A computational experiment to explore better robustness measures for project scheduling under two types of uncertain environments

Zhiqiang Ma[a,b], Erik Demeulemeester[c], Zhengwen He[a,b,*], Nengmin Wang[a,b]

[a] *School of Management, Xi'an Jiaotong University, Xi'an 710049, P.R. China*

[b] *ERC for Process Mining of Manufacturing Services in Shaanxi Province, Xi'an 710049, P.R. China*

[c] *Research Center for Operations Management, Faculty of Economics and Business, KU Leuven, Leuven 3000, Belgium*

[*] *Corresponding author. E-mail: zhengwenhe@mail.xjtu.edu.cn*

Acknowledgements

This research was supported by the National Natural Science Foundation of China under Grants No. 71871176, 71371150, 71572138, 71732006, 71390331 and 71731009 and China Postdoctoral Science Foundation under Grant No. 2018M643685. The authors would like to acknowledge the China Scholarship Council for the financial support under Grant No. 201606280246 and the Research Center for Operations Management of KU Leuven for providing a visiting research position for the first author.

Highlights

- ✓ A general framework of the slack-based surrogate robustness measures is proposed
- ✓ The best surrogate measures are found for two kinds of uncertain environments
- ✓ The improvement is 13.79% under the environment of stochastic activity durations
- ✓ The impact of parameters on the performance of the robustness measure is analyzed
- ✓ The time buffering strategies are different under the two uncertain environments

A computational experiment to explore better robustness measures for project scheduling under two types of uncertain environments

Abstract This paper addresses the proactive resource-constrained project scheduling problem, aiming to explore better surrogate robustness measures for project managers that want to generate robust baseline schedules under uncertain environments. The contribution of this paper is threefold. First, we propose a general framework of slack-based surrogate robustness measures and introduce three parameters to distinguish different alternative calculations of the measure. A computational experiment based on reactive simulation is constructed where the performance of the surrogate measures is evaluated by the reactive cost and two types of uncertain environments, i.e. stochastic resource availabilities and stochastic activity durations, are taken into account. Second, we analyze the impact of the three parameters as well as the two uncertain environments on the surrogate robustness measures and find the best measures for different situations. The proposed surrogate robustness measures are shown to be effective. Compared with benchmark measures, the improvements are respectively 2.67% and 13.79% under the two uncertain environments. Third, we investigate the difference of buffering strategies between the two uncertain environments. For the environment of stochastic resource availabilities, it turns out to be better to have a uniform distribution of time buffers throughout the schedule, while the reverse is true for the environment of stochastic activity durations.

Keywords Proactive project scheduling; Solution robustness; Surrogate robustness measure; Stochastic resource availabilities; Stochastic activity durations

1. Introduction

It is a well-known fact that project activities are subject to considerable uncertainties, which may lead to numerous schedule disruptions during project execution. Accordingly, proactive project scheduling has been the subject of many research efforts, aiming to generate robust baseline schedules that are protected against disruptions that may occur during project execution (Leus, 2003). Proactive scheduling plays an important role in robust project scheduling and has gained much attention during the last two decades (Herroelen and Leus, 2004a; Herroelen and Leus, 2005; Van de Vonder et al., 2007b; Demeulemeester and Herroelen, 2011). For the proactive scheduling literature in production scheduling under uncertain environments, we refer the interested readers to Aytug et al. (2005), Sabuncuoglu and Goren (2009), Liu et al. (2017) and Cui et al. (2018).

In a scheduling environment, many kinds of schedule disruptions may occur (Zhu et al., 2005), such as new activity disruptions, precedence disruptions, activity duration disruptions, activity resource requirement disruptions, resource availability disruptions, and milestone disruptions. Among them, two types of disruptions, i.e., resource availability disruptions and activity duration disruptions, are typically considered in proactive project scheduling. Regarding resource availability disruptions, resources randomly break down, which may result in the infeasibility of the baseline schedule during project execution. Lambrechts et al. (2008a) define this uncertain environment as stochastic resource availabilities and introduce two parameters, i.e., $MTTF_k$ (mean time to failure of resource type k) and $MTTR_k$ (mean time to repair of resource type k), to model resource breakdowns. As for activity duration disruptions, Van de Vonder et al. (2008) define the environment as stochastic activity durations with a right-skewed beta-distribution and distinguish between low, medium and high duration variability.

To deal with the above disruptions, two types of robustness are considered in proactive project scheduling, i.e., quality robustness and solution robustness (Van de Vonder et al., 2005). Quality robustness is defined as the insensitivity of the project makespan to schedule disruptions while solution robustness is defined as the insensitivity of planned activity starting times to schedule disruptions. With regard to quality robustness, we refer to Hazır et al. (2010, 2011) who introduce and solve the robust multi-mode discrete time/cost trade-off problem with exact and heuristic algorithms, and we refer the interested readers to Al-Fawzan and Haouari (2005), Van de Vonder et al. (2006) and Chtourou and Haouari (2008) for the trade-off problem between quality robustness and solution robustness. As for solution robustness, many mathematical models and heuristics are developed to solve the proactive scheduling problem under two types of uncertain environments, i.e., stochastic resource availabilities and stochastic activity durations, which will be summarized next.

About activity duration disruptions, for example, Herroelen and Leus (2004b) propose exact methods to construct solution robust baseline schedules for the case in which only one duration disruption occurs. Leus and Herroelen (2004) develop a branch-and-bound algorithm to solve a resource allocation model that protects a given baseline schedule against activity duration variability. Van de Vonder et al. (2006) develop a heuristic algorithm named RFDFF (resource flow-dependent float factor), while Van de Vonder et al. (2008) develop new heuristic procedures such as the starting time criticality (STC) heuristic to solve the proactive RCPSP. In addition, Schatteman et al. (2008) develop an integrated methodology for planning construction projects under uncertainty, which relies on a computer-supported risk management system for the identification, analysis and quantification of the major risks. Lamas and Demeulemeester (2016) define a new robustness measure that is completely independent of the applied reactive policy and then introduce a branch-and-cut algorithm to solve a sample average approximation of the original problem. When it comes to resource availability disruptions, a direct effective way to improve schedule robustness is to insert resource buffers throughout the baseline schedule (Lambrechts et al., 2008a). However, it is also possible to employ time buffers, that are generally used for protection against activity duration disruptions, to deal with resource availability disruptions (Lambrechts et al., 2008b; Lambrechts et al., 2011).

Typically, solution robustness is measured by the weighted sum of the expected absolute deviations between the start times in the realized schedule and those in the baseline schedule. Because the analytic evaluation is very cumbersome, the scheduling problem for robustness maximization with the above measure is generally solved by simulation, which is computationally demanding. Even though this approach is feasible for small project networks, the computational demands quickly increase with project size to an unpractical level, which is why surrogate measures (such as slack-based functions) are necessary. The slack-based surrogate robustness measure is first proposed by Al-Fawzan and Haouari (2005) in project scheduling who define the free slack of one activity as the total amount of time this activity can be delayed without causing any precedence or resource constraint violations and measure schedule robustness as the sum of free slacks over all activities. Kobylański and Kuchta (2007) discuss a limitation of this surrogate measure and propose to use the minimum of free slacks or the minimum of free slack/duration ratios. However, Hazır et al. (2010) point out that focusing on the minimum values has the weakness that two schedules with the same minimum values may have different slack patterns and the measures proposed by Kobylański and Kuchta (2007) fail to differentiate between these schedules. On the other hand, Chtourou and Haouari (2008) formulate twelve predictive indicators for the robustness maximization purpose, and Lambrechts et al. (2008b) introduce a free slack utility function for each activity with diminishing returns per extra unit of its free slack, which is partly adopted by Hazır et al. (2010) and Lambrechts et al. (2011). To sum up, the above proposed slack-based robustness measures are generally made up of two parts, i.e., the slack utility function of each activity and the instability coefficient for weighting the utilities of different activities. Considering the first part, one important difference among the existing measures is whether the marginal utility of free slacks is set to be decreasing. As for the second part, the important difference is whether to take into account the impact of a delay of one activity on its direct and indirect successors when calculating the instability coefficient.

The performance of the surrogate robustness measures can be evaluated by the simulated reactive cost, which is incurred when the planned starting times of activities have to be adjusted to cope with unexpected disruptions. The lower the reactive cost that is incurred, the better the performance of the surrogate robustness measure. Reactive scheduling aims at finding the best policy to react to schedule disruptions when they render the baseline schedule infeasible, whose typical objective is to generate a feasible reactive schedule that deviates as little as possible from the original baseline schedule, i.e., with the lowest possible reactive cost. With regard to the reactive scheduling that copes with resource availability disruptions, we cite Lambrechts et al. (2008a) and Chakrabortty et al. (2016). With regard to the disruptions related to activity durations, we refer to Van de Vonder et al. (2007a), Deblaere et al. (2007) and Hu et al. (2017). Particularly, we refer the interested readers to Deblaere et al. (2011) who consider the two kinds of disruptions at the same time.

To summarize, many surrogate robustness measures have been proposed, which can measure solution robustness efficiently. However, the effectiveness of these measures is still an open question. Furthermore, we still do not know which kind of robustness measure performs best and what is the most effective surrogate robustness measure for proactive scheduling under uncertain environments. To bridge this gap, in this paper, we focus on solution robustness and propose a general framework of slack-based surrogate robustness measures, which includes two parts, i.e., the slack utility function of each activity and the instability coefficient for weighting the utilities of different activities. In this framework, we introduce three parameters to distinguish different alternative calculations of the surrogate robustness measure. More specifically, the first parameter determines whether the marginal slack utility is decreasing, the second one measures the average impact of a delay of one activity on its direct and indirect successors, and the third parameter reflects the importance of protecting the starting time of the dummy end activity. Using computational experiments based on reactive simulations, the performance of the proposed surrogate robustness measures is evaluated in terms of the reactive cost. In this paper, two types of uncertain environments that are typically investigated in proactive project scheduling, i.e., stochastic resource availabilities and stochastic activity durations, are taken into account. The aim of this study is to analyze the impact of the three parameters as well as the two uncertain environments on the surrogate robustness measures and to determine the best measures for different situations. Then the proposed surrogate robustness measures can be used by project managers to generate robust baseline schedules under uncertain environments. Through a comparison, we can also investigate the difference between the characteristics of the two uncertain environments and find the best buffering strategies for each of them. We believe that the proposed problem, which to the best of our knowledge has not thus far been investigated, is of great significance because it is a fundamental problem in proactive scheduling and is worthwhile to be further investigated. More details about the differences and the similarities between this paper and the literature on proactive project scheduling can be found in Table 1.

The rest of this paper is organized as follows. In Section 2, we define the studied robust scheduling problem, propose different calculations of surrogate robustness measures and explain how to evaluate the performance of the measures. Section 3 is devoted to describing the computational experiment, in which two types of uncertain environments are considered, i.e., stochastic resource availabilities and stochastic activity durations. In Section 4, we analyze the experimental results, from which we try to explore the best robustness measures to cope with the two different environments. Section 5 is devoted to a discussion about the uncertain environment in which both resource availability disruptions and activity duration disruptions occur. Finally, general conclusions and directions for further research are presented in Section 6.

Table 1 Differences and similarities between this paper and existing proactive works

References	Reference type	Objective	Resource availability variability	Activity duration variability	Slack-based surrogate robustness measure	The three parameters in the general framework	Algorithm
Leus (2003)	PhD thesis	Robustness	×	√	×	×	Multiple algorithms
Herroelen and Leus (2004a)	Review	/	/	/	/	/	/
Herroelen and Leus (2004b)	Research	Robustness	×	√	×	×	EWD1
Al-Fawzan and Haouari (2005)	Research	Trade-off	×	√	√	×	Tabu search
Herroelen and Leus (2005)	Review	/	/	/	/	/	/
Van de Vonder et al. (2005)	Research	Trade-off	×	√	×	×	Multiple algorithms
Van de Vonder et al. (2006)	Research	Trade-off	×	√	×	×	RFDFF
Kobylański and Kuchta (2007)	Note	Robustness	×	√	√	×	ILOG OPL
Van de Vonder et al. (2007b)	Review	/	/	/	/	/	/
Chtourou and Haouari (2008)	Research	Trade-off	×	√	√	×	Priority rule based
Lambrechts et al. (2008a)	Research	Robustness	√	×	×	×	Priority rule based
Lambrechts et al. (2008b)	Research	Robustness	√	×	√	×	Tabu search
Schatteman et al. (2008)	Research	Trade-off	×	√	×	×	Integrated method
Van de Vonder et al. (2008)	Research	Robustness	×	√	×	×	Multiple algorithms
Hazır et al. (2010)	Research	Robustness	×	√	√	×	Two-stage algorithm
Demeulemeester and Herroelen (2011)	Book	/	/	/	/	/	/
Hazır et al. (2011)	Research	Time/cost	×	×	×	×	Multiple algorithms
Lamas and Demeulemeester (2016)	Research	Robustness	×	√	×	×	Branch-and-cut
Lambrechts et al. (2011)	Research	Robustness	√	×	√	×	Multiple algorithms
Ma et al. (2019)	Research	Robustness	×	√	√	×	Genetic algorithm
This paper	Research	Robustness	√	√	√	√	Tabu search

2. Problem definition

2.1 Proactive scheduling model

Consider a project represented in activity-on-the-node (AoN) format by means of a digraph $G = (N, A)$, where the set of nodes N represents the activities and the set of arcs A the finish-start, zero-lag precedence relations. The activities are numbered from the dummy start activity 1 to the dummy end activity n, and each activity i has a duration d_i and requires r_{ik} units of resource type k during each period in which it is processed. The project deadline is denoted as D. There are K different resource types with an availability in each period $[t, t+1)$, $(t = 0,1,\cdots,D)$, of R_k units, $k = 1,2,\cdots,K$. Parameter w_i denotes the activity weight of activity i.

For such a project, we aim to generate a robust baseline schedule by solving the presented proactive scheduling model below. The objective function is solution robustness maximization which is measured by formula (1) where s_i^B represents the planned starting time of activity i and $E(s_i^R)$ represents the expected starting time of activity i during project execution. To sum up, solution robustness $Robu$ is measured by the weighted sum of the expected absolute deviations between the start times in the realized schedule and those in the baseline schedule. The reason why an expected value is calculated here for s_i^R is that it is an uncertain environment during project execution where schedule disruptions may occur on resource availabilities R_k or on activity durations d_i, which are respectively corresponding to the two types of uncertain environments, i.e., stochastic resource availabilities and stochastic activity durations. In this proactive scheduling model, d_i is the expected duration of activity i, while R_k represents the resource availability of resource type k without disruptions.

Maximize $Robu = \sum_{i \in N} w_i \left| E(s_i^R) - s_i^B \right|$ (1)

subject to $s_i^B + d_i \le s_j^B \quad \forall (i,j) \in A$ (2)

$\sum_{i \in S_t} r_{ik} \le R_k \quad \forall t, \forall k$ (3)

$s_n^B \le D$ (4)

s_i^B is a non-negative integer (5)

With regard to the constraints, formula (2) describes the precedence constraints, while formula (4) represents the project deadline constraint. As S_t is defined to represent the set of activities that are in progress during time interval $[t, t+1)$, formula (3) enforces the renewable resource constraints, which imply that there does not exist a time period t and a resource type k for which the cumulative resource requirements of the active activities exceed the per-period availability of the considered resource type. Besides, the range of values for s_i^B is given in constraint (5).

2.2 Surrogate robustness measures

As mentioned before, project execution is subject to resource availability disruptions or activity duration disruptions, which results in a cumbersome analytic evaluation of formula (1). For this reason, $E(s_i^R)$ is typically calculated through simulation and it will take much time to calculate the objective function value for each feasible solution. Therefore, it is necessary to use surrogate measures to replace formula (1) in order to improve the efficiency of the calculation. This is just what many researchers did in the literature, such as Al-Fawzan and Haouari (2005), Kobylański and

Kuchta (2007), Chtourou and Haouari (2008) and Lambrechts et al. (2008b).

In this paper, we propose a general framework of the slack-based surrogate robustness measures and intend to use surrogate robustness measures to replace formula (1) in the proactive scheduling model. We formulate the general framework with two parts, i.e., the slack utility function of each activity and the instability coefficient for weighting the utilities of different activities, and introduce three parameters to generate different alternative calculations of the surrogate robustness measure. Concerning the first part of the general framework of surrogate robustness measures, we adopt an exponential function $\sum_{j=1}^{FS_i} \exp(\lambda \cdot j)$ as the slack utility function in which a non-positive parameter λ is introduced and FS_i denotes the amount of free slack of activity i. It can be observed that the utility function for each activity is with diminishing returns per extra unit of free slack that is allocated to that activity, i.e., the utility of the latter slack unit of one activity is $e^{\lambda}(e^{\lambda} \leq 1)$ times that of the former one. For example, if λ is set to be negative one and activity 3 has a free slack of two units, i.e., $\lambda = -1$ and $FS_3 = 2$, then the utility of the first slack unit equals $e^{-1} = 0.37$ while the utility of the second one is only $e^{-2} = 0.14$. This indicates that the first slack is more beneficial to schedule robustness, which is just corresponding to the practice in project management.

Simply maximizing the sum of slack utilities as a surrogate robustness measure would assume the contribution of free slack values to schedule robustness to be equivalent for each activity whereas our real objective function consists of a weighted sum. Let W_i denote the instability coefficient for weighting the utility of activity i, then schedule robustness $Robu$ can be formulated as follows to replace formula (1):

$$Robu = \sum_{i=1}^{n} W_i \sum_{j=1}^{FS_i} \exp(\lambda \cdot j) \tag{6}$$

With respect to the calculation of the instability coefficients W_i, we employ S_i^* to denote the set of direct and indirect successors of activity i and use a parameter $\varphi_i (0 \leq \varphi_i \leq 1)$ to represent the average percentage of activities in S_i^* that are expected to be postponed due to the delay of activity i. In other words, the parameter φ_i represents the average impact of the delay of activity i on its direct and indirect successors. Taking this average impact into account, one way to calculate W_i is as follows:

$$W_i = w_i + \varphi_i \sum_{j \in S_i^*} w_j, \quad 0 \leq \varphi_i \leq 1. \tag{7}$$

As the free slacks after the activities serve to protect the starting times of their successors, it seems more reasonable to exclude w_i in the calculation of W_i. Besides, it seems more reasonable to increase the relative instability coefficients of the predecessors of the dummy end activity so that enough project buffers can be added to protect the dummy end activity. Based on the above analysis, we now remove w_i from formula (7) and set φ_i as 1 for each direct predecessor of the dummy end activity. This indicates another way of calculating W_i:

$$W_i = \varphi_i \sum_{j \in S_i^*} w_j, \quad 0 < \varphi_i \leq 1 \text{ for } (i,n) \notin A; \varphi_i = 1 \text{ for } (i,n) \in A \tag{8}$$

Table 2 Surrogate robustness measures

$version$	W_i	φ_i		λ	
		level(φ_i)	value(φ_i) [q]	level(λ)	value(λ)
1	$W_i = w_i + \varphi_i \sum_{j \in S_i^*} w_j$, $0 \leq \varphi_i \leq 1$	1	0	1	-2
		2	[0.1,0.3]	2	-1
		3	[0.4,0.6]	3	-1/2
		4	[0.7,0.9]	4	-1/4
		5	1	5	0
2	$W_i = \varphi_i \sum_{j \in S_i^*} w_j$, $0 < \varphi_i \leq 1$ for $(i,n) \notin A$, $\varphi_i = 1$ for $(i,n) \in A$	/	/	1	-2
		2	[0.1,0.3]	2	-1
		3	[0.4,0.6]	3	-1/2
		4	[0.7,0.9]	4	-1/4
		5	1	5	0

[q] For levels 2, 3 and 4, φ_i will be randomly generated from the intervals.

Because this is a new way of calculating W_i, we introduce a parameter $version$ to distinguish it from the original calculation. Accordingly, there are two levels of parameter $version$ in total. With regard to parameters λ and φ_i, we generate each of them in five levels. Note that the combination of the three parameters is just one specific robustness measure, and there are 5*5+4*5=45 different calculations of the robustness measure, just as Table 2 shows. For the sake of description, we choose to use the format RMabc to represent the robustness measure RM with $version = a$, level(φ_i) = b and level(λ) = c. It can be observed that the measure RM152 is just the one proposed by Lambrechts et al. (2008b), which can serve as a benchmark to test the effectiveness of the robustness measures.

2.3 Evaluation of the surrogate robustness measures

The performance of the surrogate robustness measures can be evaluated by the reactive cost, which is obtained through solving the reactive scheduling model. Since there are two kinds of uncertain environments, i.e., stochastic resource availabilities and stochastic activity durations, two reactive scheduling models are presented to calculate the reactive cost under each uncertain environment, just as Table 3 shows. Note that we consider resource availability disruptions and activity duration disruptions separately, which means that in each environment only one kind of disruptions is considered. For the sake of description, in the following, we use E1 to represent the environment of stochastic resource availabilities and E2 to denote the environment of stochastic activity durations. The uncertain environment in which both resource availability disruptions and activity duration disruptions occur will be discussed in Section 5.

Table 3 The reactive scheduling models under two types of uncertain environments

E1 (stochastic resource availabilities)			E2 (stochastic activity durations)		
Minimize	$Loss = \sum_{i \in N} w_i (s_i^R - s_i^B)$	(9)	Minimize	$Loss = \sum_{i \in N} w_i (s_i^R - s_i^B)$	(14)
subject to	$s_i^R + d_i \leq s_j^R \quad \forall (i,j) \in A$	(10)	subject to	$s_i^R + d_i' \leq s_j^R \quad \forall (i,j) \in A$	(15)
	$\sum_{i \in S_t} r_{ik} \leq R_{kt}' \quad \forall t, \forall k$	(11)		$\sum_{i \in S_t} r_{ik} \leq R_k \quad \forall t, \forall k$	(16)
	$s_i^B \leq s_i^R \quad \forall i$	(12)		$s_i^B \leq s_i^R \quad \forall i$	(17)
	s_i^R is a non-negative integer	(13)		s_i^R is a non-negative integer	(18)

In the reactive scheduling models, the minimized reactive cost $Loss$, shown in formulas (9) and (14), is used to evaluate the performance of the robustness measures. The lower the reactive cost that is incurred, the better the surrogate robustness measure. In reactive scheduling, referring to Lambrechts et al. (2008b) and Van de Vonder et al. (2008), we assume that an activity can never start before its baseline starting time, just as constraints (12) and (17) describe. This means that we choose railroad scheduling rather than the roadrunner mentality for reactive simulations. The reason is that implementing the roadrunner mentality during project execution implies that the inserted time buffers are disregarded, which is likely to result in a serious deviation from the baseline schedule. In contrast, railroad scheduling is more beneficial to reduce the reactive costs since it can make good use of inserted buffers to cope with disruptions. The range of values for s_i^R is given in constraints (13) and (18). We can find that the two reactive scheduling models are different in the precedence and resource constraints, which are related to the two types of schedule disruptions. Under the environment E1, R'_{kt} represents the stochastic resource availability of resource type k at time t and formula (11) describes the resource constraints during project execution. On the other hand, under the environment E2, d'_i denotes the stochastic activity duration of activity i and formula (15) presents the precedence constraints during project execution. It is noteworthy that in reactive scheduling models, R'_{kt} and d'_i are stochastic variables, but for each simulation the values of them are generated and therefore known in advance.

3. Experimental design

The aims of the computational experiment are to investigate the performance of different surrogate robustness measures and to explore the best surrogate measure for uncertain environments of stochastic resource availabilities and stochastic activity durations. In this section, more details about the computational experiment, such as the tested instance sets, the data generation, the proactive scheduling heuristic and the reactive policy, will be presented.

Concerning the instance sets, we use the 480 30-activity RCPSP instances of the well-known PSPLIB (Kolisch and Sprecher, 1996), which includes the information of the project network, the activity durations, the activity resource requirements and the resource availabilities. With regard to activity weight w_i for all non-dummy activities, we refer to Van de Vonder et al. (2008) and generate them from a discrete, triangularly shaped distribution between 1 and 10 with $P(w_i = x) = 0.21 - 0.02x$. This setting corresponds to what is expected in real-life projects, namely that most activities will have a low activity weight whereas only a minority are heavily penalized for being started later than planned. As it is really important to meet the project due date, the activity weight of the dummy end activity is set to be 10 times the average of the activity weight distribution function, which is 3.85 for $P(w_i = x)$. In our experiment, the project due date D of each instance is set at $C_{\max}^{\text{RCPSP}}(1+\alpha)$ where $C_{\max}^{\text{RCPSP}}$ represents the optimal makespan that is solved by CPLEX under a deterministic environment, and the due date factor α is a parameter that is chosen by the project manager and constitutes the trade-off between project stability and project duration. In this paper, we have three levels of α, i.e., 10%, 20% and 30%.

With regard to the schedule disruptions, we generate stochastic resource availabilities and stochastic activity durations as suggested in existing studies (Lambrechts et al., 2008a; Van de Vonder et al., 2008). Specifically, with respect to stochastic resource availabilities, two parameters, i.e., $MTTR_k$ and $MTTF_k$, are used to model resource breakdowns. The $MTTR_k$ values are drawn from a uniform discrete distribution between 1 and 5, while the values of $MTTF_k$ are drawn from a uniform discrete distribution between 50% and 150% of the optimal makespan $C_{\max}^{\text{RCPSP}}$ of the project. In the simulation, with five levels of $MTTR_k$ and thirty levels of $MTTF_k$, we draw different resource availabilities from the availability probability function and simulate 5*30=150 times for each problem. As for stochastic activity durations, we generate activity durations from a right-skewed beta-distribution with parameters 2 and 5. Parameter σ is introduced to represent activity duration variability and we distinguish between low, medium and high duration variability. More details of the variability setting are shown in Table 4. Similarly, for each problem, 150 executions (50 ones for each duration variability) are simulated by drawing different activity durations from the described distribution functions.

Table 4 Variability settings of stochastic activity durations

	Level	Minimum duration	Mean duration	Maximum duration
	low	$0.75E(d_i)$	$E(d_i)$	$1.625E(d_i)$
Variability σ	medium	$0.5E(d_i)$	$E(d_i)$	$2.25E(d_i)$
	high	$0.25E(d_i)$	$E(d_i)$	$2.875E(d_i)$

With respect to the proactive scheduling, we use the surrogate robustness measure proposed in Section 2.2 as the objective function for maximization. This proactive scheduling problem has been proven to be NP-hard in the strong sense (Ma et al., 2019), which makes the achievement of optimal solutions a computationally difficult proposition, especially for large projects. For this reason, we apply a heuristic algorithm, the tabu search algorithm proposed by Lambrechts et al. (2008b), to solve the problem to generate a robust schedule. In this algorithm, a solution is represented by means of a priority activity list coupled with a buffer list. Accordingly, two types of neighborhoods, i.e., an activity list neighborhood and a buffer list neighborhood, are defined. As for the algorithm structure, we first consider n_{I} iterations of activity list move (type I), then n_{II} iterations of buffer list move (type II). When the set of $n_{\text{I}} + n_{\text{II}}$ iterations is finished, we start again with an iteration of type I. In the process of iterative search, two types of tabu lists are introduced to respectively store two types of moves that are forbidden for a certain number Num of iterations. The algorithm is terminated after having been executed for a preset time period $t_{\max}$. In this paper, we keep all the parameter settings unchanged, i.e., $n_{\mathrm{I}} = n_{\mathrm{II}} = 1$ and $Num = 32$, but increase $t_{\max}$ from 10 seconds to 15 seconds to obtain a more robust solution.

With respect to the reactive scheduling, for the environment of stochastic resource availabilities, we apply the tabu search based improvement heuristic on the scheduled order priority list. This procedure is proposed by Lambrechts et al. (2008a) who try to improve the starting solution by iteratively executing the best precedence feasible adjacent interchange of two activities in the priority list until a maximum number of iterations $MAXITER$ is reached. The difference is that we set $MAXITER$ as 1000 instead of 50 in order to generate a better reactive schedule. As for the reactive scheduling under the environment of stochastic activity durations, we refer to Van de Vonder et al. (2008) and construct the reactive schedule by applying a parallel schedule generation scheme to a predefined activity list based on the actual activity durations. The predefined activity list is deduced from the baseline schedule by ordering the activities in increasing order of their starting time (tiebreakers are the highest activity weight and the lowest activity number). Similarly to the reactive policy under stochastic resource availabilities, we further apply a tabu search based improvement heuristic on the predefined activity list and set $MAXITER$ to be 1000 as the stopping criterion.

In our experiment, the above proactive and reactive algorithms are programmed in the C++ language, implemented in Microsoft Visual Studio 2013 and executed on a DELL OptiPlex 3040MT with 3.20 GHz clock-pulse and 8G RAM.

4. Experimental results

In this section, we present the obtained experimental results, from which we not only obtain the best robustness measures for the two uncertain environments, but also investigate the impact of parameters on the performance of the surrogate robustness measures.

4.1 The performance of the robustness measures under the two uncertain environments

In order to detect significant differences between the performance of the robustness measures, we first order the robustness measures in the non-decreasing order of their average reactive cost $Loss$, and then, perform for each measure and the next best one a pairwise comparison with the

Table 5 Performance of the robustness measures under two uncertain environments

Environment	Order	Measure	*Loss*	Order	Measure [q]	*Loss*	Order	Measure	*Loss*
E1	1	RM241	157.32	16	RM242	161.62	31	RM244	165.53
	2	RM131	158.00	**17**	**RM152**	**161.63**	32	RM234	165.84
	3	RM251	158.63	18	RM253	161.63	33	RM223	166.33
	4	RM121	158.71	19	RM233	161.72	34	RM113	167.92*
	5	RM232	158.97	20	RM143	162.43	35	RM114	170.73*
	6	RM141	159.00	21	RM124	162.46	36	RM224	173.58*
	7	RM231	159.59	22	RM153	162.49	37	RM155	175.93
	8	RM111	159.61	23	RM112	162.79	38	RM255	177.14
	9	RM142	160.27	24	RM144	163.11	39	RM145	178.67
	10	RM132	160.31	25	RM122	163.13*	40	RM245	180.08
	11	RM252	160.45	26	RM222	163.75	41	RM135	182.23*
	12	RM133	160.94	27	RM123	163.80	42	RM125	185.68*
	13	RM151	161.22	28	RM134	164.53	43	RM235	194.42*
	14	RM243	161.35	29	RM154	164.94	44	RM115	195.51*
	15	RM221	161.39	30	RM254	165.22	45	RM225	239.85
E2	1	RM224	212.81*	16	RM241	243.52	31	RM133	251.16
	2	RM225	219.47	17	RM121	244.39	32	RM153	251.22
	3	RM223	220.43*	18	RM141	244.76	33	RM244	251.29
	4	RM222	227.22*	19	RM132	246.10	34	RM253	252.80
	5	RM221	231.98*	20	RM151	246.13	35	RM144	252.86
	6	RM233	236.27	21	RM131	246.14	36	RM134	252.96
	7	RM113	236.35	22	RM251	246.19	37	RM154	254.16
	8	RM112	236.66	**23**	**RM152**	**246.85**	38	RM115	254.32
	9	RM234	237.03	24	RM252	247.72	39	RM254	255.55*
	10	RM111	237.73	25	RM142	248.05	40	RM125	262.00
	11	RM114	238.32	26	RM123	248.32	41	RM245	263.11*
	12	RM232	238.72	27	RM243	248.91	42	RM135	268.47
	13	RM231	239.84*	28	RM143	249.44	43	RM145	268.74*
	14	RM242	242.81	29	RM235	249.60	44	RM255	273.02
	15	RM122	243.32	30	RM124	250.84	45	RM155	273.13

[q] The benchmark robustness measure RM152 is given in bold.

Wilcoxon signed-rank test using SPSS. A significant difference in performance at the 5% confidence level will be marked with a star (*). From the results shown in Table 5, the following phenomena can be observed.

Under the environment of stochastic resource availabilities, the best robustness measure is RM241 with the lowest reactive cost of 157.32. This means that the robustness measure with $version = 2$, $\varphi_i \in [0.7,0.9]$ and $\lambda = -2$ performs the best. Under the environment of stochastic activity durations, the best robustness measure is RM224 with the lowest reactive cost of 212.81. This means that the robustness measure with $version = 2$, $\varphi_i \in [0.1,0.3]$ and $\lambda = -1/4$ performs the best. It will be discussed in Section 4.2 why RM241 and RM224 perform the best respectively for the two uncertain environments.

Compared with the benchmark robustness measure RM152, the improvements are respectively 2.67% and 13.79% under the two uncertain environments. It can be observed that there is a big improvement under the environment E2, which indicates the effectiveness of the new proposed robustness measure. As for the environment E1, although the reactive cost decreases compared with that of the measure RM152, the improvement is small. This is mainly due to the fact that time buffers have a limited protection for activities against resource availability disruptions. The mechanism of time buffers is the inclusion of slack time in front of activities, in order to absorb potential disruptions caused by earlier resource breakdowns and the resulting activity shifts. However, even though enough time buffers are added in front of activities, the activities will still be delayed if resources break down at exactly their planned starting times. Therefore, it is reasonable to find a small improvement under the environment of stochastic resource availabilities. For this uncertain environment, resource buffers might be more beneficial to improve the schedule robustness (Lambrechts et al., 2008a; Lambrechts et al., 2008b). As the project is planned using a resource availability R_k^* that is lower than the deterministic availability R_k, a breakdown of one or more resource units at the starting times of activities will not always lead to a disruption of the schedule.

4.2 The impact of parameters on the performance of the robustness measure

In this part, we perform an analysis of the parameters $version$, φ_i and λ on the performance of the robustness measure, which will explain why RM241 and RM224 are respectively the best robustness measures for the two uncertain environments. We apply SPSS to conduct the multivariate regression analysis and obtain the results as presented in Table 6. The table provides for each uncertain environment the resulting R^2-value, the constant (Const) and the coefficient of each parameter. A value of "0" indicates that the coefficient is not significant at the 5% confidence level, while a star (*) implies the coefficient to be significant at the 1% confidence level. From the table, we can find that the impact of the parameters is different under the two uncertain environments. For this reason, we will analyze the results for each environment separately.

Table 6 Results of the multivariate regression of the parameters related to the robustness measure

Environment	R^2	Const	$version$	φ_i	λ
E1	0.436	148.171*	0	0	6.613*
E2	0.603	233.243*	-10.984*	5.604*	3.675*

4.2.1 The impact of the parameters under the environment E1

The resulting R^2-value is 0.436, and only the coefficient of parameter λ, with a value of 6.613, is significant at the 5% confidence level. This indicates that the parameter λ has a big impact on the performance of the robustness measure. With a decrease of parameter λ, the reactive cost of the robustness measure is expected to decrease, which means the robustness measure will perform better. This result is not surprising because under the environment of stochastic resource availabilities, even though there are enough time buffers in front of one activity, the activity will still be delayed if resources break down at its planned starting time. For this reason, it may not be wise to add many time buffers in front of one specific activity. Instead, it is better to have a uniform distribution of time buffers throughout the schedule. The smaller the value of parameter λ is, the faster the marginal slack utility decreases, and then a more uniform distribution of time buffers will result. In other words, the robustness measure performs better with a decrease of the value of λ, i.e., it is best to have $\lambda = -2$ and it is worst to have $\lambda = 0$.

In a similar vein to this conclusion, two interesting phenomena can also be observed from the results in Table 5. Firstly, the robustness measure with $\lambda = -2$, i.e., in the form of RMab1, always performs better than the benchmark measure RM152. Additionally, there are nine robustness measures in the form of RMab1, of which seven are positioned among the eight best measures. These strongly verify the big impact of parameter λ on the performance of the robustness measure and it is best to have $\lambda = -2$. The second interesting phenomenon is that the robustness measure with $\lambda = 0$, i.e., in the form of RM*ab*5, incurs a much higher reactive cost than the measure with $\lambda = -1/4$, i.e., in the form of RM*ab*4. We average the gaps between the reactive costs of the robustness measures RM*ab*4 and RM*ab*5 and find that the reactive cost on average increases by 14.50% when changing $\lambda = -1/4$ into $\lambda = 0$. This is an important finding, which indicates that the robustness measure with a constant marginal slack utility performs really bad and thus, it is necessary to have a decreasing marginal utility function of free slacks.

4.2.2 The impact of the parameters under the environment E2

The resulting R^2-value is 0.603 and the coefficients of the three parameter are all significant at the 1% confidence level, which indicates that $version$, φ_i and λ all have a big impact on the performance of the robustness measure.

The impact of parameter $\boldsymbol{version}$. The coefficient of this parameter is -10.984, showing that the reactive cost is expected to decrease with an increase of the value of $version$. From the results in Table 5, we can also find that in the first six best robustness measures, the value of $version$ is always 2. Accordingly, we can draw the conclusion that the second way of calculating the instability coefficients is better. This is reasonable because under this setting the starting time of the dummy end activity can be well protected. As the activity weight of the dummy end activity is much bigger than that of the other activities, in the second way of calculating the instability coefficients, the reactive cost is likely to be lower. To demonstrate this more clearly, we introduce three indices as follows.

- ARW: the ratio of the average instability coefficient of the predecessors of the dummy end activity to that of the other non-dummy activities. For example, for a project $G = (N, A)$ where $N = \{1,2,3,4,5\}$ and $A = \{(1,2), (2,3), (3,4), (4,5)\}$, ARW $= 2W_4/(W_2 + W_3)$.
- ATS: the average amount of time the dummy end activity can be shifted forward in the baseline schedule without causing any precedence or resource constraint violations.
- ATD: the average amount of time the dummy end activity is delayed in reactive simulation.

We average the results over different levels of the other parameters and obtain the results shown in Table 7. As the index ARW shows, compared with that of $version = 1$, the average instability coefficient of the predecessors of the dummy end activity under the case of $version = 2$ is much bigger than that of the other non-dummy activities. Accordingly, for maximizing the objective function, there will be more free slacks after the predecessors of the dummy end activity. In other words, there will be more added time buffers in front of the dummy end activity, just as the indicator ATS shows. Because the dummy end activity can be well protected with enough time buffers, it is less likely for it to be delayed. From the result of the index ATD, we observe that the average amount of time the dummy end activity is delayed decreases by 0.247 when changing $version = 1$ into $version = 2$. As a result, the average cost that is incurred by the delay of the dummy end activity will on average decrease by 0.247*38=9.386. In other words, the second way of calculating the instability coefficients is better and it is beneficial to increase the relative instability coefficients of the predecessors of the dummy end activity.

Table 7 The comparison between $version = 1$ and $version = 2$

$version$	ARW	ATS	ATD
1	69.64%	0.602	1.602
2	138.80%	1.453	1.355

The impact of parameter φ_i. The coefficient of parameter φ_i is 5.604, which implies that the bigger the value of φ_i, the higher the resulting reactive cost. In other words, a bigger φ_i has a negative influence on the performance of the robustness measure. From the results in Table 5, we can also find that in the first five best robustness measures, the level of φ_i is always the lowest, i.e., $\varphi_i \in [0.1, 0.3]$. This can be explained as follows. Under the environment of stochastic activity durations, resource availabilities are constant and the activity durations are uncertain. If there are enough time buffers in front of one activity, the activity will be well protected so that it is less likely to be postponed due to the delay of the direct and indirect predecessors of the activity. For maximizing the objective function, the starting times of the activities with high activity weights should be first well protected by adding enough time buffers in front of them. In other words, it is a good choice to add time buffers in front of specific activities and the distribution of time buffers will be less uniform. To achieve this, the distribution of the instability coefficients of activities should not be uniform as well, which requires the parameter φ_i to be small. Therefore, the performance of the robustness measure becomes better with a decrease of the value of φ_i, i.e., it is best to have $\varphi_i = [0.1,0.3]$ under the environment of stochastic activity durations.

The impact of parameter $\boldsymbol{\lambda}$. The coefficient of parameter λ is 3.675, which means that the robustness measure becomes better with a decrease of the value of λ. Note that this is an average trend based on the multivariate regression analysis, and the result may be a little different for different values of $version$ and φ_i.

4.3 The impact of other parameters on the reactive cost

In this part, we try to investigate the impact of the parameters related to the design of the problem set and the schedule disruptions on the reactive cost. The coefficient of the parameters may be a little different for different robustness measures, but the conclusion is almost identical. For the sake of brevity, here we only present the results of the best robustness measures, i.e., measure RM241 for environment E1 and measure RM224 for environment E2.

Table 8 Results of the multivariate regression of the parameters related to the design of the problem set and the schedule disruptions

Environment	R^2	Const	NC	RF	RS	α	$MTTR_k$	$MTTF_k$	σ
E1	0.367	331.680*	-8.118*	57.458*	-104.921*	-81.497*	51.194*	-5.525*	/
E2	0.086	204.264*	8.048*	-2.377*	-5.035*	-21.149*	/	/	53.916*

The results are shown in Table 8. It can be observed from the table that the resulting R^2-value is 0.367 under environment E1 and 0.086 under E2, which indicates that the parameters related to the design of the problem set and the schedule disruptions do not explain much of the variance of the reactive cost, especially under the environment of stochastic activity durations. Nevertheless, some interesting observations can be made.

We first discuss the impact of the parameters related to the design of the problem set, i.e., the network complexity NC, the resource factor RF, the resource strength RS and the due date factor α. The impact of parameter *NC* is small under the two uncertain environments and the impacts of the parameters *RF* and *RS* are much bigger for environment E1 than for E2. This is because the two parameters are related to resource availability, and thus they have a big impact on the schedule disruptions under the environment of stochastic resource availabilities. With an increase of *RF* and a decrease of *RS*, activities need more resource types to be executed and the resource constraints become stricter. At the same levels of $MTTR_k$ and $MTTF_k$, the resource availabilities are more likely to be insufficient and the reactive cost will increase. Therefore, the reactive cost will increase with an increase of *RF* and a decrease of *RS*. As for the due date factor α, the coefficients under the two environments are both smaller than zero, which implies that the reactive cost will decrease with an increase of the value of α. This is reasonable because a big value of α indicates a less strict project deadline constraint, which allows adding more time buffers to protect the starting times of activities. As a result, fewer efforts are needed in the reactive scheduling stage and the reactive cost will decrease.

With regard to the impact of the parameters related to schedule disruptions, two parameters, i.e., $MTTR_k$ and $MTTF_k$, are analyzed under environment E1, while only parameter σ is considered under environment E2. With a decrease of $MTTF_k$ and an increase of $MTTR_k$, resources break down more frequently and more time is needed for repairing them. Accordingly, resources become less available and the reactive cost increases. Similarly, with an increase of parameter σ, the duration variability is bigger, which causes that more efforts are needed to adjust the starting times of the activities and thus the reactive cost increases.

5. Discussion

In this section, we will discuss the consideration of the uncertain environment E3 in which both resource availability disruptions and activity duration disruptions occur. This type of environment is possible in practice and it is just the combination of the two uncertain environments considered in this paper. In the following, we will briefly explain how such a "combination" scenario can be modelled as well as its impact on the experimental design. The expected results of the computational experiment are also analyzed.

As concerns the modelling, the part of proactive scheduling remains the same, which means that the objective function is represented by formula (6) and the constraints are formulas (2), (3), (4) and (5) respectively. The general framework of surrogate robustness measures stays the same and there will be also 45 different calculations of the robustness measure. Similarly, the performance of the proposed surrogate robustness measures will be evaluated by the reactive cost, which is obtained through reactive simulations. Different from the reactive scheduling model for the environment E1, the precedence constraints will be now denoted by formula (15) instead of formula (10) to match the fact that stochastic activity durations are now additionally taken into account. In other words, in the reactive scheduling model for the environment E3, the objective function is formula (9) and the constraints are (15), (11), (12) and (13) respectively.

As concerns the computational experiment, the design of the tested instance sets and the proactive scheduling heuristic remains the same while the data generation and the reactive policy should be partly adjusted. With regard to the data generation, it might be now unsuitable to generate stochastic activity durations according to the way of Van de Vonder et al. (2008). If activity durations still follow right-skewed beta-distributions with parameters 2 and 5, many generated activity durations will be shorter than the expected values that are used in proactive scheduling. Through simulations using these generated data, the reactive costs are likely to be lower than those under the environment E1 with the same resource availability disruptions. This seems a little weird because compared with the environment E1, the reactive costs under the environment E3 are expected to increase when additionally taking stochastic activity durations into account. Under the environment E3, the way to generate stochastic activity durations and the development of the reactive scheduling policy rely on further research. For simulation, we can randomly generate 150 cases, for each problem, of the combination of stochastic resource availabilities and stochastic activity durations. Since the uncertainty becomes bigger when combining the two types of disruptions at the same time, to keep it realistic, we can only consider low variabilities of resource availabilities and activity durations.

Based on our analysis, the expected results under the environment E3 will be quite similar to those under the environment E1. This is due to the fact that resource availability disruptions are included in these two uncertain environments, against which time buffers have a limited protection. Note that it is just our theoretical analysis and a computational experiment is needed in future research to check whether it is correct.

6. Conclusion

In this paper, we propose a general framework of slack-based surrogate robustness measures with a focus on the slack utility function of each activity and the instability coefficient for weighting the utilities of different activities. In this framework, three parameters are introduced to distinguish different calculations of the robustness measure. A computational experiment based on reactive simulation is constructed where the performance of the surrogate measures is evaluated by the reactive cost and two types of uncertain environments, i.e. stochastic resource availabilities and stochastic activity durations, are taken into account. From the computational results, the following conclusions are drawn:

1) We analyze the impact of the three parameters as well as the two uncertain environments on the surrogate robustness measures and find the best measures for project managers to generate robust baseline schedules under different uncertain environments. The proposed measures are effective, the improvements of which are 2.67% and 13.79% respectively under the two uncertain environments compared with the benchmark robustness measures. The main reason for the small improvement under the environment of stochastic resource availabilities is that time buffers have a limited protection for activities against resource availability disruptions while resource buffers might be more beneficial to improve schedule robustness.
2) Under the environment of stochastic resource availabilities, it is necessary to have a decreasing marginal slack utility function and it is better for the marginal utility to decrease faster, i.e., with a smaller value of λ.
3) Under the environment of stochastic activity durations, we propose to calculate the instability coefficients in the second way, which is proven to be beneficial as more time buffers can be added in front of the dummy end activity to better protect its starting time. Furthermore, in calculating the instability coefficients of activities under this environment, it is better to take a small impact of a delay of one activity on its direct and indirect successors into account, i.e., with a small value of φ_i.
4) Strategies of adding time buffers are different under the two different uncertain environments. Under the environment of stochastic resource availabilities, it is not wise to add many time buffers for one specific activity. In other words, it is better to have a uniform distribution of time buffers throughout the schedule. However, the reverse is true under the environment of stochastic activity durations.

Two types of uncertain environments are taken into account in this paper while other types of disruptions can be considered in future research. Further research can be also devoted to proposing more effective surrogate robustness measures, such as taking the activity duration variance or the ratio of the slack of an activity to the corresponding activity duration into account. Besides, more effective algorithms, such as exact algorithms, can be developed to solve the proactive and reactive scheduling problem. Furthermore, for the environment of stochastic resource availabilities, resource buffers can be additionally considered to improve schedule robustness. It may be interesting to investigate the surrogate robustness measures with resource buffers and to explore the best way of combining time buffers and resource buffers.

References

Al-Fawzan, M.A., & Haouari, M. (2005). A bi-objective model for robust resource-constrained project scheduling. *International Journal of Production Economics*, *96*, 175–187.

Aytug, H., Lawley, M.A., Mckay, K., Mohan, S., & Uzsoy, R. (2005). Executing production schedules in the face of uncertainties: A review and some future directions. *European Journal of Operational Research*, *161*(1), 86-110.

Chakrabortty, R.K., Sarker, R.A., & Essam, D.L. (2016). Multi-mode resource constrained project scheduling under resource disruptions. *Computers & Chemical Engineering*, *88*, 13-29.

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

Cui, W., Lu, Z., Li, C., & Han, X. (2018). A proactive approach to solve integrated production scheduling and maintenance planning problem in flow shops. *Computers & Industrial Engineering*, *115*, 342-353.

Deblaere, F., Demeulemeester, E., Herroelen, W., & Van de Vonder, S. (2007). Robust resource allocation decisions in resource-constrained projects. *Decision Sciences*, *38*(1), 5-37.

Deblaere, F., Demeulemeester, E., & Herroelen, W. (2011). Reactive scheduling in the multi-mode RCPSP. *Computers & Operations Research*, *38*(1), 63-74.

Demeulemeester, E., & Herroelen,W. (2011). *Robust project scheduling*. Norwell: Now Publishers Inc.

Hazır, Ö., Haouari, M., & Erel, E. (2010). Robust scheduling and robustness measures for the discrete time/cost trade-off problem. *European Journal of Operational Research*, *207*(2), 633-643.

Hazır, Ö., Erel, E., & Günalay, Y. (2011). Robust optimization models for the discrete time/cost trade-off problem. *International Journal of Production Economics*, *130*(1), 87-95.

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

Herroelen, W., & Leus, R. (2004b). The construction of stable baseline schedules. *European Journal of Operational Research*, *156*, 550–565.

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

Hu, X., Demeulemeester, E., Cui, N., Wang J., & Tian, W. (2017). Improved critical chain buffer management framework considering resource costs and schedule stability. *Flexible Services and Manufacturing Journal*, *29*(2), 159-183.

Kobylański, P., & Kuchta, D. (2007). A note on the paper by M. A. Al-Fawzan and M. Haouari about a bi-objective problem for robust resource-constrained project scheduling. *International Journal of Production Economics*, *107*(2), 496-501.

Kolisch, R., & Sprecher, A. (1996). PSPLIB - A project scheduling problem library. *European Journal of Operational Research*, *96*(1), 205-216.

Lamas, P., & Demeulemeester, E. (2016). A purely proactive scheduling procedure for the resource-constrained project scheduling problem with stochastic activity durations. *Journal of Scheduling*, *19*(4), 409-428.

Lambrechts, O., Demeulemeester, E., & Herroelen, W. (2008a). Proactive and reactive strategies for resource-constrained project scheduling with uncertain resource availabilities. *Journal of Scheduling*, *11*(2), 121-136.

Lambrechts, O., Demeulemeester, E., & Herroelen, W. (2008b). A tabu search procedure for developing robust predictive project schedules. *International Journal of Production Economics*, *111*(2), 493-508.

Lambrechts, O., Demeulemeester, E., & Herroelen, W. (2011). Time slack-based techniques for robust project scheduling subject to resource uncertainty. *Annals of Operations Research*, *186*(1), 443-464.

Leus, R. (2003). *The generation of stable project plans*. PhD thesis, Department of applied economics, Katholieke Universiteit Leuven, Belgium.

Leus, R., & Herroelen, W. (2004). Stability and resource allocation in project planning. *IIE Transactions*, *36* (7), 1–16.

Liu, F., Wang, S., Hong, Y., & Yue, X. (2017). On the robust and stable flowshop scheduling under stochastic and dynamic disruptions. *IEEE Transactions on Engineering Management*, *64*(4), 539-553.

Ma, Z., He, Z., Wang, N., Yang, Z., & Demeulemeester, E. (2019). A genetic algorithm for the proactive resource-constrained project scheduling problem with activity splitting. *IEEE Transactions on Engineering Management,* to be appeared.

Sabuncuoglu, I., & Goren, S. (2009). Hedging production schedules against uncertainty in manufacturing environment with a review of robustness and stability research. *International Journal of Computer Integrated Manufacturing*, *22*(2), 138-157.

Schatteman, D., Herroelen, W., Van de Vonder, S., & Boone, A. (2008). Methodology for integrated risk management and proactive scheduling of construction projects. *Journal of Construction Engineering & Management*, *134*(11), 885-893.

Van de Vonder, S., Demeulemeester, E., Herroelen, W., & Leus, R. (2005). The use of buffers in project management: the trade-off between stability and makespan. *International Journal of Production Economics*, *97*, 227–240.

Van de Vonder, S., Demeulemeester, E., Herroelen, W., & Leus, R. (2006). The trade-off between stability and makespan in resource-constrained project scheduling. *International Journal of Production Research*, *44* (2), 215–236.

Van de Vonder, S., Ballestín, F., Demeulemeester, E., & Herroelen, W. (2007a). Heuristic procedures for reactive project scheduling. *Computers & Industrial Engineering*, *52*(1), 11-28.

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

Van de Vonder, S., Demeulemeester, E., & Herroelen, W. (2008). Proactive heuristic procedures for robust project scheduling: an experimental analysis. *European Journal of Operational Research*, *189*(3), 723-733.

Zhu, G., Bard, J.F., & Yu, G. (2005). Disruption management for resource-constrained project scheduling. *Journal of the Operational Research Society*, *56*(4), 365-381.