Open access • Journal Article • DOI:10.1007/S00291-018-0533-3

Robust resource-constrained max-NPV project scheduling with stochastic activity duration — Source link

Yangyang Liang, Nanfang Cui, Tian Wang, Erik Demeulemeester

Institutions: Hubei University, Huazhong University of Science and Technology, Zhongnan University of Economics and Law, Katholieke Universiteit Leuven

Published on: 01 Mar 2019 - OR Spectrum (Springer Berlin Heidelberg)

Topics: Robustness (computer science), Tabu search and Simulated annealing

Noname manuscript No.
(will be inserted by the editor)

Robust resource-constrained max-NPV project scheduling with stochastic activity duration

Yangyang Liang · Nanfang Cui ·
Tian Wang · Erik Demeulemeester

Received: date / Accepted: date

Abstract This study investigates the robust resource-constrained max-NPV project problem with stochastic activity duration. First, the project net present value (NPV) and the expected penalty cost (EPC) are proposed to measure quality robustness and solution robustness from the perspective of discounted cash flows, respectively. Then, a composite robust scheduling model is proposed in the presence of activity duration variability and a two-stage algorithm that integrates simulated annealing and tabu search is developed to deal with the problem. Finally, an extensive computational experiment demonstrates the superiority of the combination between quality robustness and solution robustness as well as the effectiveness of the proposed two-stage algorithm for generating project schedules compablack with three other algorithms, namely, simulated annealing, tabu search, and multi-start iterative improvement method. Computational results indicate that the proactive project schedules with composite robustness not only can effectively protect the payment plan from disruptions through allocating appropriate time buffers, but also can achieve a remarkable performance with respect to the project NPV.

Keywords Robust project scheduling · Net present value · Expected penalty cost · Quality robustness · Solution robustness

Yangyang Liang
Research Center of Hubei Logistics development, Hubei University of Economics, Wuhan, P.R. China
E-mail: yangliang0419@sina.com

Nanfang Cui
School of Management, Huazhong University of Science and Technology, Wuhan, P.R. China
E-mail: nfcui@hust.edu.cn

Tian Wang
School of Business Administration, Zhongnan University of Economics and Law, Wuhan, P.R. China
E-mail: wangtian3261@gmail.com

Erik Demeulemeester
Research Center for Operations Management, Department of Decision Sciences and Information Management, Faculty of Economics and Business, Katholieke Universiteit Leuven, Belgium
E-mail: Erik.demeulemeester@kuleuven.be

1 Introduction

The max-NPV problem introduced by Russell [31] is a new branch of the resource-constrained project scheduling problem (RCPSP), in which the project NPV is maximized by advancing activities with cash inflows as soon as possible whereas delaying activities with cash outflows as late as possible. Many research efforts on the max-NPV in the project scheduling have received much more attention in the recent years, compablack with those on the minimization of the project duration. Numberous models and algorithms for generating a workable baseline schedule under the objective of maximizing the project NPV have been advocated by various authors, such as Baroum and Patterson [3], Doersch and Patterson [10], Gu et al. [12], Hartmann and Briskorn [13], He et al. [14], Herroelen et al. [16], Leyman and Vanhoucke [26], Mika et al. [27], Neumann et al. [30], Tantisuvanichkul and Kidd [34], Waligóra [40], etc. However, the vast majority of these studies assume a static and deterministic environment with complete information, where activity durations and resource requirements are known in advance.

It is a well-known fact that in practice the projects are vulnerable to various types of disruptions, such as resource breakdowns, bad weather conditions, material supplies behind schedule, changes in delivery date, equipment failures as well as activities that have to be incorporated or abandoned [11, 39, 46]. As a result, one or more project activities may take more time than anticipated in the baseline schedule, and the realized project schedule cannot be executed exactly as planned. Moreover, these changes in the baseline schedule may exert great impacts on the expected project NPV (eNPV) [4, 44] especially for capital-intensive IT and construction projects, wherein large amounts of money are invested over long periods. Therefore, it is crucial and practical to generate a stable baseline schedule to ensure that the realized payment time of cash flows occurs as closely as possible to its original plan in a stochastic environment.

However, the study on the stochastic project scheduling under the objective of maximizing the eNPV is comparatively sparse. Buss and Rosenblatt [6] firstly address this problem and aim at maximizing the eNPV of the Markovian projects-PERT networks where activity durations are exponentially distributed. This method of the continuous-time Markov decision chain used by Buss and Rosenblatt is still the basis of some recent studies in this field [7, 33]. Wiesemann et al. [44] deal with activity durations and cash flows as discrete set of alternative scenarios with different occurrence probabilities and provide an optimal policy to obtain suitable results. Mohaghar et al. [29] examine project scheduling with the eNPV maximization through removing the possibility of the increasing in activity durations by the safe floats. To the best of our knowledge, the research on this problem is basically consideblack as a multi-stage decision process, which relies on the prior information about the distributions of activity duration. In addition, the major drawback of the previous studies is that no baseline schedules can be provided for the project decision makers.

Different from the study above, robust project scheduling, as a popular method of coping with uncertainties, involves the deployment of a stable proactive schedule to absorb disruptions as much as possible in the planning phase and of a reactive schedule to react to disruptions that cannot be absorbed by the proactive schedule during project execution [19, 25, 38]. The literature has distinguished between

two types of robustness measures: quality robustness and solution robustness. The former refers to the insensitivity of the objective value to disruptions [18]. The latter refers to the difference between the baseline schedule and the realized schedule [17].

Despite the popularity of robust project scheduling, the very few works take the project NPV into consideration, only with the project duration performance or the schedule stability itself. But these objectives may be unsuitable for capital-intensive projects, wherein financial aspects should be at the center of the decision maker's attention. Therefore, we investigate the robust resource-constrained max-NPV project scheduling problem with stochastic activity durations.

Our contribution is threefold. Firstly, we propose two indices for measuring quality robustness and solution robustness from the perspective of the project NPV. Secondly, we introduce time buffer allocation in the stochastic max-NPV problem and propose an EPC procedure to ensure that the payment plan of cash flows can be achieved as intended in the face of the activity duration variability. Thirdly, we set up a composite robust scheduling model and develop a two-stage algorithm that integrates simulated annealing and tabu search to deal with the problem with remarkable performance.

The rest of the paper is organized as follows. The next section is the problem statement. In Section 3, a time buffer allocation procedure is proposed and a simple project is used as an illustrative example. In addition, a composite robust scheduling model is constructed. In Section 4, a two-stage algorithm is developed to solve the model proposed above. In Section 5, an extensive computational experiment is performed. In the last section, the conclusions of the study are elaborated and a few future research directions are identified.

2 Problem statement

2.1 Notations and definitions

In this study, we assume that projects are given in an activity-on-node (AoN) representation. That is, a project is treated as a digraph $G = (N, A)$, where the set of nodes $N=\{0, 1...n+1\}$ represents the project activities and the set of arcs, $A \subseteq N \times N$, denotes the zero-lag finish-start precedence relations between activities. Nodes 0 and $n+1$ are the dummy activities that represent the start and completion of the project, respectively, which possess zero duration and zero resource usage. In this paper, we limit the resource category to renewable resources (e.g., machines, manpower or equipment) that are available on a period-by-period basis and for which a constant amount of the resource type k ($k \in K$) is R_k throughout project execution. r_{jk} is the resource requirement of activity j for resource type k.

It is known that the baseline schedule determines the start time of each activity, as well as sequences the activities that use the same resource unit(s) through certain resource-driven precedence relations. An elegant way to represent those resource-driven relations is a resource flow network, $G' = (N, A_R)$, with N the same set of nodes as in the original project network $G = (N, A)$ and A_R the set of resource flow arcs [1, 18]. A_R connects two nodes i and j if there exists a resource flow $f(i, j, k) > 0$ for any resource type k from activity i (when it finishes) to activity j (when it starts).

We further assume that our NPV problem is consideblack from the contractor's perspective. Negative cash flows or cash outflows include different types of costs related to resources, labors, and equipment, whereas positive cash flows or cash inflows denote payments to be made by the client to the contractor. Cash flows are associated with the execution of each activity, which is the sum of the contractor's costs and the client's payments for the activity. In this study, cash flows are discounted at the completion time of each activity with a discount rate α. In the following parts, the basic definitions and notations used throughout the paper are introduced to help understand the subsequent research.

$\mathbb{S}^{\mathbb{B}}$	baseline schedule
$\mathbb{S}^{R}$	realized schedule
s_j^B	planned start time of activity j in the baseline schedule
s_j^R	realized start time of activity j in the realized schedule
d_j^B	duration of activity j in the baseline schedule
d_j^R	duration of activity j in the realized schedule
P_j	set of all the direct and indirect pblackecessors of activity j in the arcs A
S_j	set of all the direct and indirect successors of activity j in the arcs A
cf_j	cash flows associated with activity j
cfw_j	cash flow weight of activity j
M_j^{NPV}	margin penalty cost of activity j
E_j^{NPV}	expected penalty cost of activity j
$L(i,j)$	longest path between activity i and activity j in the network $G \cup G'$
δ_{n+1}	project deadline
C	project completion time

2.2 Robustness types and measures

In this section, we introduce two indices for measuring quality robustness and solution robustness from the perspective of the project NPV.

2.2.1 Quality robustness

Quality robustness refers to the insensitivity of the baseline schedule to disruptions [37] and is measublack in terms of the objective function (e.g., project duration, project earliness and tardiness, and project costs). In our study, we adopt the objective function of the project NPV to evaluate quality robustness, which is calculated by the following formula.

$$\mathrm{NPV} = \sum_{j=1}^{n} cf_j e^{-\alpha\left(s_j^B + d_j^B\right)} \tag{1}$$

2.2.2 Solution robustness

Solution robustness or schedule stability refers to the difference between the baseline schedule $\mathbb{S}^B$ and the realized schedule $\mathbb{S}^R$ during project execution. The instability cost adopted by Leus and Herroelen [25] is normally regarded in literature as a common way of measuring solution robustness. This cost is calculated by the sum of expected weighted deviations in the start times in the realized schedule from those in the baseline schedule, i.e., $\Delta(\mathbb{S}^R, \mathbb{S}^B) = \sum_{j \in N} w_j E|s_j^R - s_j^B|$, where w_j denotes the activity disruption cost per unit time and E represents the expectation operator. However, the obtained values of the instability cost usually depend on the project execution simulation, which is a computationally demanding especially for large projects. In addition, this index contains no information of the project NPV [24]. Therefore, we propose the expected penalty cost of the project NPV index (E^{NPV}) to measure solution robustness.

E_j^{NPV} for activity j is defined as the expected penalty cost incurblack by each activity in delaying the payment time of cash flows by one-unit time during project execution from its planned payment time in the baseline schedule.

$$E_j^{\mathrm{NPV}} = M_j^{\mathrm{NPV}} \times P\left(s_j^R \geq s_j^B\right) \tag{2}$$

M_j^{NPV} denotes the margin penalty cost that is caused by starting the payment time of activity j one-unit time later than its planned payment time in the original baseline schedule. Given that the cash flows are discounted at the completion time of each activity, the M_j^{NPV} value can be calculated by the following formula:

$$M_j^{\mathrm{NPV}} = cfw_j \times \left(e^{-\alpha(s_j^B + d_j^B)} - e^{-\alpha(s_j^B + d_j^B + 1)}\right) \tag{3}$$

where the cash flow weight, cfw_j, of activity j is the sum of the cash flows of the activity itself and the cash flows of all its successors [3].

$P\left(s_j^R > s_j^B\right)$ denotes the probability that activity j cannot be started earlier than planned in the baseline schedule due to disruptions, i.e.,

$$P\left(s_j^R > s_j^B\right) = \sum_{(i,j) \in T(N, A \cup A_R)} P\left(d_i^R > s_j^B - s_i^B - L\left(i,j\right)\right) \tag{4}$$

Where $L(i,j)$ is sum of the durations of all activities on the longest path between activity i and activity j in the graph $T\left(N, A \cup A_R\right)$, which is calculated by the topological sorting method [22, 43]. The set of arcs A_R includes the additional resource arcs that connect two nodes if a resource flow exists between corresponding activities apart from the precedence relations in the project network. In our study, a feasible resource network is constructed by extending a parallel schedule generation scheme [1].

3 Rubust project schedules

3.1 Solution-robust project schedules

To protect the baseline schedule against unanticipated disruptions, researchers have advocated the use of time buffers in front of project activities so that the uncertainties during project execution can be compensated to a certain extent [9, 36]. Moreover, a number of time buffer allocation procedures have been developed for generating stable project baseline schedules [2, 25, 37]. Extensive simulation experiments in previous studies have revealed the effectiveness of time buffer allocation in providing a solution-robust project schedule [17, 38].

The activities with greater E_j^{NPV} have more negative effect on the project NPV compablack with others, and they need more time buffers to protect the propagation of delays in the payment plan throughout the project. Therefore, we develop an expected penalty cost (EPC) procedure for generating the solution-robust schedule. The mechanism of the EPC procedure is to iteratively create intermediate schedules by inserting a one-unit time buffer in front of the activity with the greatest E_j^{NPV} in the current intermediate schedule until allocating more time buffers no longer improves the schedule stability.

The computational steps of the EPC procedure are designed as follows:

Step 1: Generate an initial unbuffeblack baseline schedule with the minimized project duration by a branch and bound procedure [8].

Step 2: Calculate the E_j^{NPV} value for each activity in the current schedule following the Equation (2), and sort all the activities in descending order of the E_j^{NPV} (arbitrary tie-break).

Step 3: Allocate one-unit time buffer Δ in front of the currently selected activity. Then, increase the start times of the activity and its direct and indirect successors in the graph $G \cup G'$ by one-unit time, and update the current schedule.

Step 4: If the project completion time does not violate the project deadline ($C \leq \delta_{n+1}$), and the $\sum_{j \in N} E_j^{\mathrm{NPV}}$ results in a lower value, then the update schedule is feasible. Store this one as the input solution for the next iteration step, and then go to Step2; otherwise, remove one-unit time buffer in front of the activity and restore the schedule, and then go to Step5.

Step5: If there is no improvement found in the current schedule, take the next activity in the list, and then go to Step3.

Step 6: If $E_j^{\mathrm{NPV}} \leq 0$ and no feasible improvement is found, then the procedure terminates and a local optimum is obtained.

The EPC procedure allows the managers to mitigate the $\sum_{j \in N} E_j^{\mathrm{NPV}}$ in the planning phase by wisely allocating time buffers to improve the schedule stability. However, this procedure without quality robustness into consideration ignores the optimization of the project NPV.

3.2 Motivating example

In an effort to test the effectiveness of the EPC procedure, we select a simple project example, which consists of seven activities and only one type of resource with a constant availability of four units in

per-period. Figure 1 displays the project network of the example in the AoN representation. The number above each node represents the planned activity duration, while the number below each node denotes the per-period requirement of each activity for the single resource. Figure 2 depicts a feasible resource flow network that is expressed by the resource profile representation for the example project. The full arcs denote the direct precedence relationships in the original project network G, while the dashed arcs indicate additional precedence constraints imposed by the resource flows in the network G'. The cash flows (cf_j), cash flow weight (cfw_j), and margin penalty cost (M_j^{NPV}) of each activity j are displayed in Table 1 and the discounted rate α is set at 0.01 for this project.

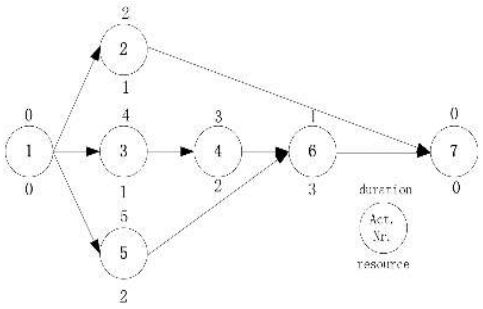

Fig. 1 A simple project network

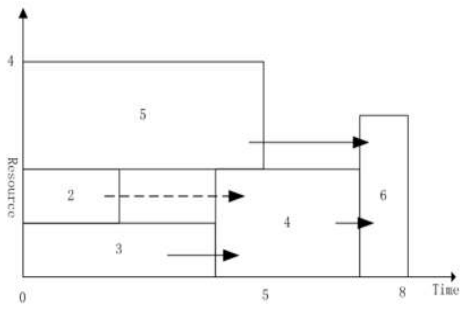

Fig. 2 A feasible resource flow network

Table 1 Margin penalty cost of each activity

activity	cf_j	cfw_j	M_j^{NPV}
2	12	12	0.119
3	-17	21	0.209
4	8	38	0.378
5	6	36	0.358
6	30	30	0.299

Table 2 Activity delay

activity	activity delay
2	1
3	1
4	0
5	3
6	0

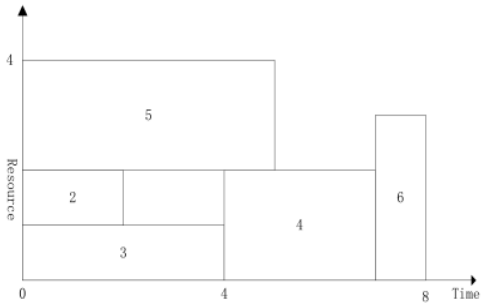

Fig. 3 Unbuffeblack baseline schedule $\mathbb{S}_1^B$

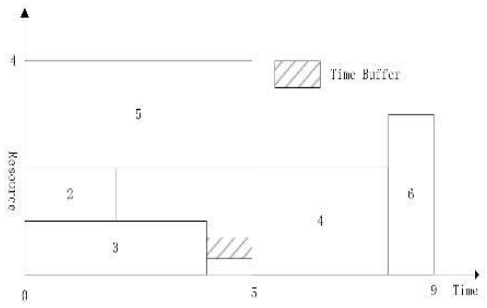

Fig. 4 Buffeblack baseline schedule $\mathbb{S}_2^B$

An unbuffeblack baseline schedule $\mathbb{S}_1^B$ is depicted in Figure 3, while a buffeblack baseline schedule $\mathbb{S}_2^B$ generated by the EPC procedure is described in Figure 4. The solution robustness values of $\mathbb{S}_1^B$ and $\mathbb{S}_2^B$ measublack by $\sum_{j \in N} E_j^{\text{NPV}}$ are 0.480 and 0.348, respectively. Therefore, the baseline schedule $\mathbb{S}_2^B$ with time buffers is more stable than the unbuffeblack schedule $\mathbb{S}_1^B$.

We then suppose that the execution of some activities do not accord with the expectations in the baseline schedule in a stochastic environment. Hence, the activities may be delayed due to disruptions, and the activity delay is displayed in Table 2.

$\mathbb{S}_1^B$ and $\mathbb{S}_2^B$ must react to the changes in activity durations and repair the schedule. The corresponding realized schedules $\mathbb{S}_1^R$ and $\mathbb{S}_2^R$ are shown in Figures 5 and 6, respectively. However, $\mathbb{S}_1^R$ and $\mathbb{S}_2^R$ exhibits the same completion time ($C = 9$) and the same project NPV (NPV = 35.816). After rescheduling, both activities 4 and 6 in $\mathbb{S}_1^R$ are affected by their pblackecessor delay (activity 3), whereas the start times of all activities in $\mathbb{S}_2^R$ remain the same as the original plan $\mathbb{S}_2^B$. In this sense, the buffeblack schedule $\mathbb{S}_2^B$ with the protection of time buffers can absorb the unexpected disruptions during project execution compablack with the unprotected schedule $\mathbb{S}_1^B$.

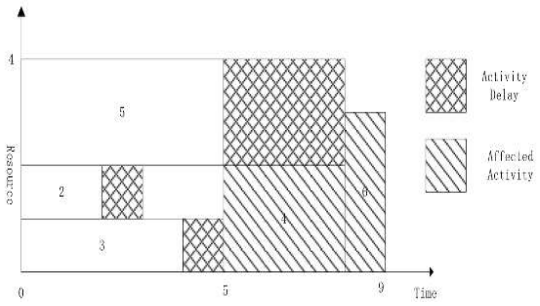

Fig. 5 Realized schedule $\mathbb{S}_1^R$

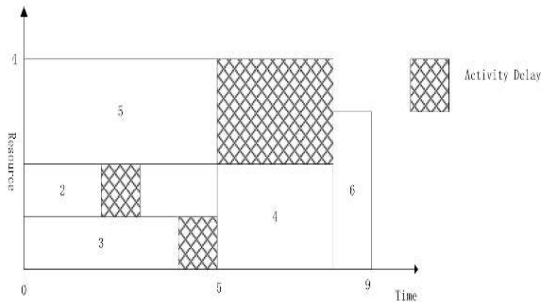

Fig. 6 Realized schedule $\mathbb{S}_2^R$

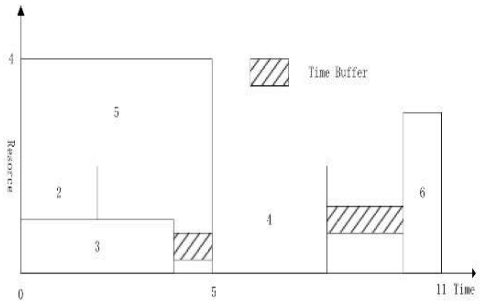

Fig. 7 Buffeblack baseline schedule $\mathbb{S}_3^B$

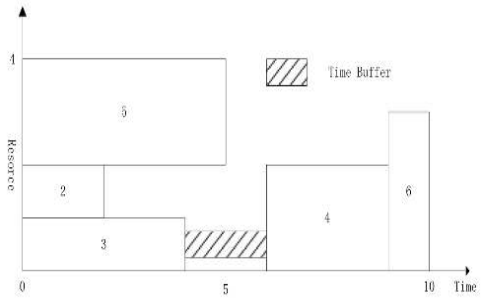

Fig. 8 Buffeblack baseline schedule $\mathbb{S}_4^B$

As described above, the EPC procedure can build a solution-robust schedule by allocating time buffers, which protects the payment plan from disruptions. However, the procedure only considers the schedule stability. In addition, different project deadlines in this procedure will directly affect the schedule stability and the project NPV for a given baseline schedule. For example, we set the deadline of the project in Figure 1 as 11 and 10, and the corresponding schedules $\mathbb{S}_3^B$ and $\mathbb{S}_4^B$ generated by the EPC procedure are shown in Figures 7 and 8, respectively. $\mathbb{S}_3^B$ shows an NPV of 35.396 and an $\sum_{j\in N} E_j^{\mathrm{NPV}}$ of 0.217, while those for the the NPV and the $\sum_{j\in N} E_j^{\mathrm{NPV}}$ are 35.741 and 0.293, respectively. In practice, managers not only expect a satisfactory NPV performance, but also pursue the schedule stability. Therefore, selecting which between $\mathbb{S}_3^B$ and $\mathbb{S}_4^B$ is a robust schedule is difficult. To obtain a stable baseline schedule that

considers solution robustness and quality robustness, a composite robustness model is constructed and presented in the following section.

3.3 Composite robustness model

In the previous section, the objective function of the project NPV is adopted to measure quality robustness. Meanwhile, the objective function of E^{NPV}, which is defined as the ability to cope with the deviations in the payment time of cash flows due to the increases in the durations of some activities, is proposed to evaluate solution robustness. In order to obtain a trade-off schedule between the two robustness measures, a composite robust scheduling model is constructed. The construction process of this model is displayed in Figure 9.

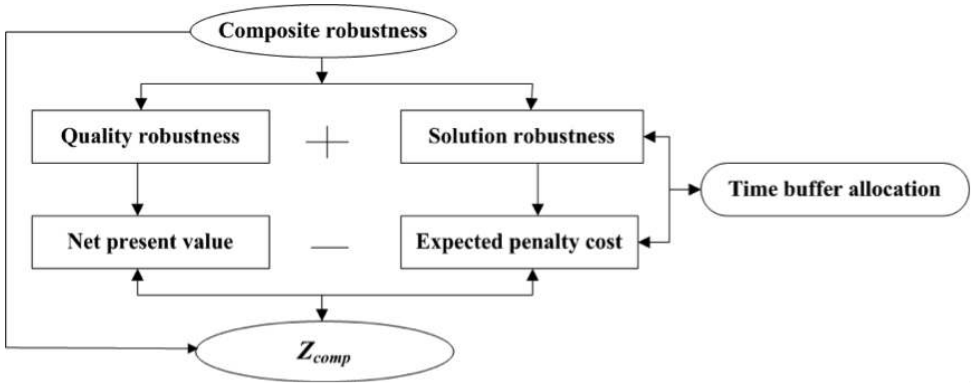

Fig. 9 The construction process of the composite robustness scheduling model

The objective function of the model is to maximize the composite robustness (Z_{comp}) of the schedules, which is defined as a simple combination of the two single objective functions described above, i.e.,

$$Max \;\; \mathrm{Z}_{comp} = \sum_{j=1}^{n} cf_j \times e^{-\alpha\left(s_j^B + d_j^B\right)} - \sum_{j=1}^{n} M_j^{\mathrm{NPV}} \times P\left(s_j^R > s_j^B\right) \tag{5}$$

Formally, the objective of our model is to determine the start time of each activity j $(j = 1, ..., n)$ in such a way that:

- The total resource requirements do not exceed the resource availability for each type of renewable resource in the per-period, $\sum_{j \in s(t)} r_{jk} \leq R_k$, where $s(t)$ is the set of activities in progress at time t.
- The baseline schedule $\mathbb{S}^B$ is generated by the proposed EPC procedure.
- The start time of each activity should satisfy the precedence constraints: $s_j^B = \Delta_j + \max_{i \in P_j}\left(s_i^B + d_i\right) \forall j \in N \backslash \{0, n+1\}$, where Δ_j is the time buffers inserted in front of activity j, and P_j is the set of pblackecessors of activity j in the digraph $G \cup G'$
- The project completion time should satisfy the negotiated deadline δ_{n+1}
- Quality robustness is measublack by $\sum cf_j \times e^{-\alpha\left(s_j^B + d_j^B\right)}$
- Solution robustness is measublack by $\sum M_j^{\mathrm{NPV}} \times P\left(s_j^R > s_j^B\right)$

The problem studied in this paper can be consideblack as an uncertain extension of the resource-constrained project scheduling problem with discounted cash flows (RCPSPDCF), which has been proven to be NP-hard by researchers [41, 45].

4 Heuristic algorithms

Tabu search (TS) and simulated annealing (SA) are popular metaheuristics and have been applied by many authors in solving the RCPSP [5, 15, 23, 28]. To avoid falling into a local optimum, SA offers the probability of accepting a non-improvement move while TS uses a mechanism for exploring wide regions in the search space. However, TS presents a stronger dependence on the initial solution than that of SA, i.e., an improved initial solution for TS can contribute to obtaining satisfactory solutions quickly [21, 42]. In this study, we develop a two-stage algorithm that integrates SA and TS to solve the proposed composite robust scheduling model.

In the first stage, we utilize SA to obtain the best-found solution. In the second stage, the best-found solution obtained in the first stage is consideblack as the initial solution of TS for subsequent exploration of solution space. This method is an attempt to improve the search efficiency of TS with an improved initial solution. In addition, we present three other standalone algorithms, namely, SA, TS, and multi-start iteration improvement (MSII, [32]) to provide comparable computational efforts for our two-stage algorithm, SA+TS.

4.1 Common elements

The common features of the above-mentioned algorithms are illustrated in the following sections.

4.1.1 Solution representation

A feasible solution is represented by two n-element lists below.

- **Activity position list, L_{posi}:** This list defines the order that activities are started. It is a precedence-feasible permutation of activities, in which each activity must be scheduled after all its pblackecessors and before all its successors so that no precedence constraints are violated. The subscript L_{posi} denotes the activity in the pth position of the list L.
- **Time buffer list, B_{posi}:** This list indicates the length of time buffers in front of each activity.

A combination of the activity position and time buffer lists, denoted by a pair of lists (L_{posi}, B_{posi}), can be decoded into a precedence and resource feasible schedule $\mathbb{S}^B = \{s_1^B, s_2^B ... s_n^B\}$ by exploiting an extended serial schedule generation scheme. The decoding procedure is described below, in which $P_{L_{posi}}$ represents the set of immediate pblackecessors of the activity in the pth position of the list L_{posi}.

Decoding procedure of the extended serial schedule generation scheme

---

$s_{L_1}^B = s_0^B = 0$

for $posi = 2$ to n do

$s^B_{L_{posi}} = max_{j \in P_{L_{posi}}}(s^B_j + d^B_j) + B_{L_{posi}}$

while $\exists k, t : \sum_{j \in s(t)} r_{jk} > R_k$ do

$s^B_{L_{posi}} = s^B_{L_{posi}} + 1$

$s^B_{L_{posi}} = s^B_{L_{posi}} + B_{L_{posi}}$

while $\exists k, t : \sum_{j \in s(t)} r_{jk} > R_k$ do

$s^B_{L_{posi}} = s^B_{L_{posi}} + 1$

$s^B_{n+1} = max(s^B_{n+1}, \delta_{n+1})$

4.1.2 Initial solution

The initial solution denoted by the pair of lists (L^{init}_{posi}, B^{init}_{posi}) is obtained by the proposed EPC procedure.

4.1.3 Objective function

For each neighbor solution, the composite objective function Z^{neig}_{comp} is calculated on the basis of the neighbor schedule, $\mathbb{S}^{neig} = \{s^{neig}_1, s^{neig}_2 ... s^{neig}_n\}$, which is characterized by the start time of each activity.

$$Z^{neig}_{comp} = Z^{neig}_{qual} - Z^{neig}_{stab} = \sum_{j=1}^{n} cf_j \times e^{-\alpha\left(s^{neig}_j + d^{neig}_j\right)} - \sum_{j=1}^{n} M^{\mathrm{NPV}}_j \times P\left(s^R_j > s^{neig}_j\right) \tag{6}$$

where Z^{neig}_{qual} and Z^{neig}_{stab} stand for quality robustness and solution robustness of the neighbor schedule $\mathbb{S}^{neig}$, respectively.

In the process of the two-stage algorithm, SA and TS use different mechanisms for generating neighborhoods. The detailed procedure of the two-stage algorithm will be introduced in the following sections.

4.2 First stage – simulated annealing

4.2.1 Mechanism of neighbor generation

The neighbor solutions of SA are generated through swapping the activity position, allocating the time buffer, and operating the combined move. Specifically, three strategies are used in the mechanism of neighbor generation.

- **Activity position swap (APS)** - a cyclical shift operates on the activity position list in the following way:

  Step1: Randomly choose one activity A on the pair of lists $L^{curr}_{posi}(SA)$;

  Step2: Find the latest pblackecessor B and the earliest successor C of activity A in the current solution, ($L^{curr}_{posi}(SA)$, $B^{curr}_{posi}(SA)$);

  Step3: Randomly choose a position D between activity B and activity C;

Step4: Move activity A to the position of activity D;

Step5: Obtain a new pair of lists ($L_{posi}^{neig}(SA)$, $B_{posi}^{neig}(SA)$) after a cyclical shift of all activities is applied between the old and the new positions.

This "activity position swap" process is illustrated in Figure 10, in which activity 3 is randomly chosen to swap the position.

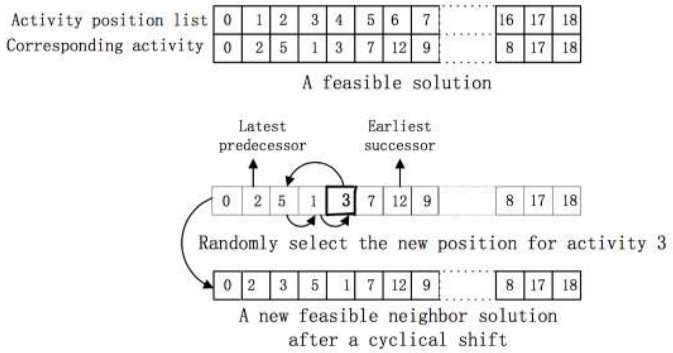

Fig. 10 An example of swapping the position of activity 3

- **Time buffer allocation (TBA)** - time buffers are allocated into a baseline schedule to offer protection against disruptions in the following way.

  Step1: Randomly choose one activity X on the $B_{posi}^{curr}(SA)$ list;

  Step2: Increase the time buffer length for activity X with a discrete value between $[-\Delta, +\Delta]$. First, the time buffer length Δ is set at 1 in our implementation. If no improvement is observed after five iterations, then we use a high Δ that is set at 3. Then, we change the time buffer length from $\Delta = 3$ to $\Delta = 5$ after ten iterations with no improved solution;

  Step3: Activity X itself and all its direct and transitive successors correspondingly move forwards or backwards, and the pair of lists ($L_{posi}^{neig}(SA)$, $B_{posi}^{neig}(SA)$)is updated;

  Step4: After adjustment, if the completion time of the project is beyond the deadline δ_{n+1}, then this neighbor is an infeasible solution; otherwise, it is regarded as a candidate.
- **Combined move (CM)** - both the activity position swap and the time buffer allocation are operated simultaneously. CM is performed in the following way:

  Step1: Randomly choose one activity Y on the $L_{posi}^{curr}(SA)$ list;

  Step2: Swap the position of activity Y following the APS strategy;

  Step3: Allocate the time buffer to activity Y following the TBA strategy;

  Step4: Update the pair of lists ($L_{posi}^{neig}(SA)$, $B_{posi}^{neig}(SA)$) and check whether the neighbor solution is feasible or not. If $C \leq \delta_{n+1}$, then the solution is feasible; otherwise, generate new ones.

APS, TBA , and CM are chosen randomly with a certain probability. For a given current solution, neighbor solutions represented by a pair of lists ($L_{posi}^{neig}(SA)$, $B_{posi}^{neig}(SA)$) can be generated following the mechanism proposed above.

4.2.2 Control parameters

- **Initial temperature:** The initial value of temperature T^{init} is calculated by the equation, $T^{init} = \Delta Z^{init}_{comp}(SA)/ln(\chi^{init})$, where $\Delta Z^{init}_{comp}(SA)$ is the range of change in the objective value after 50 random moves of the initial solution, and the initial acceptance ratio χ^{init} is defined as the assumed proportion between the accepted moves and all the moves generated for T^{init}.
- **Markov chain length:** The length of the Markov chain L determines the number of computations of the objective values at a certain temperature level, which is calculated as $L^{SA}_{SA+TS} = 5N$, where N is the number of activities in the project.
- **Cooling scheme:** To ensure the selectivity of the procedure, we gradually decrease the temperature throughout the process following the blackucing function: $T^{curr} := \mu T^{curr}$, where μ as the cooling rate is set at 0.9.
- **Stopping criterion:** The final temperature T^{stop} is set at 0.01 in this application and the search process terminates when the current temperature T^{curr} drops to the threshold, i.e., $T^{curr} \leq T^{stop}$.

4.2.3 Simulated annealing procedure

Step1: Set the pair of lists ($L^{init}_{posi}(SA)$, $B^{init}_{posi}(SA)$) as the initial solution of SA in the first stage, and calculate the corresponding objective value $Z^{init}_{comp}(SA)$; set $L^{curr}_{posi}(SA) := L^{init}_{posi}(SA)$, $L^{best}_{posi}(SA) := L^{init}_{posi}(SA)$, $B^{curr}_{posi}(SA) := B^{init}_{posi}(SA)$, $B^{best}_{posi}(SA) := B^{init}_{posi}(SA)$, $Z^{curr}_{comp}(SA) := Z^{init}_{comp}(SA)$, and $Z^{best}_{comp}(SA) := Z^{init}_{comp}(SA)$; set T^{init} as the initial temperature, $T^{curr} := T^{init} = \Delta Z^{init}_{comp}(SA)/ln(\chi^{init})$;

{The pair of lists ($L^{best}_{posi}(SA)$, $B^{best}_{posi}(SA)$) denotes the best-found solution currently found in SA. In addition, $Z^{neig}_{comp}(SA)$ and $Z^{best}_{comp}(SA)$ are the objective values that correspond to the neighbor solution and best known solution in SA, respectively.}

Step2: For a given value T^{curr}, randomly select a strategy to generate neighbor solutions denoted by the combination of $L^{neig}_{posi}(SA)$ and $B^{neig}_{posi}(SA)$. If the solution is feasible, then calculate the objective value $Z^{neig}_{comp}(SA)$. If the number of computations for the objective value $Z^{neig}_{comp}(SA)$ is more than $10N$, then go to Step5; otherwise, go to Step3.

Step3: If $\Delta Z^{curr}_{comp}(SA) = Z^{neig}_{comp}(SA) - Z^{curr}_{comp}(SA) > 0$, then store $L^{curr}_{posi}(SA) := L^{neig}_{posi}(SA)$, $B^{curr}_{posi}(SA) := B^{neig}_{posi}(SA)$, and $Z^{curr}_{comp}(SA) := Z^{neig}_{comp}(SA)$; then go to Step5; otherwise, go to Step 4.

Step4: Randomly select $R_{rand} \in (0, 1)$; If $R_{rand} < e^{-\Delta Z^{curr}_{comp}(SA)/T^{curr}}$, store $L^{curr}_{acti}(SA) := L^{neig}_{acti}(SA)$, $B^{curr}_{posi}(SA) := B^{neig}_{posi}(SA)$, and $Z^{curr}_{comp}(SA) := Z^{neig}_{comp}(SA)$, then go to Step5; otherwise, go to step 2;

Step5: Decrease the temperature following the blackuction function: $T^{curr} := \mu T^{curr}$;

Step6: Stopping criterion: If $T^{curr} \leq T^{stop}$, then the algorithm terminates, output the best solution $L^{best}_{posi}(SA)$, $B^{best}_{posi}(SA)$ and $Z^{best}_{comp}(SA)$; otherwise, go to Step2.

4.3 Second stage – tabu search

4.3.1 Initial solution

In the second stage, the initial solution of TS that is denoted by the pair of lists ($L_{posi}^{init}(TS)$, $B_{posi}^{init}(TS)$) is the best-found solution obtained by SA in the first stage. We set $L_{posi}^{init}(TS) = L_{posi}^{best}(SA)$, $B_{posi}^{init}(TS) = B_{posi}^{best}(SA)$, and $Z_{comp}^{init}(TS) = Z_{comp}^{best}(SA)$.

4.3.2 Mechanism of neighbor generation

In TS, the neighbor solutions are generated by changing the time buffer length without any confliction to the procedure relations and resource constraints [35]. At each iteration step, a maximum of $2(n-1)$ neighbor solutions exist. For each non-dummy activity in $B_{posi}^{curr}(TS)$, two possible neighbor solutions can be generated. The first one is obtained by increasing the time buffer length in front of the activity in the current schedule by one-unit time (*plus-move*). The other is obtained by decreasing the time buffer length of this activity by one time period (*minus-move*). This strategy is called the buffer length change (BLC), which operates in the following way.

Step1: For each non-dummy activity in $B_{posi}^{curr}(TS)$, increase or decrease the time buffer length of this activity by one-unit time at a time and keep the time buffers of all other activities unchanged; then update $L_{posi}^{neig}(TS)$ and $B_{posi}^{neig}(TS)$;

Step2: Correspondingly move the start times of the activity itself and all its direct and transitive successors forward or backwards by one-unit time; then update the neighbor schedule $\mathbb{S}^{neig}(TS)$;

Step3: After adjustment, if $C > \delta_{n+1}$, then this neighbor solution is infeasible. The others that satisfy the project deadline are regarded as the candidate set;

Step4: Select a feasible neighbor solution ($L_{posi}^{neig^*}(TS)$, $B_{posi}^{neig^*}(TS)$) such that $Z_{comp}^{neig^*}(TS)$ is the maximized one for all the neighbor objective values $Z_{comp}^{neig}(TS)$ in this iteration.

4.3.3 Tabu list

The tabu list that is used to record the steps of neighbor solutions is managed according to the tabu navigation method [28]. The length of tabu list is set at $\lceil\sqrt{n}\rceil$ in our implementation, where n is the number of project activities. Whenever a move is performed, its reverse is added to the tabu list in order to avoid returning to a solution that has been visited. The oldest existing move is removed from the front of the list following the FIFO (first-in-first-out) policy.

All moves in the this tabu list are forbidden. However, if a solution generated by an improvement move is better than the best solution that has been found so far, then cancel the forbidden status of the move.

4.3.4 Stop criterion

A reasonable rule must be set up to end the search process and thus ensure the computational capability of TS. In order to assure a comparable computation effort for each algorithm, the number of feasible

solutions visited by SA in the first stage, which is denoted as Num^{SA}_{SA+TS}, is recorded and taken as the stop criterion for TS in the second stage. In other words, if the total number of the solutions visited by TS reaches Num^{SA}_{SA+TS}, then the algorithm terminates and outputs the best-found solution, i.e., $Num^{TS}_{SA+TS} = Num^{SA}_{SA+TS}$.

4.3.5 Tabu search procedure

Step1: Set the best solution obtained in SA as the initial solution of TS, $L^{init}_{posi}(TS) = L^{best}_{posi}(SA)$, $B^{init}_{posi}(TS) = B^{best}_{posi}(SA)$, and $Z^{init}_{comp}(TS) = Z^{best}_{comp}(SA)$; then set $L^{curr}_{posi}(TS) := L^{init}_{posi}(TS)$, $L^{best}_{posi}(TS) := L^{init}_{posi}(TS)$, $B^{curr}_{posi}(TS) := B^{init}_{posi}(TS)$, $B^{best}_{posi}(TS) := B^{init}_{posi}(TS)$, $Z^{curr}_{comp}(TS) := Z^{init}_{comp}(TS)$, $Z^{best}_{comp}(TS) := Z^{init}_{comp}(TS)$, and $Num = 0$, $Num^{neig} = 0$;

Step2: Construct a set of neighbor solutions following the BLC strategy. If the solution is feasible, then $Num^{neig} = Num^{neig} + 1$;

Step3: Transform the pair of lists ($L^{neig}_{posi}(TS)$, $B^{neig}_{posi}(TS)$) to a feasible neighbor schedule $\mathbb{S}^{neig}(TS)$ and calculate the corresponding objective value $Z^{neig}_{comp}(TS)$;

Step4: Sort all the objective value $Z^{neig}_{comp}(TS)$ in descending order. In this list, the first solution with the maximum objective value is set as $Z^{neig*}_{comp}(TS)$. Check whether the move to $Z^{neig*}_{comp}(TS)$ is in the tabu list. If the answer is true,then go to Step5; otherwise, go to Step6.

Step5: If the move leading to the $Z^{neig*}_{comp}(TS)$ belongs to the tabu list, and $Z^{neig*}_{comp}(TS) > Z^{best}_{comp}(TS)$, then set $L^{curr}_{posi}(TS) := L^{neig*}_{posi}(TS)$, $B^{curr}_{posi}(TS) := B^{neig*}_{posi}(TS)$, and $Z^{curr}_{comp}(TS) := Z^{neig*}_{comp}(TS)$. Thereafter, cancel the forbidden attribute of the move and replace the oldest existing move in the tabu list. Meanwhile, set $L^{best}_{posi}(TS) := L^{curr}_{posi}(TS)$, $B^{best}_{posi}(TS) := B^{neig*}_{posi}(TS)$, $Z^{best}_{comp}(TS) := Z^{curr}_{comp}(TS)$, and $Num{=}Num{+}|Num^{neig}|$, and then go to Step7;

Step6: If the move corresponding to the solution Z^{neig*}_{comp} is non-tabu move, then set $L^{curr}_{posi}(TS) := L^{neig*}_{posi}(TS)$, $B^{curr}_{posi}(TS) := B^{neig*}_{posi}(TS)$, $Z^{curr}_{comp}(TS) := Z^{neig*}_{comp}(TS)$, and $Num{=}Num{+}|Num^{neig}|$, and update TL. If $Z^{neig*}_{comp}(TS) > Z^{best}_{comp}(TS)$, then set $L^{best}_{posi}(TS) := L^{neig*}_{posi}(TS)$, $B^{best}_{posi}(TS) := B^{neig*}_{posi}(TS)$, and $Z^{best}_{comp}(TS) := Z^{neig*}_{comp}(TS)$, and then go to Step7;

Step7: If $Num > Num^{TS}_{SA+TS}$, go to Step8, otherwise, go to Step2;

Step8: Output $L^{best}_{posi}(TS)$, $B^{best}_{posi}(TS)$, and $Z^{best}_{comp}(TS)$.

4.4 Comparable algorithms

Three other comparable algorithms, SA, TS, and MSII, start with the same initial solution and employ the same solution representation proposed above. In addition, MSII uses the same neighborhood mechanism as those applied in TS. To avoid the local optimum, MSII restarts with another random feasible solution when no improving moves exist.

To ensure a comparable computation effort for each algorithm, we set the same number of feasible solutions for SA, TS, SA+TS, and MSII in the computational experiment. The number of feasible solutions visited by SA (the single algorithm), which is denoted as Num_{SA}, is determined by two parameters, namely, Markov chain length and cooling rate. To maintain the feasible solutions of SA same with the

two-stage algorithm (SA+TS), the length of the Markov chain in SA is set to $L_{SA} = 10N$ compablack with $L^{SA}_{SA+TS} = 5N$ in SA+TS, i.e., $Num_{SA} = Num_{SA+TS} = Num^{SA}_{SA+TS} + Num^{TS}_{SA+TS}$. The feasible solutions of SA, Num_{SA}, is taken as the stop criterion for TS and MSII. In other words, TS and MSII terminate and output the best-found solutions as the desirable ones when the numbers of the solutions they have visited reach Num_{SA}, i.e., $Num_{SA} = Num^{stop}_{TS} = Num^{stop}_{MSII}$. On the basis of the above settings, the total number of feasible solutions visited by SA+TS is the same as those of SA, TS, and MSII, i.e., $Num_{SA+TS} = Num^{SA}_{SA+TS} + Num^{TS}_{SA+TS} = Num_{SA} = Num^{stop}_{TS} = Num^{stop}_{MSII}$.

5 Computational experiment

5.1 Experimental layout

To illustrate the capability of the proposed two-stage algorithm (SA+TS) to generate proactive project schedules with composite robustness, an extensive simulation study is performed to compare it with several other algorithms, namely, SA, TS, and MSII. A set of instances is randomly constructed by the project generator-ProGen to ensure that the comparison is carried out on a common basis. The parameter settings in our experiment are shown in Table 3. All the instances are put online for further verification.[1]

Table 3 The setting of control parameters for ProGen

Number of non-dummy activities	10, 30, 50
Number of instances generated for a given number of non-dummy activities	100
Number of start and end dummy activities	Randomly selected from 1, 2, and 3
Maximal number of pblackecessors or successors of each activity	4
Order strength	0.3, 0.5, 0.7
Activity durations (d^B_j)	Randomly selected from the interval $[1, 10]$
Discount rate (a)	0.01
Cash flows of activities (cf)	Randomly selected from the interval $[-100, 100]$
Types of renewable resource (K)	3
Number of renewable resource 1 (R_1)	10
Number of renewable resource 2 (R_2)	10
Number of renewable resource 3 (R_3)	10
Requirement of renewable resource 1 by activity j (r_{j1})	Randomly selected from the interval $[1, 10]$
Requirement of renewable resource 2 by activity j (r_{j2})	Randomly selected from the interval $[1, 10]$
Requirement of renewable resource 3 by activity j (r_{j3})	Randomly selected from the interval $[1, 10]$
Deadline of the project (δ_{n+1})	δ_{n+1}=1.3 × $S_{\min}$, where $S_{\min}$ is the minimum project duration

The realized activity durations are assumed to follow a right-skewed lognormal distribution in the simulation experiment, which is also used by Herroelen and Leus. [17], Tukel et al. [35] and Hu et al. [20]. More specifically, the d^R_j for activity j is randomly generated by the lognormal distribution function, allowing us to simulate the project execution with varying levels of uncertainty (represented by the standard deviation, σ) in the activity durations while keeping the mean durations unchanged. In our experiment, 1000 simulation replications for each project instance as well as 1000 realized activity durations for each activity are generated, and the average performances are calculated for various methods for testing. Three levels of σ (i.e., $\sigma \in \{0, 3, 0.6, 0.9\}$) are used to represent a project uncertainty that is Low (L), Medium (M) or High (H), respectively.

[1] https://ww2.mathworks.cn/matlabcentral/fileexchange/67332-900-project-instances-are-randomly-constructed-by-the-project-generator-progen

The expected penalty cost is developed to measure solution robustness in Section 2.2.2, however, the probability $P(s_j^R \geq s_j^B)$ becomes a deterministic result for a given d_j^R in the simulation experiment. Therefore, we propose a surrogate measurement that is defined as the total expected penalty cost of cash flows resulting from deviations between the realized start time s_j^R and the planned start time s_j^B to estimate solution robustness, i.e.,

$$Z_{stab} = \sum_{j=1}^{n} M_j^{\mathrm{NPV}} \times E\left|s_j^R - s_j^B\right| \tag{7}$$

The simulated execution of a baseline schedule uses the parallel scheduling generation scheme to determine the realized start time s_j^R of each activity. The literature has distinguished two types of scheduling policies in the project simulations: roadrunner scheduling policy and railway scheduling policy [36]. In the former, the activities should start as soon as possible when all their pblackecessors have finished and when enough resource units are available. The latter one requires that the realized start time s_j^R should not start earlier than its planned start time s_j^B to increase the schedule stability [38]. Therefore, we apply the railway scheduling policy to maintain the stability of the payment plan of activities during project execution, i.e.,

$$s_j^R = \max\left(s_j^B, \quad \Delta_j + \max_{i \in P_j}\left(s_i^R + d_i^R\right)\right), \forall j \in N \tag{8}$$

with the time buffer length Δ_j inserted in front of activity j.

5.2 Performance of algorithms

In this section, we conduct a set of experiments regarding the efficiency of the proposed two-stage algorithm (SA+TS) as opposed to three other algorithms, i.e. SA, TS, and MSII. Six indicators are defined to evaluate the performance of these proposed four algorithms.

- $Best(\%)$: Percentage of the instances for which a certain algorithm yields the maximum value of the objective function Z_{comp} among SA, TS, MSII, and SA+TS.
- Z_{comp}^{best}: Best-found solutions of the objective function.
- $A\text{-}GAP(\%)$: Average relative deviations from the best-found solutions Z_{comp}^{best} after 1000 evaluations.
- $M\text{-}GAP(\%)$: Maximal relative deviations from the best-found solutions Z_{comp}^{best} after 1000 evaluations.
- $A\text{-}CPU(s)$: Average computational time.
- $M\text{-}CPU(s)$: Maximal computational time.

The comparison results of the experiment are presented in Table 4. When computational times are not taken into consideration, our two-stage algorithm (SA+TS) outperform SA, TS, and MSII in terms of $Best$, Z_{comp}^{best}, $A\text{-}GAP$, and $M\text{-}GAP$, and these superiorities become more remarkable especially when the project instances become larger. In searching for the best-found solutions, we find that 520 instances out of 900 (57.78%) are solved effectively by SA+TS compablack with SA (23.33%), TS (12.78%), and MSII (6.11%). These results are not surprising because the search process of our two-stage algorithm integrates SA and TS, thereby possessing higher search efficiency and exploring more solution space than do simple

search procedures, such as SA, TS, and MSII. Table 4 also shows that the performance of SA is better than that of TS because SA uses a more complex mechanism to generate neighbor solutions than that of TS. TS and MSII employ the same neighbor generation mechanism and termination criterion. However, TS can obtain better results than those of MSII because the former can avoid the feasible solutions to be visited repeatedly.

With regard to computational time on the basis of the same number of feasible solutions visited by each algorithm, which is reflected by the *A-CPU* and *M-CPU* indices, MSII is the fastest, followed by TS, SA+TS, and SA as the slowest. This result is due to the fact that SA exhibits the most complex search process among the studied algorithms and thus requires much computational effort within the acceptance scope to obtain a satisfactory solution. Concerning TA and MSII, since MSII does not need to spend additional time to manage the tabu list and the cyclical shift compablack with TS such that the times obtained for MSII are significantly shorter than those for others. The performance of the *A-CPU* and *M-CPU* indicators for SA+TS is between SA and TS, it is quite understand that half of feasible solutions of SA+TS are generated by the mechanism of SA; the other half ones are constructed by TS.

Table 4 Computational resluts of the algorithm performance

N-2	OS	Algorithm	Best(%)	Z_{comp}^{best}	A-GAP(%)	M-GAP(%)	A-CPU(s)	M-CPU(s)
10	0.7	SA	34	60.56	0.39	3.13	5.34	5.97
		TS	13	59.75	0.53	4.71	4.03	4.64
		MSII	8	55.61	0.95	5.38	2.63	3.12
		SA+TS	45	61.83	0.26	1.75	4.88	5.23
	0.5	SA	30	58.59	0.42	3.91	4.56	5.21
		TS	19	57.42	0.56	5.18	3.77	4.26
		MSII	8	53.35	1.12	6.23	2.01	2.68
		SA+TS	43	59.44	0.31	1.79	4.05	4.83
	0.3	SA	32	58.26	0.35	3.48	4.21	4.56
		TS	17	57.31	0.47	4.49	3.23	3.67
		MSII	11	54.89	0.79	5.93	1.86	2.34
		SA+TS	40	59.17	0.25	1.57	3.78	4.13
30	0.7	SA	23	154.65	1.72	4.34	16.73	17.52
		TS	12	148.42	2.47	6.03	12.26	13.49
		MSII	6	141.65	3.83	8.59	7.55	8.08
		SA+TS	59	161.03	1.36	3.68	14.43	15.64
	0.5	SA	24	165.85	1.47	5.21	17.32	17.95
		TS	10	161.56	2.63	6.54	13.28	14.08
		MSII	5	157.93	3.27	8.02	7.64	8.27
		SA+TS	61	170.45	1.29	3.46	15.33	16.98
	0.3	SA	22	158.52	1.16	4.89	16.34	17.04
		TS	14	152.34	2.26	6.79	12.66	13.05
		MSII	6	145.03	3.28	7.88	8.03	8.84
		SA+TS	58	164.26	0.92	2.81	14.37	15.56
50	0.7	SA	15	398.03	3.93	7.53	59.84	60.96
		TS	10	380.25	5.02	8.87	45.03	46.21
		MSII	4	363.55	5.98	10.73	31.16	32.57
		SA+TS	71	412.53	2.83	4.72	53.42	54.81
	0.5	SA	16	387.12	3.26	6.65	58.15	59.26
		TS	12	376.43	4.27	8.03	43.18	44.05
		MSII	2	358.65	5.63	9.87	30.41	31.54
		SA+TS	70	398.33	2.05	3.98	52.28	53.95
	0.3	SA	14	418.26	3.56	6.62	58.18	58.54
		TS	8	405.25	4.84	8.42	45.68	46.47
		MSII	5	389.38	6.02	10.73	28.51	30.36
		SA+TS	73	431.73	2.37	4.58	50.19	51.25

Figure 11 shows how the best-found solution of each algorithm (SA+TS, SA, TS, and MSII) improves as the simulation replications increase for the case with order strength 0.3, 10 activities. As denoted in Figure 11, SA+TS has the fastest convergence speed and best quality, and then SA, TS, and MSII works

worsts. This follows from the fact that SA+TS both has an improved initial solution and several strategies in the mechanism of neighbor generation, which ensures higher search efficiency and more solution space than SA, TS, and MSII.

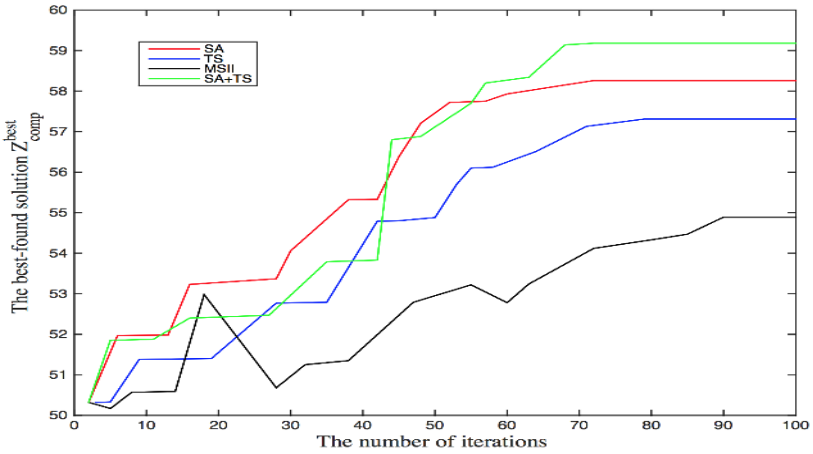

Fig. 11 Best-found solutions Z_{comp}^{best} as a function of replications

5.3 Robustness analysis

To verify whether the buffeblack schedules with composite robustness outperform the schedules with the single robustness or the schedules without time buffers, five types of schedules (F1, F2, F3, F4, and F5) are constructed.

- F1: buffeblack schedules with composite robustness maximization.
- F2: unbuffeblack schedules with composite robustness maximization.
- F3: buffeblack schedules with the EPC minimization.
- F4: unbuffeblack schedules with the project NPV maximization.
- F5: buffeblack schedules with the project NPV maximization.

For each instance generated by the ProGen, we adopt four algorithms (i.e., SA, TS, MSII, and SA+TS) to generate the corresponding baseline schedules F1, F2, F3, F4, and F5 according to their different objective functions. Tables 5-7 provide the statistical results of $\overline{Z}_{comp}$, $\overline{Z}_{qual}$, and $\overline{Z}_{stab}$ that represent the average objective values of composite robustness, quality robustness, and solution robustness, respectively, after 1000 simulation runs over all instances under three levels of duration variability. First of all, as uncertainty (σ) in the project environment increases, $\overline{Z}_{qual}$ decreases, whereas $\overline{Z}_{stab}$ increases as pblackicated in all cases. It is obvious that the higher the duration variability, the more risks the project will be faced with, inevitably causing an adverse effect on the schedule robustness. Secondly, as the order strength decreases, meaning that the project networks are more complicated, $\overline{Z}_{stab}$ in general increased whereas $\overline{Z}_{qual}$ seems to be dependent of the order strength. Thirdly, it is important to note that the proposed two-stage algorithm, i.e. SA+TS, always achieves better valuse of $\overline{Z}_{comp}$, $\overline{Z}_{qual}$, and $\overline{Z}_{stab}$

than those of three other algorithms (SA, TS, and MSII), which further demonstrates the superiority of SA+TS for generating composite robust schedules.

For different types of schedules, it can be seen obviously that $\overline{Z}_{comp}$ obtained by Schedule F1 is the best among those obtained by Schedules F2, F3, F4, and F5. This result may come from the fact that if the decision makers consider to generate a buffeblack baseline schedule with composite robustness (Schedule F1), which not only can protect the payment plan from disruptions through allocating appropriate time buffers, but also can obtain a remarkable performance with respect to the project NPV. Tables 5-7 also indicates that Schedules F1, F3, and F5 can improve solution robustness by allocating time buffers against disruptions unlike the execution of a nominal schedule without time buffers (i.e., Schedules F2 and F3); e.g., $\overline{Z}_{stab}(F1) < \overline{Z}_{stab}(F2)$, $\overline{Z}_{stab}(F3) < \overline{Z}_{stab}(F4)$, and $\overline{Z}_{stab}(F5) < \overline{Z}_{stab}(F4)$.

What's more, the single objective value of $\overline{Z}_{qual}$ in Schedules F1 and F3 outperform those of Schedules F2 and F4, i.e., $\overline{Z}_{qual}(F1) > \overline{Z}_{qual}(F2)$ and $\overline{Z}_{qual}(F3) > \overline{Z}_{qual}(F4)$. The reason lies in the following facts that there is a complementary relationship exists between quality robustness and solution robustness in our model. The baseline schedules F2 and F4 are generated only by changing the position of activities without any time buffer protection. As a result, the realized project schedules of F2 and F4 would suffer greater losses in $\overline{Z}_{qual}$ than those of F1 and F3 compablack with the planned values during project execution. On the other hand, the realized schedules of F1 and F3 with time buffers can implement better according to the original plan and can blackuce the loss of $\overline{Z}_{qual}$ to a certain extent compablack with unbuffeblack ones, F2 and F4.

As denoted in Tables 5-7, $\overline{Z}_{qual}$ and $\overline{Z}_{stab}$ in Schedules F1 are better than those of F5. This follows the fact that the project NPV in F5 is maximized by allocating time buffers to the activities with cash outflows as much as possible whereas allocating time buffers to the activities with cash inflows as few as possible on the basis of satisfying the project deadline, which is not compatible with the motivation of time buffer allocation. Consequently, it is impossible to ensure that all time buffers in F5 can play their proper role in protecting the payment plan against disruptions. This finding means that F5 may suffer more losses both in $\overline{Z}_{qual}$ and $\overline{Z}_{stab}$ compablack with F1 during project execution.

Solution robustness in F3 is better than that of F5, whereas quality robustness in F3 performs worse than that of F5. This finding can be explained by two primary factors. First, the objective function of F3 is to minimize the expected penalty cost of the project NPV, but F5 pursues the maximization of the project NPV. Thus, time buffers allocated in F3 can more effectively protect the payment plan against disruptions during project execution compablack with F5, i.e., $\overline{Z}_{stab}(F3) < \overline{Z}_{stab}(F5)$. Second, F3 only considers solution robustness without optimizing the project NPV in the planning phase. As a result, time buffers in F5 can prevent disruptions in the payment plan to a certain extent during project execution compablack with F3. In other words, F5 can obtain a more satisfactory result of the expected project NPV compablack with F3, i.e., $\overline{Z}_{qual}(F3) < \overline{Z}_{qual}(F5)$.

Table 5 Average results of different algorithms with low duration variability

N-2	OS	Algorithm	Composite robustness						Single robustness								
			Buffeblack (F1)			Unbuffeblack (F2)			Buffeblack (F3)			Unbuffeblack (F4)			Buffeblack (F5)		
			$\overline{z}_{comp}$	$\overline{z}_{qual}$	$\overline{z}_{stab}$	$\overline{z}_{comp}$	$\overline{z}_{qual}$	$\overline{z}_{stab}$	$\overline{z}_{comp}$	$\overline{z}_{qual}$	$\overline{z}_{stab}$	$\overline{z}_{comp}$	$\overline{z}_{qual}$	$\overline{z}_{stab}$	$\overline{z}_{comp}$	$\overline{z}_{qual}$	$\overline{z}_{stab}$
10	0.7	SA	55.45	60.92	5.47	45.59	55.13	9.54	49.35	53.57	4.22	31.83	44.78	12.95	47.71	54.69	6.98
		TS	52.82	60.14	7.32	41.26	53.62	12.36	47.31	52.48	5.17	26.25	42.02	15.77	44.72	53.01	8.29
		MSII	50.45	59.21	8.76	35.35	49.54	14.19	41.03	48.85	7.82	19.51	38.39	18.88	39.03	49.18	10.15
		SA+TS	57.18	61.84	4.66	48.34	57.22	8.88	51.98	54.65	2.67	36.19	46.95	10.76	51.61	56.84	5.23
	0.5	SA	53.44	59.59	6.15	44.72	55.27	10.55	47.24	52.72	5.48	28.72	42.79	14.07	46.07	53.56	7.49
		TS	50.81	58.54	7.73	40.54	53.29	12.75	44.59	50.84	6.25	22.96	40.14	17.18	43.31	52.23	8.92
		MSII	48.85	57.98	9.13	33.82	48.74	14.92	37.92	46.25	8.33	13.91	33.23	19.32	36.78	47.16	10.38
		SA+TS	54.77	60.52	5.75	48.16	57.35	9.19	50.17	53.95	3.78	33.73	45.48	11.75	49.02	55.88	6.86
	0.3	SA	56.54	63.72	7.18	45.88	58.12	12.24	48.68	54.93	6.25	31.81	46.76	14.95	48.13	56.85	8.72
		TS	54.69	62.95	8.26	42.09	56.84	14.75	44.71	51.84	7.13	27.22	45.25	18.03	43.99	53.84	9.85
		MSII	50.41	60.66	10.25	34.34	50.68	16.34	38.12	47.96	9.84	16.85	37.82	20.97	37.11	48.57	11.46
		SA+TS	58.82	65.11	6.29	49.21	59.17	9.96	51.25	55.68	4.43	36.53	49.34	12.81	48.53	56.15	7.62
30	0.7	SA	150.66	175.82	25.16	116.01	157.36	41.35	134.38	146.85	12.47	80.21	130.16	49.95	119.67	152.31	32.64
		TS	139.93	168.45	28.52	102.19	147.78	45.59	121.46	137.25	15.79	65.18	119.62	54.44	105.15	141.33	36.18
		MSII	120.91	155.76	34.85	76.63	126.26	49.63	91.35	115.47	24.12	38.73	107.84	69.11	74.34	118.09	43.75
		SA+TS	159.09	182.57	23.48	129.62	166.81	37.19	142.49	153.44	10.95	99.42	139.51	40.09	130.14	160.19	30.05
	0.5	SA	150.21	180.33	30.12	111.79	159.08	47.29	129.17	147.82	18.65	82.42	134.94	52.52	113.08	153.06	39.98
		TS	133.64	169.62	35.98	100.26	152.97	52.71	118.13	139.96	21.83	56.38	118.66	62.28	97.54	143.16	45.62
		MSII	118.99	158.95	39.96	67.44	125.56	58.12	85.74	115.22	29.48	31.98	106.47	74.49	69.75	120.21	50.46
		SA+TS	163.42	191.16	27.74	125.83	168.45	42.62	137.82	154.08	16.26	99.97	145.95	45.98	121.92	159.78	37.86
	0.3	SA	135.77	170.93	35.16	91.64	140.83	49.19	106.97	132.84	25.87	61.45	121.82	60.37	92.93	135.66	42.73
		TS	112.65	151.63	38.98	79.48	137.25	57.77	91.37	121.12	29.75	43.23	108.68	65.45	75.94	125.89	49.95
		MSII	92.66	140.41	47.75	58.31	122.37	64.06	79.06	111.99	32.93	17.75	98.44	80.69	56.94	115.75	58.81
		SA+TS	146.01	179.64	33.63	114.39	160.74	46.35	127.12	147.28	20.16	83.79	135.35	51.56	112.83	154.39	41.56
50	0.7	SA	271.65	332.54	60.89	206.23	297.98	91.75	235.34	276.17	40.83	142.85	242.04	99.19	215.47	290.71	75.24
		TS	249.48	316.82	67.34	192.34	289.29	96.95	218.68	265.44	46.76	116.85	232.67	115.82	195.18	278.05	82.87
		MSII	210.71	287.57	76.86	170.52	270.05	99.53	188.71	251.66	62.95	81.56	212.83	131.27	172.68	262.31	89.63
		SA+TS	294.42	348.04	53.62	239.27	315.63	76.36	259.38	289.26	29.88	176.53	263.36	86.83	241.87	306.28	64.41
	0.5	SA	282.15	347.83	65.68	209.25	311.98	102.73	238.49	286.45	47.96	142.78	256.84	114.06	223.71	305.95	82.24
		TS	250.33	322.55	72.22	192.58	302.07	109.49	220.93	277.64	56.71	110.81	237.25	126.44	194.86	284.81	89.95
		MSII	226.37	306.72	80.35	157.68	283.62	125.94	185.94	259.76	73.82	81.15	221.44	140.29	169.15	272.83	103.68
		SA+TS	304.19	359.51	55.32	230.69	323.81	93.12	263.11	299.39	36.28	177.24	276.86	99.62	242.14	312.41	70.27
	0.3	SA	291.03	361.17	70.14	215.44	323.72	108.28	246.09	306.46	60.37	134.28	260.04	125.76	226.54	314.36	87.82
		TS	267.96	344.44	76.48	197.69	314.95	117.26	224.49	293.25	68.76	102.07	251.92	149.85	207.97	302.53	94.56
		MSII	235.05	321.92	86.87	150.61	289.49	138.88	187.33	269.81	82.48	59.39	233.76	174.37	168.02	278.05	110.03
		SA+TS	305.71	364.26	58.55	238.12	335.57	97.45	264.24	313.52	49.28	172.36	284.52	112.16	247.43	326.14	78.71

Table 6 Average results of different algorithms with medium duration variability

N-2	OS	Algorithm	Composite robustness						Single robustness								
			Buffeblack (F1)			Unbuffeblack (F2)			Buffeblack (F3)			Unbuffeblack (F4)			Buffeblack (F5)		
			$\overline{z}_{comp}$	$\overline{z}_{qual}$	$\overline{z}_{stab}$	$\overline{z}_{comp}$	$\overline{z}_{qual}$	$\overline{z}_{stab}$	$\overline{z}_{comp}$	$\overline{z}_{qual}$	$\overline{z}_{stab}$	$\overline{z}_{comp}$	$\overline{z}_{qual}$	$\overline{z}_{stab}$	$\overline{z}_{comp}$	$\overline{z}_{qual}$	$\overline{z}_{stab}$
10	0.7	SA	48.01	55.89	7.88	33.65	48.97	15.32	41.36	47.41	6.05	23.49	41.64	18.15	37.28	48.31	11.03
		TS	43.86	53.25	9.39	28.65	45.83	17.18	37.26	44.52	7.26	15.35	36.76	21.41	32.90	46.36	13.46
		MSII	31.95	44.99	13.04	20.52	42.36	21.84	29.38	40.25	10.87	6.21	30.15	23.94	23.23	41.48	18.25
		SA+TS	53.34	60.26	6.92	40.04	52.75	12.71	46.68	51.47	4.79	30.65	44.47	13.82	43.75	53.63	9.88
	0.5	SA	44.61	53.77	9.16	31.11	49.32	18.21	39.37	46.68	7.31	18.31	40.18	21.87	33.41	47.65	14.24
		TS	40.87	51.32	10.45	23.92	44.87	20.95	33.11	41.76	8.65	11.07	34.82	23.75	26.61	43.26	16.65
		MSII	24.51	40.84	16.33	17.65	40.74	23.09	20.87	34.39	13.52	1.36	29.39	28.03	15.39	36.25	20.86
		SA+TS	48.57	57.41	8.84	37.01	51.43	14.42	42.78	48.66	5.88	25.29	41.03	15.74	38.95	50.32	11.37
	0.3	SA	48.51	58.68	10.17	29.27	50.22	20.95	40.65	49.34	8.69	20.50	45.26	24.76	34.61	50.86	16.25
		TS	41.33	53.25	11.92	24.92	48.61	23.69	37.35	46.59	9.24	14.92	42.73	27.81	27.58	47.27	19.69
		MSII	29.39	46.92	17.53	16.22	42.94	26.72	26.36	41.71	15.35	6.35	40.14	33.79	18.94	42.48	23.54
		SA+TS	49.79	59.16	9.37	38.79	54.12	15.33	45.15	52.19	7.04	26.24	46.29	20.05	41.22	54.34	13.12
30	0.7	SA	116.33	157.11	40.78	80.83	135.58	54.75	97.83	121.74	23.91	49.88	114.86	64.98	86.35	133.44	47.09
		TS	95.26	138.82	43.56	63.35	126.76	63.41	89.23	117.32	28.09	36.54	109.27	72.73	74.87	125.85	50.98
		MSII	70.51	120.75	50.24	45.19	115.51	70.32	64.89	102.21	37.32	15.62	94.27	78.65	48.85	111.19	62.34
		SA+TS	139.61	173.43	33.82	96.57	145.05	48.48	117.28	135.37	18.09	73.92	129.44	55.52	101.69	142.17	40.48
	0.5	SA	120.09	163.18	43.09	81.72	140.33	58.61	95.99	126.72	30.73	46.84	117.39	70.55	87.53	138.86	51.33
		TS	98.97	147.43	48.46	65.49	132.24	66.75	93.69	129.24	35.55	31.34	110.61	79.27	73.18	131.02	57.84
		MSII	64.76	123.74	58.98	46.43	121.75	75.32	59.44	104.15	44.71	10.57	95.72	85.15	53.14	117.71	64.57
		SA+TS	142.05	178.76	36.71	101.92	153.26	51.34	118.28	140.54	22.26	73.96	134.78	60.82	106.73	149.12	42.39
	0.3	SA	102.15	150.33	48.18	64.64	129.51	64.87	79.08	114.65	35.57	31.09	109.32	78.23	69.21	125.66	56.45
		TS	80.36	133.98	53.62	48.53	119.94	71.41	65.56	104.42	38.86	11.84	101.55	89.71	49.66	114.73	65.07
		MSII	54.67	119.43	64.76	28.42	106.25	77.83	39.83	89.64	49.81	1.17	93.89	92.72	33.79	103.45	69.66
		SA+TS	125.03	165.67	40.64	81.79	138.88	57.09	94.62	121.27	26.65	46.52	115.71	69.19	83.99	133.89	49.90
50	0.7	SA	212.12	305.08	92.96	161.72	279.47	117.75	174.82	235.55	60.73	81.82	214.97	133.15	162.78	265.92	103.14
		TS	193.12	289.79	96.67	139.86	268.28	128.42	151.27	220.54	69.27	51.75	193.62	141.87	133.84	249.18	115.34
		MSII	148.86	266.99	118.13	101.56	243.15	141.59	133.15	209.01	75.86	25.36	180.28	154.92	99.11	225.49	126.38
		SA+TS	249.12	316.87	67.75	187.33	289.66	102.33	199.74	243.19	43.45	113.01	225.45	112.44	192.87	277.72	84.85
	0.5	SA	214.36	314.28	99.92	160.35	288.64	128.29	178.97	247.64	68.67	84.14	225.53	141.39	164.35	278.24	113.89
		TS	193.11	302.47	109.36	140.82	278.16	137.34	160.66	236.48	75.82	67.97	215.85	147.88	145.38	265.48	120.10
		MSII	147.72	275.56	127.84	113.97	263.83	149.86	134.36	216.49	82.13	31.85	200.72	168.87	95.91	234.89	138.98
		SA+TS	256.89	332.54	75.65	186.81	298.04	111.23	206.18	258.86	52.68	115.61	239.33	123.72	187.45	280.36	92.91
	0.3	SA	216.72	324.85	108.13	148.29	291.96	143.67	187.78	265.97	78.19	78.76	236.25	157.49	164.16	285.54	121.38
		TS	189.05	311.61	122.56	134.08	280.22	146.14	160.57	248.22	87.65	60.05	226.02	165.97	133.84	269.42	135.58
		MSII	156.52	290.54	134.02	105.17	272.52	167.35	135.32	231.55	96.23	21.65	206.56	184.91	109.50	257.13	147.63
		SA+TS	244.83	338.20	93.37	189.88	308.05	118.17	221.08	279.46	58.38	114.23	249.78	135.55	194.89	298.61	103.72

Table 7 Average results of different algorithms with high duration variability

N-2	OS	Algorithm	Composite robustness						Single robustness								
			Buffeblack (F1)			Unbuffeblack (F2)			Buffeblack (F3)			Unbuffeblack (F4)			Buffeblack (F5)		
			$\overline{z}_{comp}$	$\overline{z}_{qual}$	$\overline{z}_{stab}$	$\overline{z}_{comp}$	$\overline{z}_{qual}$	$\overline{z}_{stab}$	$\overline{z}_{comp}$	$\overline{z}_{qual}$	$\overline{z}_{stab}$	$\overline{z}_{comp}$	$\overline{z}_{qual}$	$\overline{z}_{stab}$	$\overline{z}_{comp}$	$\overline{z}_{qual}$	$\overline{z}_{stab}$
10	0.7	SA	32.75	44.51	11.76	21.15	39.89	18.74	27.31	35.83	8.52	12.25	33.96	21.71	23.35	37.51	14.16
		TS	28.59	41.76	13.17	14.82	36.78	21.96	25.15	34.64	9.49	7.63	31.49	23.86	18.11	35.73	17.62
		MSII	18.63	36.55	17.92	4.45	30.97	26.52	16.56	29.92	13.36	-1.53	25.88	27.41	10.81	31.56	20.75
		SA+TS	39.29	48.44	9.15	27.05	42.43	15.38	33.59	40.36	6.77	21.50	37.76	16.26	30.88	42.76	11.88
	0.5	SA	30.31	42.73	12.42	14.33	36.67	22.34	24.95	34.52	9.57	7.02	31.66	24.64	19.91	36.35	16.44
		TS	25.58	40.23	14.65	8.99	34.92	25.93	20.14	32.12	11.98	-0.99	26.28	27.27	14.79	34.07	19.28
		MSII	12.19	30.34	18.15	0.42	29.83	29.41	9.33	24.36	15.03	-7.38	23.74	31.12	8.25	30.58	22.33
		SA+TS	35.97	45.83	9.86	25.67	42.52	16.85	29.33	38.08	8.75	15.46	35.19	19.73	28.17	41.25	13.08
	0.3	SA	32.27	45.51	13.24	17.49	41.15	23.66	28.20	38.04	9.84	7.48	37.16	29.68	19.82	39.37	19.55
		TS	28.14	42.89	14.75	11.29	38.38	27.09	20.94	34.16	13.22	3.41	33.72	30.31	16.43	37.85	21.42
		MSII	17.36	37.92	20.56	2.16	32.98	30.82	11.93	30.32	18.39	-7.76	27.18	34.94	4.41	32.22	27.81
		SA+TS	36.06	48.08	12.02	24.79	44.67	19.88	32.02	41.49	9.47	12.72	38.25	25.53	25.79	42.63	16.84
30	0.7	SA	84.28	133.79	49.51	44.85	120.83	75.98	68.23	97.48	29.25	12.52	93.14	80.62	51.09	112.87	61.78
		TS	64.31	116.46	52.15	25.39	108.74	83.35	53.31	89.96	36.65	-2.72	85.56	88.28	32.77	99.73	66.96
		MSII	37.44	99.08	61.64	1.19	90.81	89.62	41.39	81.28	39.89	-20.77	72.48	93.25	12.52	87.09	74.57
		SA+TS	109.23	151.95	42.72	66.13	132.52	66.39	88.45	113.56	25.11	36.85	104.53	67.68	75.34	127.42	52.08
	0.5	SA	82.99	136.74	53.75	43.74	122.39	78.65	70.58	106.52	35.94	5.09	93.84	88.75	52.19	118.71	66.52
		TS	65.89	125.52	59.63	26.02	112.94	86.92	59.22	99.48	40.26	-5.11	89.28	94.39	35.88	107.66	71.78
		MSII	37.99	103.45	65.46	3.23	95.31	92.08	31.37	80.04	48.67	-30.17	78.36	108.53	8.47	88.82	80.35
		SA+TS	106.56	155.53	48.97	70.56	140.83	70.27	90.57	119.75	29.18	41.65	117.52	75.87	70.75	129.33	58.58
	0.3	SA	64.57	124.69	60.12	20.24	104.97	84.73	54.11	92.18	38.07	-14.23	87.08	101.31	30.23	103.67	73.44
		TS	43.24	111.17	67.93	6.82	94.66	87.84	41.70	82.84	41.14	-30.38	79.19	109.57	9.75	89.24	79.49
		MSII	21.91	97.76	75.85	-14.90	83.15	98.05	21.58	71.96	50.38	-48.01	68.74	116.75	-7.92	78.05	85.97
		SA+TS	90.13	141.28	51.15	43.17	119.43	76.26	66.44	98.48	32.04	2.69	95.91	93.22	47.81	112.29	64.48
50	0.7	SA	146.14	261.72	115.58	84.96	241.91	156.95	137.28	226.75	89.47	19.49	183.18	163.69	103.47	235.29	131.82
		TS	109.29	243.26	133.97	68.11	230.89	162.78	111.73	212.36	100.63	-2.54	170.34	172.88	82.85	226.41	143.56
		MSII	70.14	219.35	149.21	29.95	207.64	177.69	78.44	184.28	105.84	-31.56	156.15	187.71	33.88	194.96	161.08
		SA+TS	190.72	282.48	91.76	115.23	252.06	136.83	168.83	238.16	69.33	66.79	209.64	142.85	126.16	244.35	118.19
	0.5	SA	153.17	272.62	119.45	87.07	249.32	162.25	130.61	225.96	95.35	28.77	203.29	174.52	92.92	238.48	145.56
		TS	124.84	263.18	138.34	65.79	239.97	174.18	110.21	216.24	106.03	9.22	189.48	180.26	70.28	223.57	153.29
		MSII	86.08	238.56	152.48	33.06	222.72	189.66	78.57	195.39	116.82	-22.03	171.16	193.19	40.54	208.22	167.68
		SA+TS	188.45	288.42	99.97	115.65	258.41	142.76	155.32	242.51	87.19	66.39	216.76	150.37	124.62	249.98	125.36
	0.3	SA	147.09	279.45	132.36	84.06	255.78	171.72	135.22	239.34	104.12	25.97	208.43	182.46	91.15	247.42	156.27
		TS	117.79	264.36	146.57	49.97	238.32	188.35	113.55	227.53	113.98	10.43	202.26	191.83	68.05	232.78	164.73
		MSII	96.39	252.87	156.48	23.17	218.91	195.74	87.63	208.18	120.55	-18.06	188.38	206.44	29.36	214.82	185.46
		SA+TS	180.81	299.32	118.51	108.22	267.45	159.23	157.85	248.16	90.31	58.76	219.68	160.92	120.09	258.03	137.94

Additionally, we derive the upper bounds on quality robustness and solution robustness for a given schedule to evaluate the performance of our algorithms versus our performance measures. The upper bound on quality robustness (Z_{qual}^{upper}) is the maximized project NPV obtained by a modified version of the SA algorithm in Section 4.2, assuming a static and deterministic environment. Meanwhile, the upper bound on solution robustness (Z_{stab}^{upper}) is the minimized expected penalty cost obtained by the EPC procedure in Section 3.1. We then design a relative percentage index, $\Delta\overline{Z}_{comp}$, to measure the magnitude of composite robustness in the realized schedule versus the upper bound schedule. The new index is calculated as follows:

$$\Delta\overline{Z}_{comp} = (Z_{qual}^{upper} - \overline{Z}_{qual})\Big/ Z_{qual}^{upper} - (Z_{stab}^{upper} - \overline{Z}_{stab})/Z_{stab}^{upper} \tag{9}$$

As noted in Table 8, $\Delta\overline{Z}_{comp}$ in Schedule F1 is lower than that in four other schedules other schedules for different three levels of the duration variability. Meanwhile, the two-stage algorithm (SA+TS) has a lowest $\Delta\overline{Z}_{comp}$ compablack with SA, TS, and MSII. These results further demonstrate the superiority of combination between solution robustness and quality robustness as well as the effectiveness of the proposed two-stage algorithm for generating composite robust schedules. All the solutions are uploaded in the Mathworks WebSet.[2]

Table 8 Average results of $\Delta\overline{Z}_{comp}$ for different algorithms

N-2	OS	Algorithm	Composite robustness						Single robustness								
			Buffeblack (F1)			Unbuffeblack (F2)			Buffeblack (F3)			Unbuffeblack (F4)			Buffeblack (F5)		
			L	*M*	*H*	*L*	*M*	*H*	*L*	*M*	*H*	*L*	*M*	*H*	*L*	*M*	*H*
10	0.7	SA	0.41	0.94	1.31	1.48	3.22	3.85	0.46	1.09	1.85	2.15	4.00	5.03	0.84	2.34	3.11
		TS	0.62	1.08	2.00	1.97	3.76	4.53	0.67	1.49	2.16	2.75	4.61	5.20	1.10	3.03	3.48
		MSII	1.18	2.29	3.02	2.35	4.40	5.31	1.08	2.63	3.32	3.40	5.76	6.16	1.47	3.81	4.75
		SA+TS	0.36	0.59	1.20	1.04	2.12	3.10	0.28	0.93	1.58	1.71	3.11	4.24	0.65	1.70	2.56
	0.5	SA	0.56	1.08	1.78	1.72	3.62	4.78	0.62	1.42	2.34	2.73	4.62	5.39	1.02	2.71	3.39
		TS	0.77	1.46	2.39	2.27	4.33	5.65	1.01	1.76	2.91	3.51	5.14	6.09	1.38	3.34	4.09
		MSII	1.32	2.72	3.22	2.85	4.90	6.55	1.35	3.30	3.88	4.11	6.23	7.04	1.80	4.43	4.86
		SA+TS	0.44	0.71	1.54	1.37	2.70	3.40	0.51	1.29	1.69	2.15	3.15	4.18	0.84	1.99	2.52
	0.3	SA	0.51	1.16	2.09	2.12	3.98	5.15	0.79	1.60	2.95	3.31	4.95	6.15	1.34	2.67	3.79
		TS	0.82	1.57	2.40	3.00	4.59	6.18	1.37	2.10	3.42	4.21	6.01	6.84	1.77	3.44	4.87
		MSII	1.68	2.74	3.65	3.62	6.06	7.66	1.82	3.34	4.95	5.22	6.88	8.01	2.39	4.98	5.89
		SA+TS	0.24	0.72	1.49	1.89	3.12	4.09	0.53	1.24	2.10	2.61	3.58	4.43	0.78	2.24	3.01
30	0.7	SA	0.81	1.41	1.64	2.03	2.92	4.10	1.14	1.93	2.69	2.71	3.73	5.07	1.70	2.48	3.50
		TS	1.07	1.63	1.85	2.50	3.32	4.31	1.43	2.30	3.17	3.04	4.38	5.55	2.13	3.00	3.89
		MSII	1.28	2.29	2.40	2.91	3.72	4.91	1.95	2.96	3.65	3.91	4.58	5.99	2.66	3.29	4.29
		SA+TS	0.43	0.90	1.29	1.78	2.46	3.58	1.02	1.46	2.13	2.18	3.22	4.60	1.56	2.10	2.98
	0.5	SA	0.66	1.63	2.10	2.68	3.58	5.10	1.34	2.35	3.24	3.16	4.54	5.96	2.18	3.06	4.25
		TS	0.93	1.97	2.44	3.09	4.20	5.74	1.81	2.81	3.72	3.94	5.20	6.39	2.63	3.57	4.68
		MSII	1.59	2.74	3.14	3.61	4.87	6.20	2.15	3.68	4.24	4.88	5.69	7.46	3.08	4.11	5.38
		SA+TS	0.46	0.96	1.55	2.30	2.99	4.42	1.12	1.82	2.81	2.64	3.76	4.93	1.99	2.37	3.63
	0.3	SA	0.52	1.84	2.52	3.46	4.96	7.24	1.69	3.40	4.43	4.49	6.13	7.87	2.58	4.18	5.81
		TS	0.92	2.30	3.33	3.95	5.91	8.07	2.07	3.79	4.79	5.01	6.96	8.70	3.01	4.62	6.42
		MSII	1.89	3.33	3.71	4.48	6.68	8.81	2.79	4.57	5.87	6.60	7.66	9.28	3.91	5.88	7.27
		SA+TS	0.33	1.16	2.01	2.98	4.26	6.18	1.48	2.60	3.63	3.42	5.07	6.47	2.27	3.45	4.73
50	0.7	SA	0.72	1.26	1.98	1.86	2.82	3.61	0.82	1.86	2.58	2.46	3.31	4.00	1.38	2.29	3.25
		TS	0.96	1.54	2.25	2.11	2.92	4.07	1.02	2.25	2.97	3.08	3.55	4.25	1.58	2.68	3.50
		MSII	1.36	1.80	2.46	2.71	3.46	4.31	1.34	2.59	3.24	3.74	4.07	4.65	2.02	3.01	4.06
		SA+TS	0.42	0.74	1.61	1.56	2.15	3.27	0.52	1.46	2.18	2.06	2.73	3.44	1.12	1.81	2.77
	0.5	SA	0.79	1.57	2.50	2.51	3.40	4.62	1.20	2.41	3.16	3.03	4.00	5.14	1.86	2.96	4.10
		TS	1.10	1.83	2.87	2.75	3.73	5.03	1.48	2.75	3.80	3.48	4.24	5.36	2.16	3.20	4.39
		MSII	1.71	2.09	3.28	3.34	4.17	5.58	1.79	3.42	4.33	3.98	4.96	5.84	2.64	3.90	4.90
		SA+TS	0.38	1.02	2.19	2.16	2.82	3.96	0.83	1.57	2.48	2.50	3.39	4.32	1.45	2.27	3.41
	0.3	SA	0.80	1.69	2.84	2.72	3.79	5.42	1.42	2.75	3.76	3.17	4.57	5.85	2.10	3.26	4.46
		TS	1.06	2.07	3.31	2.95	4.24	5.68	1.72	2.94	4.53	3.85	4.97	6.25	2.44	3.78	4.95
		MSII	1.73	2.36	3.60	3.11	4.82	6.33	2.17	3.84	5.19	4.51	5.52	6.87	2.74	4.28	5.72
		SA+TS	0.62	1.41	2.68	2.79	4.12	5.90	1.60	2.37	3.63	3.45	4.80	6.31	2.24	3.31	5.02

[2] https://ww2.mathworks.cn/matlabcentral/fileexchange/67333-computational-results

6 Conclusion and further research

This study considers the resource constrained max-NPV project scheduling problem with stochastic activity durations and aims at generating a proactive baseline schedule with composite robustness. First, the project NPV and the EPC are proposed to measure quality robustness and solution robustness, respectively. Secondly, the EPC procedure is designed to generate a solution-robust schedule by iteratively allocating time buffers in front of the activities, which can protect the payment plan from disruptions as well as possible. Thirdly, a bi-objective scheduling model is proposed under the objectives of the project NPV maximization (quality robustness) and the schedule stability maximization (solution robustness), and a two-stage algorithm that integrates SA and TS is developed to solve the problem. Finally, an extensive computational experiment is constructed to verify the benefits of the composite robust schedules that strike a balance between quality solution and robustness solution as well as the effectiveness of the proposed two-stage algorithm for generating robust schedules compablack with three other algorithms, namely, SA, TA, and MSII.

Notably, we only generate a feasible resource flow network in this study. However, different resource flow networks applied to a given schedule can result in different time buffer allocations. Therefore, the impact of resource allocation on the time buffer allocation is an interesting topic for future works.

Acknowledgements This work was supported by the National Natural Science Foundation of China [Grant numbers: 71271097, 71701073, and 71701067]. We would also like to acknowledge the Research Center for Operations Management of the KU Leuven for providing support to Yangyang Liang and Tian Wang as visiting research associates.

References

1. Artigues C and Roubellat F (2000). A polynomial activity insertion algorithm in a multi-resource schedule with cumulative constraints and multiple modes. *European Journal of Operational Research* **127**(2), 297-316.
2. Artigues C, Michelon P and Reusser S (2003). Insertion techniques for static and dynamic resource-constrained project scheduling. *European Journal of Operational Research* **149**(2): 249-267.
3. Baroum S M and Patterson J H (1996). The development of cash flow weight procedures for maximizing the net present value of a project. *Journal of Operations Management* **14**(3): 209-227.
4. Bey R B, Doersch R H and Patterson J H (1981). The net present value criterion: its impact on project scheduling. *Project Management Quarterly* **12**(2): 35-45.
5. Bouleimen K and Lecocq H (2003). A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version. *European Journal of Operational Research* **49**(2): 268-281.
6. Buss A H and Rosenblatt M J (1997). Activity delay in stochastic project networks. *Operations Research* **45**(1): 126-139.
7. Creemers S, Leus R and Lambrecht M (2010). Scheduling markovian pert networks to maximize the net present value. *Operations Research Letters* **38**(1): 51-56.

8. Demeulemeester E and Herroelen W (1992). A branch-and-bound procedure for the multiple resource-constrained project scheduling problem. *Management science* **38**(12): 1803-1818.
9. Demeulemeester E and Herroelen W (2011). Robust project scheduling. *Foundations & Trends in Technology Information & Operations Management* **3**(3-4): 201-376.
10. Doersch R H and Patterson J H (1977). Scheduling a project to maximize its present value: a zero-one programming approach. *Management Science* **23**(8): 882-889.
11. Elmaghraby S E (2005). On the fallacy of averages in project risk management. *European Journal of Operational Research* **165**(2): 307-313.
12. Gu H, Schutt A, Stuckey P J, Wallace M G and Chu G. (2015). Exact and heuristic methods for the resource-constrained net present value problem. *In Handbook on Project Management and Scheduling (Schwindt C and Zimmermann J eds.), Springer*, 299-318.
13. Hartmann S and Briskorn D (2008). A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research* **207**(1):1-14.
14. He Z, Wang N, Jia T and Xu Y. (2009). Simulated annealing and tabu search for multi-mode project payment scheduling. *European Journal of Operational Research* **198**(3): 688-696.
15. He Z, Liu R and Jia T (2012). Metaheuristics for multi-mode capital-constrained project payment scheduling. *European Journal of Operational Research* **223**(3): 605-613.
16. Herroelen W, Dommelen P and Demeulemeester E (1997). Project network models with discounted cash flows a guided tour through recent developments. *European Journal of Operational Research* **100**(1): 97-121.
17. Herroelen W and Leus R (2001). On the merits and pitfalls of critical chain scheduling. *Journal of operations management* **19**(5): 559-577.
18. Herroelen W and Leus R (2004). The construction of stable project baseline schedules. *European Journal of Operational Research* **156** (3): 550-565.
19. Herroelen W and Leus R (2005). Project scheduling under uncertainty: Survey and research potentials. *European journal of operational research* **165**(2): 289-306.
20. Hu X, Cui N and Demeulemeester E (2015). Effective expediting to improve project due date and cost performance through buffer management. *International Journal of Production Research* **53**(5):1460-1471.
21. Icmeli O and Erenguc S S (1994). A tabu search procedure for the resource constrained project scheduling problem with discounted cash flows. *Computers & Operations Research* **21**(8): 841-853.
22. Kahn A B (1962). Topological sorting of large networks. *Communications of the ACM* **5**(11): 558-562.
23. Lambrechts O, Demeulemeester E and Herroelen W (2008). A tabu search procedure for developing robust pictive project schedules. *International Journal of Production Economics* **111**(2): 493-508.
24. Leus R (2004). The generation of stable project plans. *Quarterly Journal of the Belgian French & Italian Operations Research Societies* **2**(3): 251-254.
25. Leus R and Herroelen W (2004). Stability and resource allocation in project planning. *IIE Tranictions* **36** (7): 667-682.

26. Leyman P and Vanhoucke M (2016). Payment models and net present value optimization for resource-constrained project scheduling. *Computers & Industrial Engineering* **91**:139-153.
27. Mika M, Waligóra G and Węglarz J (2005). Simulated annealing and tabu search for multi-mode resource-constrained project scheduling with positive discounted cash flows and different payment models. *European Journal of Operational Research* **164** (3): 639-668.
28. Mika M, Waligóra G and Węglarz J (2008). Tabu search for multi-mode resource-constrained project scheduling with schedule-dependent setup times. *European Journal of Operational Research* **187**(3): 1238-1250.
29. Mohaghar A, Khoshghalb A, Rajabi M and Khoshghalb A (2016). Optimal delays, safe floats, or release dates? Applications of simulation optimization in stochastic project scheduling. *Procedia Economics & Finance* **39**:469-475.
30. Neumann K, Schwindt C and Zimmermann J (2003). Order-based neighborhoods for project scheduling with nonregular objective functions. *European Journal of Operational Research* **149**(2): 325-343.
31. Russell A H (1970). Cash flows in networks. *Management Science* **16**(5): 357-373.
32. Skorinkapov J (1990). Tabu search applied to the quadratic assignment problem. *Orsa Journal on Computing* **2**(1): 33-45.
33. Sobel M J, Szmerekovsky J G and Tilson V (2009). Scheduling projects with stochastic activity duration to maximize expected net present value. *European Journal of Operational Research* **198**(3): 697-705.
34. Tantisuvanichkul V and Kidd M (2011). Maximizing net present value a review through literature. *International Proceedings of Economics Development and Research* **15**(2): 93-97.
35. Tukel O I, Rom W O and Eksioglu S D (2006). An investigation of buffer sizing techniques in critical chain scheduling. *European Journal of Operational Research* **172**(2): 401-416.
36. Vonder S V D, Demeulemeester E, Herroelen W and Leus R (2005). The use of buffers in project management: The trade-off between stability and makespan. *International Journal of Production Economics* **97**(2): 227-240.
37. Vonder S V D, Demeulemeester E, Herroelen W and Leus R (2006). The trade-off between stability and makespan in resource-constrained project scheduling. *International Journal of Production Research* **44**(2): 215-236.
38. Vonder S V D, Demeulemeester E and Herroelen W (2008). Proactive heuristic procedures for robust project scheduling: An experimental analysis. *European Journal of Operational Research* **189**(3): 723-733.
39. Wang J (2005). Constraint-based schedule repair for product development projects with time-limited constraints. *Journal of Production Economics* **95**(3): 399-414.
40. Waligóra G (2014). Discrete-continuous project scheduling with discounted cash inflows and various payment model: a review of recent results. *Annals of Operations Research* **213**(1): 319-340.
41. Waligóra G (2014). Simulated annealing and tabu search for discrete-continuous project scheduling with discounted cash flows. *RAIRO-Operations Research* **48**(1): 1-24.

42. Waligóra G (2016). Comparative analysis of some metaheuristics for discrete-continuous project scheduling with activities of identical processing rates. *Asia-Pacific Journal of Operational Research*, **33** (3): 1650015.

43. Wang Q, Zhang X, Mahadevan S and Deng Y (2015). Solving the longest path problem in directed acyclic graphs based on amoeba algorithm. *International Journal of Unconventional Computing* **11**(2): 147-163.

44. Wiesemann W, Kuhn D and Rustem B (2010). Maximizing the net present value of a project under uncertainty. *European Journal of Operational Research* **202** (2): 356-367.

45. Zheng W, He Z and Wang N (2017). Proactive and reactive resource-constrained max-NPV project scheduling with random activity duration. *Journal of the Operational Research Society* **2**(2):1-12.

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