, 35(2): 164-176.

Chen, W., Shi, Y.-J., Teng, H.-F., Lan, X.-P., & Hu, L.-C. (2010). An efficient hybrid algorithm for resource-constrained project scheduling. Information Sciences, 180(6): 1031-1039.

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.

Debels, D., De Reyck, B., Leus, R., & Vanhoucke, M. (2006). A hybrid scatter search/electromagnetism metaheuristic for project scheduling. European Journal of Operational Research, 169(2): 638-653.

Demassey, S., Artigues, C., & Michelon, P. (2005). Constraint-propagation-based cutting planes: an application to the resource-constrained project scheduling problem. INFORMS Journal on Computing, 17(1): 52-65.

Demeulemeester, E., Herroelen, W., & Leus, R. (2008). Proactive-reactive project scheduling. In: Resource-Constrained Project Scheduling: Models, Algorithms and Applications, Artigues C., Demassey S. & Néron E. (Eds.), Wiley, London, UK, 203-212.

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

Dzeng, R.-J. (2006). Identifying a design management package to support concurrent design in building wafer fabrication facilities. Journal of Construction Engineering and Management, 132(6): 606-614.

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.

Grèze, L., Pellerin, R., & Leclaire, P. (2011). Processus d'accélération de projets sous contraintes de ressources avec modes de chevauchement. Paper presented at the Conférence Internationale de Génie Industriel (CIGI 2011), St-Sauveur, Canada, 12-14 October.

Grèze, L., Pellerin, R., Leclaire, P., & Perrier, N. (2014a). Evaluating the effectiveness of task overlapping as a risk response strategy in engineering projects. International Journal of Project Organisation and Management, 6(1-2): 33-47.

Grèze, L., Pellerin, R., Leclaire, P., & Perrier, N. (2014b). CIGI2011: A heuristic method for resource-constrained project scheduling with activity overlapping. Journal of Intelligent Manufacturing, 25(4): 787-811.

Guéret, C., & Jussien, N. (2008). Reactive approaches. In Resource-Constrained Project Scheduling: Models, Algorithms and Applications, Artigues C., Demassey S. & Néron, E. (Eds.), Wiley, London, UK, 191-201.

Hartmann, S. (1999). Project Scheduling under Limited Resources: Models, Methods, and Applications. New York, USA, Berlin, Heidelberg: Springer-Verlag.

Hartmann, S., & Kolisch, R. (2000). Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem. European Journal of Operational Research, 127(2): 394-407.

Herroelen, W., & Leus, R. (2004). Robust and reactive project scheduling: a review and classification of procedures. International Journal of Production Research, 42(8): 1599-1620.

Herroelen, W., & Leus, R. (2005). Project scheduling under uncertainty: Survey and research potentials. European Journal of Operational Research, 165(2): 289-306.

Hosmer, D.W., & Lemeshow, S. (2000). Applied Logistic Regression. New York, USA: Wiley-Interscience.

Huang, E., & Chen, S.-J. (2006). Estimation of project completion time and factors analysis for concurrent engineering project management: A simulation approach. Concurrent Engineering Research and Applications, 14(4): 329-341.

Imai, K.-I., Ikujiro, N., & Hirotaka, T. (1985). Managing the new product development process: how japanese companies learn and unlearn. In: The Uneasy Alliance: Managing the Productivity-Technology Dilemma, Hayes R., Clark K. & Lorenz C. (Eds.), Harvard Business School Press, Boston, MA, USA, 337-375.

Khoueiry, Y., Srour, I., & Yassine, A. (2013). An optimization-based model for maximizing the benefits of fast-track construction activities. Journal of the Operational Research Society, 64(8): 1137-1146.

Klein, R., & Scholl, A. (1999). Computing lower bounds by destructive improvement: An application to resource-constrained project scheduling. European Journal of Operational Research, 112(2): 322-346.

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., Schwindt, C., & Sprecher, A. (1999). Benchmark instances for project scheduling problems. In: Project Scheduling, Weglarz J. (Ed.), Springer Science+Business Media, New York, USA, 197-212.

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.

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

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.

Lim, T.-K., Yi, C.-Y., Lee, D.-E., & Arditi, D. (2014). Concurrent construction scheduling simulation algorithm. Computer-Aided Civil and Infrastructure Engineering, 29(6): 449-463.

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.

McCullagh, P., & Nelder, J.A. (1989). Generalized Linear Models. London, UK: Chapman and Hall.

Mobini, M., Rabbani, M., Amalnik, M.S., Razmi, J., & Rahimi-Vahed, A.R. (2009). Using an enhanced scatter search algorithm for a resource-constrained project scheduling problem. Soft Computing, 13(6): 597-610.

Paraskevopoulos, D.C., Tarantilis, C.D., & Ioannou, G. (2012). Solving project scheduling problems with resource constraints via an event list-based evolutionary algorithm. Expert Systems with Applications, 39(4): 3983-3994.

Paul, P., Pennell, M.L., & Lemeshow, S. (2013). Standardizing the power of the Hosmer–Lemeshow goodness of fit test in large data sets. Statistics in Medicine, 32(1): 67-80.

Pena-Mora, F., & Li, M. (2001). Dynamic planning and control methodology for design/build fast-track construction projects. Journal of Construction Engineering and Management, 127(1): 1-17.

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

Ranjbar, M., De Reyck, B., & Kianfar, F. (2009). A hybrid scatter search for the discrete time/resource trade-off problem in project scheduling. European Journal of Operational Research, 193(1): 35-48.

Roemer, T.A., & Ahmadi, R. (2004). Concurrent crashing and overlapping in product development. Operations Research, 52(4): 606-622.

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

Sprecher, A., Hartmann, S., & Drexl, A. (1997). An exact algorithm for project scheduling with multiple modes. OR Spectrum, 19(3): 195-203.

Talbi, E. G. (2009). Metaheuristics: from design to implementation (Vol. 74). Hobboken, NJ, USA: John Wiley & Sons.

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

Tyagi, S.K., Yang, K., & Verma, A. (2013). Non-discrete ant colony optimisation (NdACO) to optimise the development cycle time and cost in overlapped product development. International Journal of Production Research, 51(2): 346-361.

Valls, V., Ballestin, F., & Quintanilla, S. (2004). A population-based approach to the resource-constrained project scheduling problem. Annals of Operations Research, 131(1): 305-324.

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

Wooldridge, J.M. (2002). Econometric Analysis of Cross Section and Panel Data. Cambridge, MA, USA: MIT Press.

AUTHORS

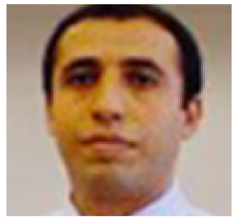
Francois Berthaut

Berthaut received the B.Sc. degree in industrial and mechanical engineering from Arts et Métiers ParisTech, Paris, France, and the M.Sc. degree and the Ph.D. degree in industrial engineering from École Polytechnique de Montréal, Montreal, Canada. His is currently working as an advisor on modelling electricity production at Hydro-Québec, Montreal, Canada. His current research interests include project scheduling and management, and manufacturing systems modelling and control.

Robert Pellerin

Pellerin is Full Professor in the Department of Mathematics and Industrial Engineering at Ecole Polytechnique de Montreal. He holds degrees in engineering management (B.Eng.) and industrial engineering (Ph.D.). He has practiced for more than 12 years in project management and enterprise resource planning (ERP) systems implementation in the aerospace and defense industry. He is also a certified professional in Operations Management (CPIM) and Project Management (PMP). His current research interests include project management and enterprise system implementation and integration. He is the current chairman of the Jarislowsky/SNC-Lavalin Research Chair in the management of international projects and he is a member of the CIRRELT research group.

Adnène Hajji



Hajji received his undergraduate degree in Mechanical Engineering from École Nationale d'Ingénieurs de Tunis, Tunisia (1999) and his M.Eng. (2003) and Ph.D. (2007) in Automated Production Engineering, both from École de Technologie Supérieure, Montréal. After two years as Postdoctoral Researcher at Ecole Polytechnique de Montréal, in 2010 he joined Laval University, where he is full professor at department of operations and decision systems (Faculty of Business Administration). He gained practical and research experience through several collaborations and applied research projects. His main research interest is manufacturing systems modelling, and control, simulation as well as integrated reactive models in ERP systems. He is a member of the Interuniversity Research Centre on Enterprise Networks, Logistics, and Transportation (CIRRELT) and Jarislowsky/SNC-Lavalin Research Chair in the management of international projects.

Nathalie Perrier

Perrier is Research Associate at the Department of Mathematics and Industrial Engineering of École Polytechnique de Montréal (Canada). She received her Ph.D. in Mathematics for Engineers from École Polytechnique de Montréal. Since September 2010, she is Research Associate for the Jarislowsky/SNC-Lavalin Research Chair in the management of international projects. Her research interests include optimization of transportation systems, logistics for emergency response, optimization of winter road maintenance operations, and management of international projects. She is a member of the Interuniversity Research Centre on Enterprise Networks, Logistics, and Transportation (CIRRELT).