

Fikri KUCUKSAYACIGIL, Gündüz ULUSOY

Hybrid genetic algorithm for bi-objective resource-constrained project scheduling

© Higher Education Press 2020

Abstract In this study, we considered a bi-objective, multi-project, multi-mode resource-constrained project scheduling problem. We adopted three objective pairs as combinations of the net present value (NPV) as a financial performance measure with one of the time-based performance measures, namely, makespan (C_{\max}), mean completion time (MCT), and mean flow time (MFT) (i.e., $\min C_{\max}/\max NPV$, $\min MCT/\max NPV$, and $\min MFT/\max NPV$). We developed a hybrid non-dominated sorting genetic algorithm II (hybrid-NSGA-II) as a solution method by introducing a backward–forward pass (BFP) procedure and an injection procedure into NSGA-II. The BFP was proposed for new population generation and post-processing. Then, an injection procedure was introduced to increase diversity. The BFP and injection procedures led to improved objective functional values. The injection procedure generated a significantly high number of non-dominated solutions, thereby resulting in great diversity. An extensive computational study was performed. Results showed that hybrid-NSGA-II surpassed NSGA-II in terms of the performance metrics hypervolume, maximum spread, and the number of non-dominated solutions. Solutions were obtained for the objective pairs using hybrid-NSGA-II and three different test problem sets with specific properties. Further analysis was performed by employing cash balance, which was another financial performance measure of practical importance. Several managerial insights and extensions for further research were presented.

Keywords backward–forward scheduling, hybrid bi-

objective genetic algorithm, injection procedure, maximum cash balance, multi-objective multi-project multi-mode resource-constrained project scheduling problem

1 Introduction

With the changing business paradigm in recent decades, great emphasis has been placed on project-based work. We have observed an increase in the number of companies that provide engineering, managerial, and financial services and technological firms that structure themselves as project organizations. In line with these developments, the relevance and importance of effectively dealing with multiple simultaneous projects have increased. The on-time completion of these projects by meeting the quality requirements without exceeding the allocated budget is a major task. This task provides a great challenge for project owners and managers. Project planning and scheduling are the major tools used to meet this challenge. Project scheduling serves the engineering and management functions of the organization, as it is used to generate schedules in line with the organizational objective(s). Project scheduling, as a tool, supports the project manager in the evaluation of different alternatives and creates a time, resource, and risk management plan for the project implementation. The resulting project schedule is the so-called baseline schedule, which not only represents the initial plan but also constitutes the backbone for the control and monitoring of the project during its implementation. Project scheduling serves as the project manager that determines ways to minimize the deviations from the planned schedule and cost throughout the project implementation.

The core problem underlying project scheduling in project organizations is the resource-constrained project scheduling problem (RCPS). The RCPS is a complex, NP (nondeterministic polynomial)-hard problem (Blazewicz et al., 1983). In recent decades, an extensive amount of work has been accomplished for developing exact and

Received December 30, 2018; accepted February 15, 2020

Fikri KUCUKSAYACIGIL
Industrial Engineering Department, University of Arkansas, Fayetteville,
AR 72701, USA

Gündüz ULUSOY (✉)
Industrial Engineering Department, Sabanci University, Istanbul, Turkey
E-mail: gunduz@sabanciuniv.edu

heuristic algorithms for the solution of RCPSP and its extensions, such as multi-mode RCPSP (MRCPSP), multi-project RCPSP (RCMPSP), and multi-project, multi-mode RCPSP (MRCMPSP) (Özdamar and Ulusoy, 1996; Herroelen et al., 1998; Brucker et al., 1999; Kolisch and Padman, 2001; Herroelen and Leus, 2005; Hartmann and Briskorn, 2010). A rich body of literature in multi-project scheduling under resource scarcity is available.

Apart from task complexity, we have observed relational complexity in project management resulting from multiple stakeholders with conflicting interests; this situation can lead to disagreements about project goals and priorities among tasks and features of the project outcome (Pich et al., 2002). A multi-objective programming approach can be employed to handle relational complexity. This study deals with the bi-objective MRCMPSP. The minimization of the makespan of projects ($\min C_{\max}$) is a common and frequently used objective in project scheduling. This objective is crucial because it allows—among other things—the early release of renewable resources for subsequent projects and helps prevent the possible violation of imposed deadlines (Demeulemeester and Herroelen, 2002). The maximization of the net present value (NPV) of projects ($\max NPV$) is another significant objective in project scheduling. Many researchers and practitioners prefer the NPV as a financial performance measure because it can effectively reflect the financial aspects of the decision environment (Gu et al., 2015). The objective is to minimize the NPV when only the costs are involved. The parallel processing of projects, namely, concurrency, in construction environments has gained great acceptance since the 1990s. Concurrent engineering, a term referring to the parallel execution of tasks, has been used by practitioners to minimize project lead times (Ahmad et al., 2016). In addition to C_{\max} and NPV , the project manager may also be interested in minimizing the mean flow time ($\min MFT$) of individual projects to ensure that the mean throughput times of projects are reduced, thereby leading to a general reduction in the work-in-progress (Herroelen and Leus, 2001). The objective $\min MFT$ also reflects the increasing attention of the contractors to reduce non-value adding activities further and wasted time and resources because the competition is fierce in the world today (Sacks et al., 2017). The minimization of the mean completion time ($\min MCT$) for individual projects can be considered another relevant time-based objective. A decision maker may seek a project schedule that strategically uses renewable resources, thereby leading to acceptable project completion times. When a contractor carries out multiple projects, each of which pertains to different clients, a key factor of success for the contractor is to meet their individual time-based requirements. The MCT can be closely associated with customer satisfaction and may lead to favorable cash profiles. We do not consider deadlines for projects and penalties for their violations because the minimization of

MCT and C_{\max} explicitly refers to project terminations as soon as possible.

The financial impact of reducing the duration of a project is essential information, which the decision maker uses in the project-scheduling phase. Vanhoucke (2009) has presented a study on the trade-off between C_{\max} and NPV for RCPSP. In that formulation, a soft deadline constraint is imposed to allow a project deadline violation at a certain penalty cost. All the payments and receipts throughout the duration of an activity are discounted up to the completion time of the activity to represent the cash flow associated with it. The objective function is the sum of the discounted cash flows of the activities and the penalty cost. This framework is considered a multi-objective optimization model because the C_{\max} and NPV are included in the objective function. Khalili et al. (2013) have simultaneously considered the bi-objective problem of $\min C_{\max}$ and $\max NPV$ for RCPSP by approximating the Pareto front. Two meta-heuristic algorithms are employed for solving the bi-objective RCPSP: Multi-population genetic algorithm (GA) (Cochran et al., 2003) and two-phase sub-population GA (Chang et al., 2005).

The analysis of the trade-off between C_{\max} and NPV is a problem of interest in project scheduling. C_{\max} and NPV intuitively conflict, but they can be mutually supporting under certain conditions. Smith-Daniels and Aquilano (1987) have demonstrated this notion in a case where the resources are renewable, and a lump sum payment is made at the termination of the project. Activity costs are dependent on activity durations and are incurred at the start of activities. Ulusoy and Özdamar (1995) have investigated the mutual support of these two objectives under certain circumstances. The authors have considered two different models. In the first one, activity-related cash outflows take place at the start of the activity, and a lump sum payment occurs at the completion of the project. The activity-related costs depend on the total resource demand required by the activity to complete it. The second model is a multi-mode version of the first one.

In addition to the trade-off between C_{\max} and NPV , the trade-offs between MFT and NPV and between MCT and NPV are relevant when managing multiple projects. The reason is as follows: C_{\max} , by definition, only refers to the completion time of the last project and, as such, is an aggregate measure over all the projects. However, each project is an entity itself, possibly with different owners and project managers. Hence, measures to follow individual projects in a multi-project environment are important.

In this study, we investigate three bi-objective cases for MRCMPSP in detail: (i) the minimization of C_{\max} and the maximization of NPV ($\min C_{\max}/\max NPV$), (ii) the minimization of MFT and the maximization of NPV ($\min MFT/\max NPV$), and (iii) the minimization of MCT and the maximization of NPV ($\min MCT/\max NPV$). We aim to gain a wide perspective of the decision problem by dealing with three different bi-objective models.

The contributions of this study are threefold. (i) We cope with a niche area in MRCPSP, which includes multi-project and multi-objective aspects. The literature review below reveals a few studies in this area. We propose $\min MFT$, $\min MCT$, $\max NPV$, and $\min C_{\max}$ as objectives in this complicated problem structure. We further analyze our results from the perspective of maximum cash balance (CB) (i.e., the maximum cumulative gap between cash inflow and outflow) for a multi-project scheduling environment. (ii) We reveal important managerial insights concerning the preferences among the objectives used for multi-project scheduling. Moreover, we elaborate on the effects of changing renewable resource capacities on schedules, that is, increasing or decreasing the activity progress rates. Lastly, we analyze the CB diagrams of different schedules to find the possible interactions between renewable resource capacities and CB . (iii) Our aim is to develop an algorithm for generating solutions at acceptable computational times for arriving at the conclusions stated in (i) and (ii) by comparing the solution procedure proposed here. Although non-dominated sorting genetic algorithm (NSGA)-II is used in this work, we propose two different improvements: One for local search and post-processing using backward–forward pass (BFP) procedure to find better solutions and the other one for enlarging the set of non-dominated solutions by an injection procedure. Our main focus is not to propose a competitive algorithm for scheduling tasks but to improve an existing algorithm and its capabilities.

In the next sections, we first present the relevant studies from the literature, followed by the mathematical programming formulation of our problem. Thereafter, we explain the adopted solution methodology and extension of that methodology with BFP and injection procedures. We then carry out an extensive computational study that discusses the results regarding the impacts of BFP and the injection procedures and the relationships among the three bi-objective problems. Finally, we conclude the study by summarizing the key findings and presenting several managerial insights and future research avenues.

2 Literature review

Efforts have been recently exerted to bring theory and practice together in project scheduling for dealing with the real-life concerns of project practitioners. This situation has drawn the attention of researchers to the modeling and solution of—among others—MRCMPSP and multi-objective RCPSPs. Wauters et al. (2016) have reported about the results of the nine algorithms for multi-project scheduling in the final competition of the MISTA 2013 Challenge. All submitted algorithms were heuristic procedures in which some included exact components. The primary objective of the Challenge was to minimize the total project delay. The secondary objective was to

minimize the total project duration, which was employed as a tiebreaker. To the best of our knowledge, a few studies have simultaneously analyzed RCPSP with its multi-objective and multi-project aspects. The current literature can be classified into three approaches:

(i) Representing the multiple objectives in a single objective function and solving the problem as a single objective optimization problem;

(ii) Treating the objectives in vector form and seeking an approximation set to the Pareto front;

(iii) Approaching the problem in an interactive way, wherein the decision maker guides the search through the feasible solutions by the choice of parameters, such as the weights of the involved multiple objectives.

All the studies reported below treat single mode problems unless otherwise stated, that is, they are multi-objective RCMPSPs.

The study by Liu and Wang (2009) is an example of the first approach. The authors have aimed to minimize the overall C_{\max} of projects and the flow time of individual projects by combining them in a single weighted objective function. Individual projects are also assigned weights to represent the importance of these projects to the decision maker. The abovementioned authors have implemented a greedy search algorithm to find effective solutions under resource constraints. Xu and Feng (2014) have developed a particle swarm optimization algorithm for MRCMPSP under a fuzzy random environment. The weighted combination of the overall C_{\max} and individual prioritized project C_{\max} , project cost consisting of fixed, variable, and crashing costs of activities, and quality of projects are accepted as objectives. Then, they were combined into a single formulation with a weighted sum approach. Wang et al. (2014) have proposed a cloud GA to solve a multi-objective RCPSP with time, cost, quality, and robustness as the objectives. The objective function is defined as the weighted sum of the utility function of each objective. Riise et al. (2016) have studied the practical application of project scheduling. The operational surgery scheduling problem is modeled as MRCMPSP with generalized time and application-specific additional constraints and is solved with an iterative search algorithm. Taking an arriving patient as a project, the authors have considered numerous performance measures represented as a weighted sum, such as the number of unscheduled patients, patient waiting times, finishing early in the day (e.g., C_{\max}), and child early objective (i.e., limiting children surgery to the morning hours).

We group the following recently published papers under the second approach (Kucuksayacigil and Ulusoy (2018) provides an extensive review). Gang et al. (2013) have solved multi-mode and multi-project resource allocation problems with a bi-level approach under stochastic activity durations and costs defined as the sum of resource costs and the total tardiness penalty for the multiple projects. The decision maker at the upper level, that is, the company

manager, seeks to allocate the company resources to multiple projects at the lowest total cost where the costs are defined as above. At the lower level, each project manager tries to schedule the allocated resources for minimizing the duration of the project that they manage. Singh (2014) has solved the problem via a hybrid method consisting of priority rules and an analytical hierarchy process application used for assigning weights to projects. The overall C_{\max} and cost of multi-projects are considered as objectives. Can and Ulusoy (2014) have created a hierarchical model for the problem, as proposed earlier by Speranza and Vercellis (1993), which regards each project as a macro-activity and solves the NPV maximization problem. The authors have implemented a post-processing scheme to minimize C_{\max} . Moreover, the authors have developed an exact solution method and a GA for solving the problem. Shahsavari et al. (2015) have considered three objectives in a resource-constrained multi-project problem setting. The objectives are the minimization of the overall C_{\max} of the projects, the total cost associated with the resources, and the variability of the resource usage. The authors have employed three self-adaptive GAs to generate non-dominated solutions. 180 problems are solved by an evaluation using five performance metrics.

No attempt has been made regarding the solution for multi-objective MRCMPSPs using the interactive multi-objective approach. Gagnon et al. (2005) have introduced a triple objective model for RCPSP that considers C_{\max} , resource availability cost, and the amount of each resource type allocated as objectives. The authors have used the tabu search to obtain non-dominated solutions. All non-dominated solutions found during the search are stored in a dominance tree, and they are available to the project manager for examination.

Our exhaustive literature review points out that RCPSP has been studied with several modifications, such as the transfer times of resources and the dynamic arrival of projects, due to the demands by industrial partners. Multi-skill RCPSP has gained increasing attention from research practitioners. The stochastic nature of problem parameters and reactive/proactive scheduling is another aspect that has drawn attention. The decentralized scheduling and rework possibility of activities are other significant research themes encountered in the literature survey. The literature review indicates that solution techniques based on the Pareto front is much more pervasive than those combining objectives into a single expression. Numerous metaheuristic methods (and hybrid forms) have been proposed and implemented. NSGA-II shows superior performance in many cases and gains appreciation from research practitioners. The review also indicates that NSGA-II and other evolutionary algorithms have been constantly improved by deriving new modules, integrating heuristic methods, and undergoing optimization procedures. The abovementioned

review discloses that the literature addressing the multi-objective MRCMPSP is scarce. The bi-objective pairs ($\min MFT/\max NPV$) and ($\min MCT/\max NPV$) have not been investigated before even in a single project decision environment. This work also aims to fill that gap in the literature for these types of decision problems.

3 Mathematical formulation of the problem

As previously stated, we focus on the bi-objective MRCMPSP in this work. MRCMPSP is a combinatorial optimization problem, and it can be described as follows: $|P|$ projects exist, each of which has $|J_p|$ activities, excluding the dummy source and sink activities. Each activity of a project utilizes $|R|$ different renewable and $|N|$ non-renewable resources and has $|M_{pj}|$ execution modes. The mode m of an activity reflects the technological mix selected to execute that activity. Here, technological mix is defined as a combination of resources, such as machines, skilled workers, process lines, and suppliers. A likely decrease in the duration of the corresponding activity is expected with the increase of the technological mix level of a mode as well as an increase in the associated cost. Each mode m of an activity has a duration d_{pjm} and renewable and non-renewable resource requirements \hat{r}_{pjm_r} and \hat{n}_{pjm_n} , respectively. The dummy activities have zero duration and resource requirements. The activities are assumed to be non-preemptive, that is, an activity cannot be interrupted whenever it is executed. The project activities have precedence relations. Here, we assume finish-to-start precedence relations with zero-time lags. No precedence relations are assumed to exist among projects. Renewable and non-renewable resources have capacities, which should not be exceeded for a schedule to be feasible. The capacities of renewable resources, \bar{r}_{r_t} , are expressed as units per period and can vary over the time horizon. By contrast, the capacities of non-renewable ones, \bar{n}_n , are specified over the whole project duration. If the resource capacities are specified a priori, then a resource portfolio has already been determined for a given budget. The problem lies in determining the schedule, start of activity, and completion times as well as their execution modes to ensure that the precedence constraints are satisfied, the resource capacities are not exceeded, and the problem objectives are simultaneously optimized.

The precedence relations among activities are demonstrated with a directed acyclic activity-on-node graph. The multiple projects are represented as a composite project network in general with dummy source and sink nodes. Table 1 provides the notation for the mathematical formulation. Equations (1) to (7) exhibit the mathematical formulation for the problem denoted by MF. This formulation is an extension of the single objective formulation given by Talbot (1982).

Table 1 Notations for the mathematical formulation

Notations	Definitions
H, t	Time horizon and time period index, $t = 1, \dots, H$
P, p	Set of projects and project index, $p = 1, \dots, P $
J_p, j	Set of activities in project p and activity index, $j = 1, \dots, J_p $ for project p
M_{pj}, m	Set of modes of activity j in project p and mode index, $m = 1, \dots, M_{pj} $ for project p and activity j
d_{pjm}	Duration of activity j of project p in mode m
R, r	Set of renewable resources and renewable resource index, $r = 1, \dots, R $
N, n	Set of non-renewable resources and non-renewable resource index, $n = 1, \dots, N $
\hat{r}_{pjmr}	Amount of renewable resource r required by activity j of project p in mode m
\hat{n}_{pjmn}	Amount of non-renewable resource n required by activity j of project p in mode m
\bar{r}_t	Capacity of renewable resource r in period t
\bar{n}_n	Capacity of non-renewable resource n
E_{pj}, L_{pj}	Earliest and latest completion times of activity j in project p
C	Set of all pairs of activities with immediate predecessor relationship; for example, $(i, j) \in C$ means that activity i precedes activity j
x	Set of decision variables
V	Number of objective functions
$f_k(x)$	k th objective function, $k = 1, \dots, V$
$f(x)$	Vector of objective functions

MF Opt $f(x) = [f_1(x), f_2(x), \dots, f_V(x)]$, (1)

subject to

$$x_{pjmt} = \begin{cases} 1, & \text{if activity } j \text{ of project } p \text{ in mode } m \text{ ends} \\ & \text{in period } t \\ 0, & \text{otherwise} \end{cases}$$

$$\sum_{m=1}^{|M_{pj}|} \sum_{t=E_{pj}}^{L_{pj}} x_{pjmt} = 1, \forall p \in P, \forall j \in J_p, \quad (2)$$

$$\forall p \in P, \forall j \in J_p, \forall m \in M_{pj}, \forall t \in [1, H]. \quad (7)$$

$$-\sum_{m=1}^{|M_{pi}|} \sum_{t=E_{pi}}^{L_{pi}} tx_{pimt} + \sum_{m=1}^{|M_{pj}|} \sum_{t=E_{pj}}^{L_{pj}} (t - d_{pjm})x_{pjmt} \geq 0,$$

$$\forall p \in P, \forall i, j \in J_p, \forall (i, j) \in C, \quad (3)$$

$$\sum_{p=1}^{|P|} \sum_{j=1}^{|J_p|} \sum_{m=1}^{|M_{pj}|} \sum_{q=t}^{t+d_{pjm}-1} \hat{r}_{pjmr} x_{pjmq} \leq \bar{r}_{rt}, \forall r \in R, \forall t \in [1, H], \quad (4)$$

$$\sum_{p=1}^{|P|} \sum_{j=1}^{|J_p|} \sum_{m=1}^{|M_{pj}|} \sum_{t=E_{pj}}^{L_{pj}} \hat{n}_{pjmn} x_{pjmt} \leq \bar{n}_n, \forall n \in N, \quad (5)$$

$$E_{pj} \leq tx_{pjmt} \leq L_{pj}, \forall p \in P, \forall j \in J_p, \forall m \in M_{pj}, \forall t \in [1, H], \quad (6)$$

Equation (1) demonstrates the vector optimization problem for V conflicting objectives. Equation (2) represents the assignment constraints that require each activity to be completed exactly once. Precedence relations among the activities are maintained by inequality (3). Renewable and non-renewable resource limitations are enforced by inequalities (4) and (5), respectively. Doubly constrained resources are also covered by the formulation MF (Talbot, 1982; Węglarz et al., 2011). Constraint set (6) imposes lower (E_{pj}) and upper (L_{pj}) bounds on the completion times of the activities. Variables E_{pj} and L_{pj} can be obtained by performing forward and backward recursions on the resource unconstrained version of the problem using the mode with minimal duration. In backward recursion, the completion of the dummy sink activity is set to a known heuristic completion time, H . If such an estimate is unknown, it is set to the sum of the maximum durations of all activities. The decision variables x_{pjmt} are defined in Eq. (7).

The assumption of no precedence relation among the projects can be easily removed. Different types of precedence relations can be considered when creating a

composite network. A project may precede not just another one but also an activity or a set of activities in another project. Minimum delays may exist between two consecutive projects. If so desired, these possible extensions can be incorporated into MF without causing additional difficulty.

The assumptions made concerning the cash flows in the projects are as follows. (i) The activity costs are incurred at their completion. Dummy activities are excluded because no cost is defined for them. (ii) A lump sum payment is received at the termination of each project. (iii) Each project starts with an upfront investment, which can be interpreted as a relatively large-scale expense to make assets ready for executing the activities (set-up costs). All these financial parameters enable us to calculate the *NPV* of a given multi-project schedule by using an appropriate discount factor.

As mentioned in the Introduction, the objective functions dealt with in this work are C_{\max} , *NPV*, *MCT*, and *MFT* over a set of projects. The calculation of these objective functions for a given schedule is described below:

- Completion time of project $p = C_{\max,p} = \operatorname{argmax}_t (tx_{pj^{(p)}mt})$, where $j^{(p)}$ is the last activity of project p , and $j^{(p)} = \operatorname{argmax}_j (tx_{pjmt})$.
- Starting time of project $p = S_p = \operatorname{argmin}_t (tx_{pj^{(p)}mt} - d_{pj^{(p)}m})$, where $j^{(p)}$ is the first scheduled activity of project p , and $j^{(p)} = \operatorname{argmin}_j (tx_{pjmt} - d_{pjm} \geq 0)$.
- *NPV* of project p discounted to $S_p = NPV_p =$ lump sum payment realized at $C_{\max,p}$ discounted to $S_p -$ (the investment cost realized at $S_p +$ sum over each activity in project p of activity execution costs discounted to S_p).
- Overall C_{\max} of projects $= C_{\max} = \max_p C_{\max,p}$.
- Overall *NPV* of projects $= NPV = \sum_p (NPV_p \text{ discounted to } t = 1)$.
- Mean completion time $= MCT = \sum_p C_{\max,p} / |P|$.
- Mean flow time $= MFT = \sum_p C_{\max,p} - S_p / |P|$.

4 Solution methodology

The approach used in this study was based on the approximation of the Pareto front that aimed to provide the decision maker(s) with a set of non-dominated solutions from which to choose. A solution here was a vector of V objective functions that corresponded to conflicting objectives under consideration.

4.1 Definition

A solution a dominates another solution b if all the

objective components of a are at least as good as those of b . Moreover, at least one objective component of a is strictly better than that of b . If a is not dominated by any other solution in the set of solutions, then a is said to be non-dominated.

In this study, NSGA-II was utilized to handle the multiple objectives (Deb et al., 2002). NSGA-II was particularly preferred due to its wide popularity and superior performance in project scheduling literature and its observed effectiveness in practical engineering problems (Balouka et al., 2016; Xiao et al., 2016; Wang and Zheng, 2018). NSGA-III was recently proposed by Deb and Jain (2014) to cope with the simultaneous optimization of numerous objectives (typically, more than three). Moreover, NSGA-III was developed to increase diversity of non-dominated solutions. In the sequel of this paper, we implemented NSGA-II because we focused on the bi-objective version of a project scheduling problem. We also implemented an injection procedure to obtain a diverse set of non-dominated solutions.

The NSGA-II parameters, namely, population size, number of generations, crossover rate, and mutation rate, were determined by an extensive fine-tuning experiment. Apart from the standard GA operators, NSGA-II has a non-dominated sorting procedure and crowding distance operator as additional mechanisms. We contributed to NSGA-II by applying BFP (Özdamar and Ulusoy, 1996; Li and Willis, 1992) to the NSGA-II solutions as an improvement procedure. Ballestín and Blanco (2015) reported that BFP or its modifications were versatile techniques that could be employed for the solution of multi-objective RCPSPs. As previously pointed out, we also applied an injection procedure to increase the diversity in the NSGA-II solution set.

4.2 Individual representation

An individual was represented by a double list consisting of the precedence feasible activity list (henceforth, we will call it the feasible list) and the feasible mode list (Hartmann, 2001; Ulusoy et al., 2001). In the feasible list, the activities were placed into genes to ensure that all the predecessors of an activity appeared before it. The mode list consisted of modes assigned to activities from their mode sets. Figure 1 presents an example of feasible and mode lists for a problem with seven activities and a single renewable resource. The activities and their predecessors and assigned modes are listed in Table 2. Each activity in the feasible list is placed into its gene after all its predecessors have already been assigned to the previous genes. In the mode list, each gene shows the assigned mode selected from the mode set of the corresponding activity. If multiple activities, which can be placed, are present while placing an activity into a gene, then we select one of them with equal probabilities.

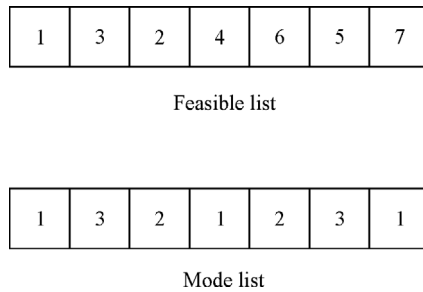


Fig. 1 Example of the individual representation.

Table 2 Example of the individual representation

Activity ID	Predecessor activity IDs	Assigned modes
1	–	1
2	1	2
3	1	3
4	2, 3	1
5	2, 3	3
6	3, 4	2
7	6, 5	1

4.3 Initial population generation

The initial population was generated by randomly creating feasible lists and their corresponding mode lists. A dummy source activity was placed into the first gene to create a feasible list. Thereafter, a set of activities, which could be placed into the current position, was created for the second gene. An activity was randomly selected by assuming equal probabilities of selection from this set and placed into the second gene. The set of eligible activities for the third gene was updated, and this procedure was repeated until all of the genes in the feasible list were filled. A mode for each activity was randomly selected from the corresponding mode set of the activity. Kolisch and Drexler (1997) verified that the associated feasibility problem for two or more non-renewable resources ($|N| \geq 2$) was NP-complete. This condition for NP-completeness was indeed also valid for the test problems we employed in this study. The strategy for handling the cases, where the mode selections are not feasible with regard to the non-renewable resource capacities, is explained in Section 4.5.

4.4 Scheduling the activities

After the feasible and mode lists were obtained, the start and completion times were assigned to the activities by using a schedule generation scheme. Demeulemeester and Herroelen (2002) stated that among the various schedule generation schemes, researchers commonly preferred the serial schedule generation scheme (SSGS) and parallel

schedule generation scheme (PSGS) to the others. Both mechanisms demonstrated the same computational complexity for the same feasible list. However, we preferred the SSGS for generating the schedules in this study because a schedule generated by this scheme belonged to the set of active schedules (Kolisch, 1996).

We applied the SSGS to the given individual representation in Section 4.2 to generate a schedule displayed in Fig. 2. The resource requirements for the assigned modes in the mode list in Table 2 are provided in Table 3. The capacity of the renewable resource was seven units. At each step of building the schedule, the algorithm selected the activity from the next gene of the feasible list and scheduled it considering the precedence relations and resource usages. Here, for instance, activity 1 was first scheduled. Once activity 1 was finished, activity 3 was scheduled, followed by activity 2. The earliest precedence feasible period that activity 2 could be assigned to was period 4. Activities 2 and 3 were simultaneously processed for one period—the duration of activity 2, and their joint resource usage was five units. This assignment of activity 2 was also resource feasible. Activity 4 was scheduled next according to the feasible list. Given that the predecessors of activity 4 were activities 2 and 3, the earliest time it could be assigned was when both activities were finished, which would be period 9. In the whole duration of four periods of activity 4, the resource requirement would be six units because no other activity had yet been simultaneously assigned during its execution periods. Hence, the scheduling of activity 4 starting at period 9 and lasting for four periods was precedence and resource feasible. The next activity to be scheduled was activity 6. Activity 6 could only be scheduled once activities 1, 3, and 4 were finished due to its precedence relations. The earliest period that activity 6 could be assigned was period 13. In the whole duration of three periods of activity 6, the resource requirement would be seven units because no other activity had yet been simultaneously assigned during its execution periods. Hence, the scheduling of activity 6 starting at period 13 and lasting for three periods was precedence and resource feasible. The schedule generation continued along the same lines, thus resulting in the schedule in Fig. 2.

4.5 Chromosome evaluation

In NSGA-II, the fitness value of an individual is given by its so-called rank value, which is defined as follows. Within the set of all individuals, the subset of non-dominated individuals constitutes a Pareto front designated to be of rank 1. If certain individuals are left after this subset is eliminated from the set of all individuals, then the process is repeated, thereby resulting in a Pareto front of rank 2. This process continues until all individuals are assigned to a Pareto front.

Some individuals might be infeasible with respect to non-renewable resource usage because we randomly

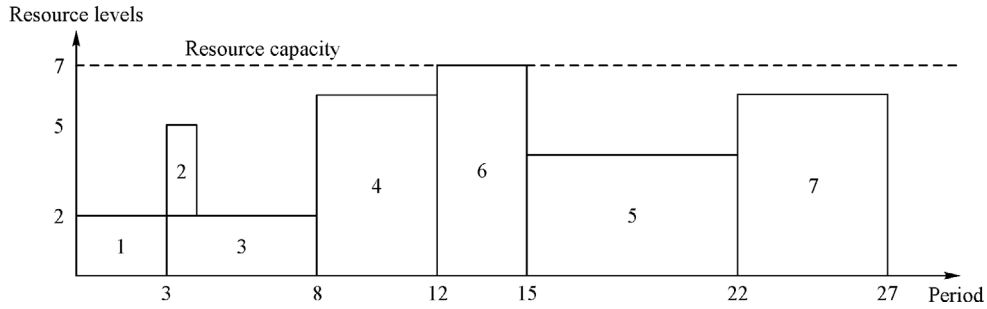


Fig. 2 Example of an SSGS application.

Table 3 Example of an SSGS application

Activity ID	Predecessor activity IDs	Duration	Assigned modes	Resource requirement
1	–	3	1	2
2	1	1	2	3
3	1	5	3	2
4	2, 3	4	1	6
5	2, 3	7	3	4
6	3, 4	3	2	7
7	6, 5	5	1	6

generated the initial population as explained in Section 4.3. In such cases, we assigned a relatively high rank value to such individuals to secure their elimination in the consecutive generations of the algorithm. Test problem sets used in this work for the computational study were not restrictive in this respect, thereby resulting only in few infeasible instances. Thus, we adopted the strategy of assigning a relatively high rank rather than devising a repair mechanism, forming a new list from scratch, or a combination of the above.

We also assigned relatively high rank values to individuals that are infeasible with respect to non-renewable resources generated in the intermediate stages of the algorithm. Infeasibility can only result from non-renewable resources. The SSGS schedules the activities in a way that renewable resource usages do not exceed the specified capacity of the renewable resources.

A crowding distance operator was used particularly for binary tournament selection and population reduction to maintain diversity (Deb et al., 2002). The crowding distance of an individual measures its gap from the neighboring individuals on the same front in the objective space using the Euclidean distance. An individual with a large crowding distance is preferable.

4.6 Forming the next generation

We implemented a one-point crossover and a two-point crossover modified to accommodate their use for multiple

modes (Hartmann, 2001). The multi-component uniform order-based crossover (MCUOX) proposed by Sivrikaya-Şerifoğlu (1997) was the third crossover mechanism implemented in this study.

A mutation operator was applied to the feasible and mode lists. On the feasible list, for every position j , the activities existing in position j and $j + 1$ were swapped with a probability equal to the mutation rate if the precedence relations were satisfied. Once this process was completed, the mutation was applied to the mode list. In every position j , the mode of the activity in position j was mutated with a probability equal to the mutation rate (Hartmann, 2001). If mutation occurred, the current mode was randomly replaced by another mode, thereby indicating that the current one could also be preserved.

Parent selection was performed in this study with binary tournament selection (Deb et al., 2002) used the same selection procedure in NSGA-II, in which the rank and crowding distance values determined the winner (Goldberg, 1989). Between two individuals, the one with the lower rank was selected as the parent, with the higher crowding distance being the tiebreaker in case of a tie of ranks.

We let POP be the population size. Parent selection and generation of POP offspring were managed to ensure that a new individual list of size $2POP$ was obtained. The reduction of this individual list to the population size POP was implemented as described by Deb et al. (2002).

In this study, an external archive was kept on the side throughout the whole solution procedure to keep the recent set of non-dominated solutions. In each generation, we placed the copies of all rank 1 individuals into the archive and sorted them using the non-dominated sorting procedure. Accordingly, the dominated individuals were removed from the archive.

4.7 Fine-tuning of parameters and performance measures

The algorithm parameters, namely, population size, number of generations, crossover rate, and mutation rate, were determined by response surface optimization (Myers et al., 2016). Multiple output variables were optimized on the basis of multiple input variables. In our case, the input

variables were the algorithm parameters, and the output variables were its performance measures. Najafi et al. (2009) provided an alternative application of the response surface methodology. In the published literature, several performance measures were proposed to evaluate a given set of non-dominated solutions. We preferred hypervolume (Zitzler and Thiele, 1998), maximum spread (Zitzler, 1999), and size of the set of non-dominated solutions because they did not require a reference set of non-dominated solutions.

Large hypervolume values indicated optimal performance for an algorithm. Origin was selected as the nadir point for a bi-objective problem with two maximization objectives. In our case, we defined the nadir point of C_{max} as the sum of the longest duration of each activity among its modes (denoted by C_{max}^r) because C_{max} was minimized.

The maximum spread evaluated the spread of the approximation set across the objective space by measuring the size of the space covered by the set. When the problem was bi-objective, this metric was reduced to the calculation of the Euclidean distance between the two farthest points in the bi-objective space. For instance, in Fig. 1, the maximum spread is equal to the Euclidean distance between the points with the minimum and maximum C_{max} values. Zitzler (1999) suggested scaling of the objective values because the objective magnitudes might be quite different.

In this study, we found the maximum spread MS of a given approximation set as follows: We let $\underline{C_{max}}$ and $\overline{C_{max}}$ be the minimum and maximum values of C_{max} in the approximation set, respectively. Correspondingly, we let \underline{NPV} and \overline{NPV} be defined in the same way. Thus:

$$MS = \sqrt{\left(\frac{C_{max} - \overline{C_{max}}}{C_{max}^r}\right)^2 + \left(\frac{NPV - \overline{NPV}}{NPV^r}\right)^2}, \quad (8)$$

where NPV^r is a large value of the corresponding objective. C_{max}^r and NPV^r should always be larger than the numerators to ensure that the maximum spread can stay between 0 and 1. NPV^r was determined as follows. We considered a multi-project instance where all lump sum payments of the projects were paid at time zero. Thereafter, we ordered all activities according to the increasing order of activity costs, thereby randomly breaking ties. The investments were assumed to occur at the end of each project. This situation represented the optimal hypothetical financial scenario (so-called ideal point) and thus would result in an upper bound on the NPV objective, namely, NPV^r . We presented the formulation only for the bi-objective case, but it could be easily generalized to other multi-objective cases.

In Fig. 3, we present a set of non-dominated solutions (gray points) obtained for the $\min C_{max}/\max NPV$ problem.

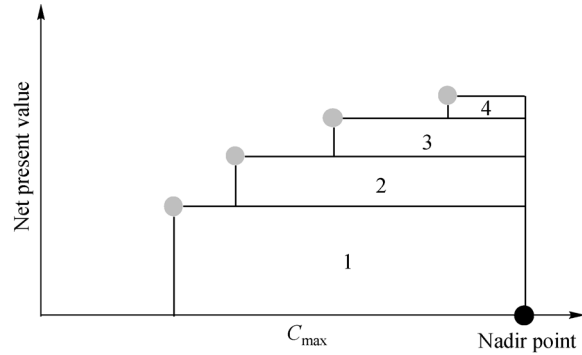


Fig. 3 Example of the hypervolume, maximum spread, and size of the set of non-dominated solutions.

Given that a nadir point was set, the hypervolume measure computed the sum of rectangular areas labeled 1, 2, 3, and 4. The maximum spread measured the Euclidean distance between two farthest points, shown as the points at the corner of rectangular areas 4 and 1 in this particular case. The size of the set of non-dominated solutions is four in Fig. 3.

4.8 Fine-tuning experiments

The 10-activity, 20-activity, and 30-activity problem sets from PSPLIB (Project Scheduling Problem Library) (Kolisch and Sprecher, 1997) were utilized to apply response surface optimization. Five instances from each of these problem sets were selected to ensure that the portfolio of the selected instances was a good representative of all instances. The experiments involved operator and parameter combinations. Operator combination refers to the combinations of binary tournament selection and crossover types (one-point, two-point, and MCUOX). Hence, we have three operator combinations. By contrast, parameter combination implies a combination of crossover rate, mutation rate, population size, and number of generations. Table 4 shows the possible values that these parameters can take on.

Table 4 Parameter ranges

Parameter	Range	Increase in increments
Crossover rate	[0.6, 1.0]	0.1
Mutation rate	[0.01, 0.25]	0.04
Population size	[20, 100]	20
Number of generations	[25, 150]	25

Most research on fine-tuning of GA parameters concluded with large crossover rates and small mutation rates, thereby resulting in relatively good solutions. Hence, we started with crossover and mutation rate ranges of 0.6 and 0.01, respectively. The corresponding increments were selected to cover sufficient search space. With regard to the

population size and number of generations, we chose the bounds on the ranges and increments (Table 4) to maintain the size of the fine-tuning experiment at a reasonable level. The details of the fine-tuning experiments were reported by Kucuksayacigil and Ulusoy (2018).

At the end of the fine-tuning experiments for the objective combination $\min C_{\max}/\max NPV$, the best population size and number of generation multiples were determined to be 1.25 and 2.5, respectively. For instance, the population size and number of generations for a 200-activity project network were set at 250 and 500, respectively.

5 Incorporating the BFP procedure into NSGA-II

The BFP procedure depends on the idea of assigning new start and completion times to the activities by applying left- and right-shifts to the scheduled activities. This procedure shifts the activities by using their slack time and includes two different pass processes. Backward pass increases the start and completion times of the scheduled activities by applying right-shifts; forward pass decreases them by applying left-shifts. A single backward pass followed by a forward pass constitutes one iteration in the BFP

procedure. The BFP procedure must not violate the resource and precedence constraints. Figure 4 presents a pseudo-code for the BFP procedure. Once the BFP procedure was applied to an individual, the decision maker was provided with a list of non-dominated solutions, which more likely improved objective function values than those of the original solutions on which BFP had been applied.

BFP was applied in two different modes. The first mode was designated here as “BFP on the archive”, where the archive referred to the set of non-dominated solutions on hand at the end of NSGA-II implementation. BFP was applied to this archive. In the second mode, BFP was not only applied at the end of NSGA-II implementation but also each time after a certain number of generations called the plateau length was generated. The second mode was designated as “BFP in the intermediate stages”. The objective combination $\min C_{\max}/\max NPV$ was utilized to solve the A, B, and C sets of the test problems with both modes of BFP (refer to Section 7 for the problem sets and their descriptions). The result of the testing and statistical analysis indicated that the BFP on the archive was superior to that in the intermediate stages (Kucuksayacigil and Ulusoy, 2018). On the basis of this result, only BFP on the archive mode was employed in the remainder of this paper.

Algorithm: BFP procedure on an individual

L : Empty list of individuals

a_0 : Individual on which BFP will be applied

S_j : Starting time of activity j of a_0

B_j : Ending time of activity j of a_0

$T > 0$: Predetermined iteration number

While ($T > 0$)

$L = L \cup \{a_0\}$

1 Sort the activities of a_0 in decreasing order of B_j

2 j^i : The i th activity in this sorted list

3 Starting from j^1 , shift all activities to the right satisfying resource and precedence relations

4 Label this intermediate individual as a_0'

If $S_{\text{Dummy start activity}} > 0$

Shift all activities of a_0' to the left so that $S_{\text{Dummy start activity}} = 0$

Label this individual as a_1 and calculate objective function values

$L = L \cup \{a_1\}$

Else if $S_{\text{Dummy start activity}} = 0$

5 Sort the activities of a_0' in increasing order of S_j

6 j^i : The i th activity in this sorted list

7 Starting from j^1 , shift all activities to the left satisfying resource and precedence relations

8 Label this individual as a_2 and calculate objective function values

9 $L = L \cup \{a_2\}$

If there are dominated individuals in $L = \{a_0, a_1, a_2\}$

Remove them from L

Else if there is no dominated individual

$T = T - 1$

Label the last individual in L as a_0

Go to step 1

End

Fig. 4 Pseudocode for the BFP procedure.

6 Incorporating the injection procedure into NSGA-II with BFP on the archive

The diversity of NSGA-II with BFP on the archive can be increased through the injection of new solutions into the population while the algorithm is in progress. In this context, a new solution was defined as a solution in which the projects were executed in some feasible order in sequence without any delay between the projects. Hence, only one project was executed in each period. The solutions differed in the ordering of the projects.

Injection was performed at every H generation. Thus, deciding on the value of H and the number of new solutions to be injected into the population was critical. The number of generations G for problem set A was 350, and the population size POP was 176. After testing with a small number of values, we decided to perform injection every 40 generations and inject 50 solutions to the population at each injection. After satisfactory results were obtained, we employed the same multipliers in proportion to problem sets B and C. Specifically, $\lceil 0.284POP \rceil$ solutions were injected to the population after every $H = \lceil 0.114G \rceil$ generation, where $\lceil \cdot \rceil$ represented rounding up to the nearest integer.

From here on, NSGA-II with BFP on the archive and injection procedure was referred to as hybrid-NSGA-II. Figure 5 presents a flowchart for the hybrid-NSGA-II.

7 Computational study

A computational study was performed employing three different multi-project test problem sets A, B, and C introduced by Can and Ulusoy (2014). The authors used the single project instances presented in PSPLIB and combined them into multi-project networks. Sets A, B, and C consisted of 81, 84, and 27 instances, respectively. A cost assignment technique was proposed by Can and Ulusoy (2014) because those instances did not have any cost and payment structure for the activities. Lump sum payments for dummy sink activities of projects and investment costs for dummy source activities of projects were defined. Different projects with individual renewable and non-renewable resource capacities were brought together to constitute a multi-project network. Hence, renewable and non-renewable resource capacities for the multi-project network were specified. The discount rate was assumed to be 0.2885% per week (15% per year) in this study. The financial parameters were set in a way that each project had a positive NPV .

A preprocessing operation was performed to eliminate non-executable modes, redundant non-renewable resources, and inefficient modes from the search space and to obtain the final version of the test problem sets (Sprecher et al., 1997). The final version of test problem sets A, B, and C is available from the corresponding author.

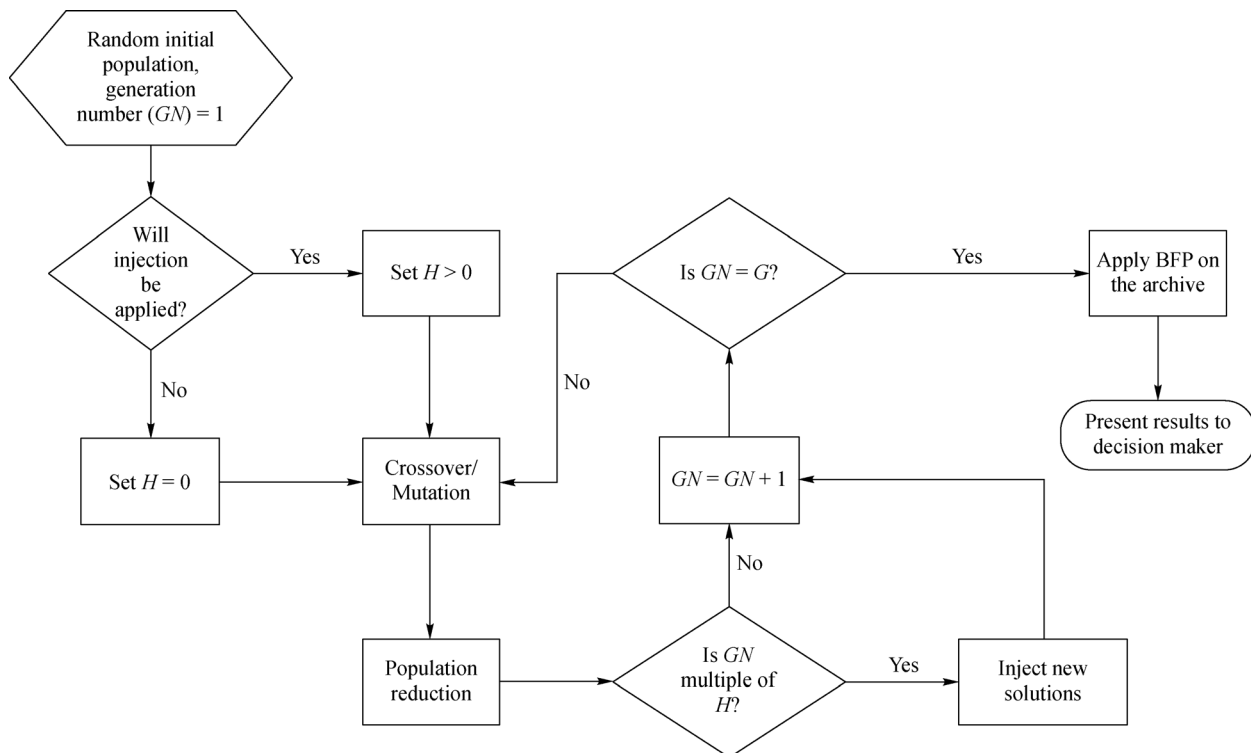


Fig. 5 Flowchart of hybrid-NSGA-II.

Table 5 Comparison of the performances of NSGA-II and NSGA-II with BFP on the archive

Test sets	Number of instances	$\overline{AC_{max}}$			\overline{ANPV}		
		NSGA-II	BFP on the archive	<i>p</i> -value	NSGA-II	BFP on the archive	<i>p</i> -value
A	81	110.97	106.96	5E-15*	281806	282119	0.03*
B	84	114.75	110.69	2E-15*	332864	333123	0.17
C	27	108.31	104.44	9E-11	385864	385461	0.36*

The algorithms were implemented in C# and run on a PC with 4 GB RAM and 3.00 GHz Intel Core 2 Quad Processor Q9650 (12M Cache, 1333 MHz FSB).

7.1 BFP on the archive for $\min C_{max}/\max NPV$

In this section, the impact of incorporating BFP on the archive into NSGA-II was tested. Table 5 summarizes the results of NSGA-II and NSGA-II with BFP on the archive implementation. The CPU times of the algorithms presented in this study were reported by Kucuksayacigil and Ulusoy (2018). A test instance resulted in multiple non-dominated solutions and multiple C_{max} values. The results in Table 5 manifest that the average of these C_{max} values ($\overline{AC_{max}}$) were first calculated for each instance. Then, the average of $\overline{AC_{max}}$ values for all the instances were computed ($\overline{AC_{max}}$). With regard to the NPV , \overline{ANPV} denoted the average of NPV values for each instance, and \overline{ANPV} represented the average of $ANPV$ s for all the instances.

The same instances were used to run NSGA-II and BFP on the archive. We used paired *t*-test with 0.05 confidence level whenever the difference between two data sets compared fit the normal distribution by using Andersen–Darling test with 0.05 confidence level. This task was carried out to conduct $\overline{AC_{max}}$ and \overline{ANPV} comparisons. Otherwise, we used Wilcoxon signed-rank test with 0.05 confidence level, which does not require normality of the difference. In Tables 5–11, the *p*-values with * indicate that the Wilcoxon signed-rank test was used to obtain the corresponding results.

In the following analyses reported in Tables 5–8, null hypothesis H_0 refers to the case of equality between the means of two data sets. On the one hand, the alternative hypothesis, H_A , meant that one data set had a smaller/larger mean than that of the other. In the $\overline{AC_{max}}$ comparison, Table 5 shows that we have sufficient

evidence to reject H_0 , thereby implying that NSGA-II with BFP on the archive outperforms NSGA-II in obtaining good $\overline{AC_{max}}$ values for all test sets. On the other hand, a significant improvement was not observed for \overline{ANPV} except for problem set A, which reflected a statistically significant improvement for \overline{ANPV} .

7.2 Effects of the injection procedure

We compared the solutions obtained by NSGA-II with BFP on the archive without implementing injection (the solutions presented in Table 5 under the “BFP on the archive” heading) and those acquired by hybrid-NSGA-II (recall that hybrid-NSGA-II was BFP on the archive with the injection procedure). This task was performed to analyze the effects of the injection procedure.

Table 6 reveals that the injection procedure was effective in improving C_{max} because we have sufficient evidence to reject the associated H_0 . This procedure is also effective in obtaining solutions with high NPV .

Figure 6 demonstrates Pareto front approximations obtained with NSGA-II and hybrid-NSGA-II algorithms in four different instances. Hybrid-NSGA-II obtained improved solutions, with a chance of presenting a large number of non-dominated solutions. The computational times for the NSGA-II algorithm and BFP on the archive and injection procedures are presented in Appendix A.

Table 7 summarizes the corresponding \overline{AMFT} and \overline{AMCT} results obtained from the schedules resulting from the ($\min C_{max}/\max NPV$) problem. The injection procedure was highly effective in reducing the mean completion and flow times of projects.

As previously stated, new solutions were injected to maintain the algorithm diversity. Hence, we report the number of non-dominated solutions obtained with and without injection in Table 8 to illustrate the aforementioned concept. The table demonstrates that the injection

Table 6 Effects of the injection procedure on $\overline{AC_{max}}$ and \overline{ANPV}

Test sets	Number of instances	$\overline{AC_{max}}$			\overline{ANPV}		
		BFP on the archive w/o injection	Hybrid-NSGA-II	<i>p</i> -value	BFP on the archive w/o injection	Hybrid-NSGA-II	<i>p</i> -value
A	81	106.96	112.99	3E-29	282119	296752	5E-15*
B	84	110.69	118.42	8E-26	333123	354743	2E-15*
C	27	104.44	110.99	7E-06*	385461	400699	1E-11

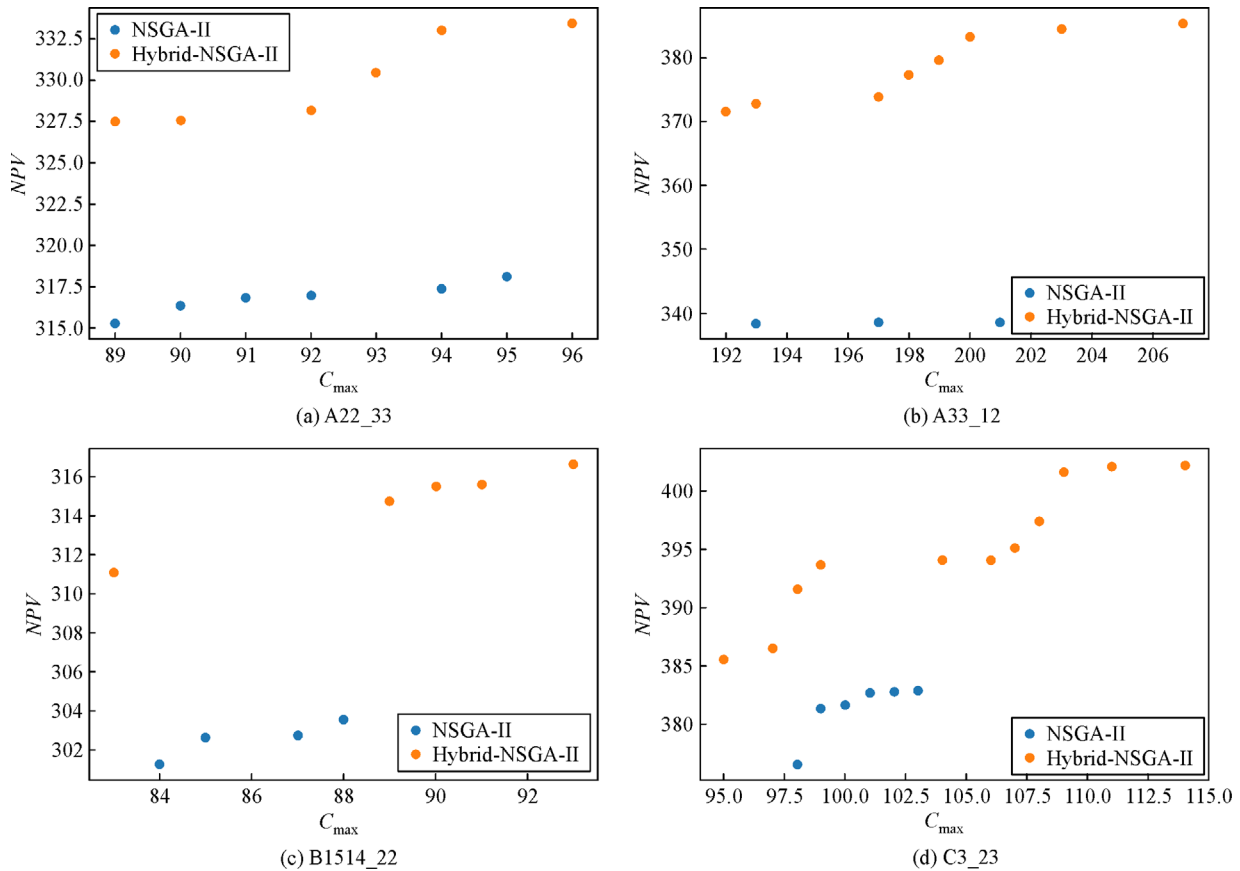


Fig. 6 Comparison of NSGA-II and hybrid-NSGA-II in four different instances.

Table 7 Effects of the injection procedure on \overline{AMFT} and \overline{AMCT}

Test sets	Number of instances	\overline{AMFT}			\overline{AMCT}		
		BFP on the archive w/o injection	Hybrid-NSGA-II	<i>p</i> -value	BFP on the archive w/o injection	Hybrid-NSGA-II	<i>p</i> -value
A	81	84.73	60.80	5E-15*	88.59	76.54	5E-15*
B	84	91.40	66.75	2E-15*	94.65	80.48	2E-15*
C	27	79.32	61.40	7E-11	82.21	73.91	8E-09

Table 8 Comparison of the average number of non-dominated solutions with and without injection

Test sets	Number of instances	Average number of non-dominated solutions		
		BFP on the archive w/o injection	Hybrid-NSGA-II	<i>p</i> -value
A	81	2.58	5.57	4E-12*
B	84	2.85	7.29	1E-21
C	27	3.37	7.37	5E-07

procedure helps find significantly non-dominated solutions because we have sufficient evidence to reject the corresponding H_0 .

We compared the algorithms on the basis of their average objective function values. The algorithms should also be compared on the basis of some performance

metrics, such as the previously defined hypervolume, maximum spread, and size of the set of non-dominated solutions. Table 9 reports such comparisons with the corresponding *p*-values. The tables illustrate that hybrid-NSGA-II provides solutions with higher hypervolumes, maximum spreads, and larger sets of non-dominated

Table 9 Hypervolume, maximum spread, and size of the non-dominated solution set metrics for NSGA-II and hybrid-NSGA-II

Test sets	Number of instances	Hypervolume			Maximum spread			Size of the solution set		
		NSGA-II	Hybrid-NSGA-II	<i>p</i> -value	NSGA-II	Hybrid-NSGA-II	<i>p</i> -value	NSGA-II	Hybrid-NSGA-II	<i>p</i> -value
A	81	0.116	0.136	1E-16	0.031	0.101	2E-22	2.802	5.568	6E-11*
B	84	0.131	0.155	3E-21	0.029	0.128	2E-32	2.762	7.286	2E-23
C	27	0.137	0.171	7E-11	0.038	0.142	5E-13	3.037	7.370	4E-08

solutions. Figures B1, B2, and B3 in Appendix B provide visual support for the statistical tests reported in Table 9. These performance metric values were calculated for the objective combination $\min C_{\max}/\max NPV$. Although we computed the hypervolume metric, the objective function values were normalized because their scales were different. By contrast, maximum spread employs by definition normalized values of the objective functions (Eq. (8)).

7.3 Solutions for $\min MFT/\max NPV$ and $\min MCT/\max NPV$

In this section, we reported the results of the three bi-objective problems obtained by processing instance sets A, B, and C using hybrid-NSGA-II. We reported the results for those two parameters — out of $\overline{AC_{\max}}$, \overline{ANPV} , \overline{AMFT} , and \overline{AMCT} — that were present in the objective combination at hand. Then, the average values for the remaining two objectives were calculated using the schedules obtained for the objective combination. For example, we calculated the \overline{AMCT} and \overline{ANPV} values for the objective combination of $\min MCT/\max NPV$. In this way, we could compare the impact of the objective combination considering the remaining two objectives. Tables 10, 11, and 12 present the relevant results for each objective combination for problem sets A, B, and C, respectively.

In all problem sets, \overline{AMCT} , \overline{AMFT} , and $\overline{AC_{\max}}$ reached their best values when the corresponding objective was part of the objective combination \overline{ANPV} . The highest

Table 10 Comparison of the results of different objective combinations from set A

Objective combinations	\overline{ANPV}	\overline{AMFT}	\overline{AMCT}	$\overline{AC_{\max}}$
$\min C_{\max}/\max NPV$	296752	60.8	76.5	113.0
$\min MFT/\max NPV$	278305	37.7	103.7	194.1
$\min MCT/\max NPV$	302253	49.3	69.0	125.1

Table 11 Comparison of the results of different objective combinations from set B

Objective combinations	\overline{ANPV}	\overline{AMFT}	\overline{AMCT}	$\overline{AC_{\max}}$
$\min C_{\max}/\max NPV$	354743	66.7	80.5	118.4
$\min MFT/\max NPV$	311640	42.7	140.0	266.0
$\min MCT/\max NPV$	363284	56.1	73.7	132.4

Table 12 Comparison of the results of different objective combinations from set C

Objective combinations	\overline{ANPV}	\overline{AMFT}	\overline{AMCT}	$\overline{AC_{\max}}$
$\min C_{\max}/\max NPV$	400699	61.4	73.9	111.0
$\min MFT/\max NPV$	356321	38.5	131.7	259.8
$\min MCT/\max NPV$	407492	50.5	64.3	128.0

values for all problem sets were reached for the objective combination $\min MCT/\max NPV$. This finding was consistent with the cash flow structure adopted here with a lump sum payment at the termination of and the positive return from each project.

One other point that attracted attention was that \overline{AMCT} had its highest value for all problem sets for the objective combination $\min MFT/\max NPV$. This result implied that in a given period, the number of projects being processed in general was relatively low, thereby allowing a higher number of resource allocations and leading to smaller flow times. This notion further implied that the projects were less densely distributed, thereby increasing the completion times. The objective combination led to the smallest \overline{ANPV} values over all the problem sets. This result was mainly due to the lump sum payment at the termination of each project. Hence, increasing the completion time values decreased the contribution of the lump sum payments to the total \overline{NPV} of the projects.

Similar to \overline{AMCT} , $\overline{AC_{\max}}$ also had its highest value for all problem sets for the objective combination $\min MFT/\max NPV$. A similar line of thought could be deduced for $\overline{AC_{\max}}$ to that given above for \overline{AMCT} .

A substantial difference existed between \overline{AMFT} values when the objective combinations were $\min C_{\max}/\max NPV$ and $\min MCT/\max NPV$. This situation was essentially a result of obtaining different start and completion times for projects by changing the modes of activities.

7.4 Different perspective of looking at the schedules

\overline{NPV} has been widely preferred to point out the financial success or failure of project schedules. However, from the perspective of the contractor, the mechanism by which frequent payments are received and the level of cash available during on-going management of projects are not

revealed. Cash availability is a crucial financial necessity for a contractor to run the business continuously (Goldratt, 1997). An objective reflecting this necessity is the minimization of the maximal cumulative gap of the contractor between cash inflow and outflow (Ning et al., 2017). The difference is designated as the *CB*. We use the convention that a positive *CB* implies that the contractor is in need of compensation for cash (such as borrowing). By contrast, a negative *CB* represents that the contractor has cash on hand. If we calculate *CB* for every period once a schedule is obtained and the cumulative sum of these values, then we obtain the *CB* diagram for the contractor. Hence, we can determine the maximum of this cumulative series. We do not consider this performance measure in this study. In this section, we demonstrate the *CB* diagrams for two sample schedules and state relevant managerial insights.

The example compares *CB* diagrams of schedules pertaining to A21_11 and A21_21 instances. A21_21 differs from A21_11 in renewable resource capacities, with the former having larger capacities (A21_21 and A21_11 have 25 and 19 units and 19 and 14 units of renewable resource capacities, respectively). All remaining properties of these two instances are retained for comparison.

Figure 7 illustrates the *CB* diagrams of the schedules of A21_11 and A21_21. Given that the latter has substantial renewable resource capacities, its projects are completed early (A21_11 and A21_21 have overall C_{\max} of 157 and 107, respectively). Furthermore, A21_11 takes longer to complete its first project to the last one compared with A21_21 because its tighter renewable resource capacities do not allow for more frequent completion of projects. The *CB* diagrams initially increase due to the execution of activities and project initiations. The diagrams increase until a lump sum payment is received (reflected as decreases). We observe a dramatic increase in the *CB* diagram for A21_21 because the number of projects

initiated in this instance is large. Both *CB* diagrams nearly have the same slope until period 84. A21_21 can schedule 109 activities, whereas A21_11 can schedule only 78 activities until period 84, which is in line with our expectation. Our detailed analysis reveals that approximately 20% of the non-dummy activities of instance A21_21 are quickly executed by selecting different modes.

We derive two important managerial insights out of this case. First, the mode change is an important strategy tool for the project contractor. The contractor is able to reach the desired outcome by changing the modes of activities. Second, we examine the reason behind the small slope of the *CB* diagram of A21_21. Accordingly, we find that a large *CB* has a negative impact on the *NPV*, which is one of the objectives used for solving these instances. Specifically, if several activities are scheduled at earlier times, then the *NPV* decreases. Hence, we can state that $\max NPV$ and minimization of maximum *CB* are non-conflicting objectives to a certain degree.

8 Discussion of computational results

The computational study has resulted in the following managerial insights for decision environments in project management where bi-objective multiple projects are simultaneously executed:

(i) The computational results clearly indicate that an attempt to reduce the mean flow time of projects by adopting $\min MFT/\max NPV$ as an objective combination has an extremely negative impact on *NPV*, completion times of the projects, and the overall C_{\max} . Hence, unless the decision maker prefers short flow times for the projects above all the other objectives for some reason, this objective combination is not recommended from the *NPV* perspective.

(ii) Higher values obtained for *NPV* are generally in line

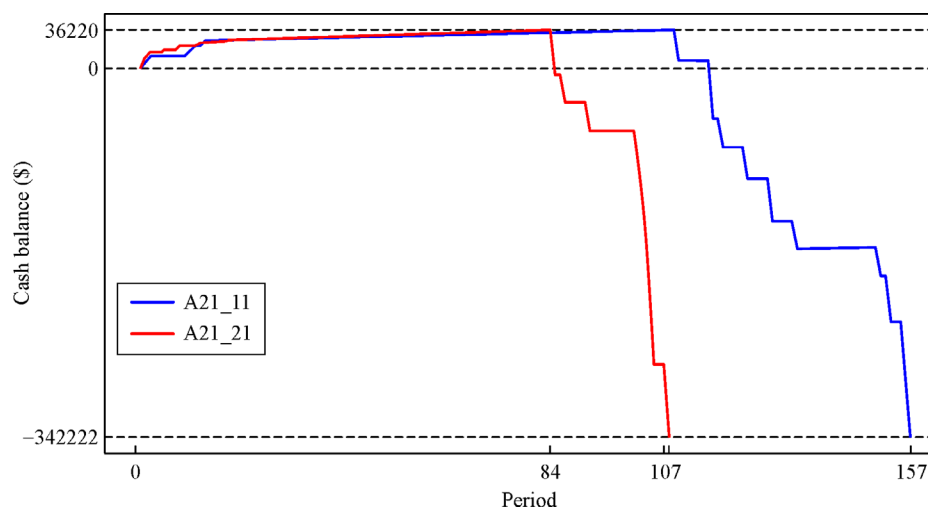


Fig. 7 *CB* diagram of instances A21_11 and A21_21.

with finishing each project as early as possible, that is, minimizing the mean completion times of the projects ($\min MCT$) rather than decreasing the overall C_{\max} ($\min C_{\max}$). This situation is due to the positive NPV resulting from the combination of the financial parameters together with the lump sum payment at the termination of each project. Competitive due dates can be quoted to the client when the project completion times are reduced.

(iii) The modification of the modes of activities, that is, increasing or decreasing the activity progress rates, is a significant strategy tool for the contractor. This step is also applied to the performance measure CB . The completion times of the projects and the overall C_{\max} considerably decrease with the increase of the capacities of renewable resources. Thus, seeking a clever policy for assigning modes to activities is definitely rewarding.

(iv) The objective combination $\min MCT/\max NPV$ still leads to acceptable values for the overall C_{\max} . This combination also leads to smaller MFT values compared with the $\min C_{\max}/\max NPV$ objective combination. The results obtained for the mean objective values by $\min C_{\max}/\max NPV$ are relatively close but inferior to those obtained by $\min MCT/\max NPV$. By contrast, the results obtained by $\min MFT/\max NPV$ are preferable. The above discussion indicates that the decision makers can limit themselves to analyzing the schedules provided by the non-dominated solutions obtained by the objective combinations $\min MCT/\max NPV$ and $\min C_{\max}/\max NPV$ given the decision-making process is restricted to NPV , C_{\max} , MFT , and MCT .

9 Conclusions

In this work, we studied bi-objective decision problems in project management, namely, $\min C_{\max}/\max NPV$, $\min MFT/\max NPV$, and $\min MCT/\max NPV$, in the case of multi-project, multi-mode RCPSP. An extensive computational study was performed using existing sets of test problem data that had been extended to include for each project an initial investment cost, activity costs, and a lump sum payment at the termination of the project. In the computational study, we also analyzed a financial performance measure referred to as CB .

We implemented hybrid-NSGA-II, which was superior to NSGA-II in terms of finding non-dominated solutions with better objective function values. Hybrid-NSGA-II could find a number of non-dominated solutions with great diversity, thus enlarging the decision space and extensively providing a versatile choice for the decision maker.

The problem investigated in this study was indeed rich, as several extensions for further research could be suggested. An interesting line of research would be to investigate different payment structures other than the lump sum payment due at the termination of each project.

Examples for such payment structures could be found among others (Ulusoy et al., 2001; Mika et al., 2005; Leyman and Vanhoucke, 2016).

The analysis of the marginal impact of increasing the budget available and its allocation to different resource capacities on the objectives under consideration is another research area of particular practical and theoretical interest. This problem is referred to as the capacity planning problem in multi-project environments (Gademann and Schutten, 2005; Beşikci et al., 2015). The solution procedure proposed here can be employed in the analysis of different scenarios for capacity planning.

Other objective functions, such as the minimization of the maximum CB and mean weighted tardiness of the projects, that have not been considered here can be applied. For example, due dates can be incorporated into the problem definition to consider tardiness, adding a new dimension of interest.

References

- Ahmad S B, Svalestuen F, Andersen B, Torp O (2016). A review of performance measurement for successful concurrent construction. *Procedia: Social and Behavioral Sciences*, 226: 447–454
- Ballestín F, Blanco R (2015). Theoretical and practical fundamentals. In: Schwindt C, Zimmermann J, eds. *Handbook on Project Management and Scheduling*, vol. 1. Cham: Springer, 411–427
- Balouka N, Cohen I, Shtub A (2016). Extending the multimode resource-constrained project scheduling problem by including value considerations. *IEEE Transactions on Engineering Management*, 63 (1): 4–15
- Beşikci U, Bilge Ü, Ulusoy G (2015). Multi-mode resource constrained multi-project scheduling and resource portfolio problem. *European Journal of Operational Research*, 240(1): 22–31
- Blazewicz J, Lenstra J K, Kan A R (1983). Scheduling subject to resource constraints: Classification and complexity. *Discrete Applied Mathematics*, 5(1): 11–24
- Brucker P, Drexl A, Möhring R, Neumann K, Pesch E (1999). Resource-constrained project scheduling: Notation, classification, models, and methods. *European Journal of Operational Research*, 112(1): 3–41
- Can A, Ulusoy G (2014). Multi-project scheduling with two-stage decomposition. *Annals of Operations Research*, 217(1): 95–116
- Chang P C, Chen S H, Lin K L (2005). Two-phase sub population genetic algorithm for parallel machine-scheduling problem. *Expert Systems with Applications*, 29(3): 705–712
- Cochran J K, Hornig S M, Fowler J W (2003). A multi-population genetic algorithm to solve multi-objective scheduling problems for parallel machines. *Computers & Operations Research*, 30(7): 1087–1102
- Deb K, Jain H (2014). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 18(4): 577–601
- Deb K, Pratap A, Agarwal S, Meyarivan T A M T (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions*

- on Evolutionary Computation, 6(2): 182–197
- Demeulemeester E L, Herroelen W S (2002). *Project Scheduling: A Research Handbook*. Boston: Kluwer Academic Publishers
- Gademann N, Schutten M (2005). Linear-programming-based heuristics for project capacity planning. *IIE Transactions*, 37(2): 153–165
- Gagnon M, Boctor F F, d'Avignon G (2005). Multicriteria project scheduling with resource availability cost. Quebec: Laval University
- Gang J, Xu J, Xu Y (2013). Multi-project resources allocation model under fuzzy random environment and its application to industrial equipment installation engineering. *Journal of Applied Mathematics*, 1–19
- Goldberg D E (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Massachusetts: Addison-Wesley Professional
- Goldratt E M (1997). *Critical Chain*. Great Barrington: North River Press
- Gu H, Schutt A, Stuckey P J, Wallace M G, Chu G (2015). Exact and heuristic methods for the resource-constrained net present value problem. In: Schwindt C, Zimmermann J, eds. *Handbook on Project Management and Scheduling*, vol. 1. Cham: Springer, 299–318
- Hartmann S (2001). Project scheduling with multiple modes: A genetic algorithm. *Annals of Operations Research*, 102(1–4): 111–135
- Hartmann S, Briskorn D (2010). A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research*, 207(1): 1–14
- Herroelen W, de Reyck B, Demeulemeester E (1998). Resource-constrained project scheduling: A survey of recent developments. *Computers & Operations Research*, 25(4): 279–302
- Herroelen W, Leus R (2001). On the merits and pitfalls of critical chain scheduling. *Journal of Operations Management*, 19(5): 559–577
- Herroelen W, Leus R (2005). Project scheduling under uncertainty: Survey and research potentials. *European Journal of Operational Research*, 165(2): 289–306
- Khalili S, Najafi A A, Niaki S T A (2013). Bi-objective resource constrained project scheduling problem with makespan and net present value criteria: Two meta-heuristic algorithms. *The International Journal of Advanced Manufacturing Technology*, 69(1–4): 617–626
- Kolisch R (1996). Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation. *European Journal of Operational Research*, 90(2): 320–333
- Kolisch R, Drexel A (1997). Local search for non-preemptive multi-mode resource-constrained project scheduling. *IIE Transactions*, 29(11): 987–999
- Kolisch R, Padman R (2001). An integrated survey of deterministic project scheduling. *Omega*, 29(3): 249–272
- Kolisch R, Sprecher A (1997). PSPLIB—A project scheduling problem library. *European Journal of Operational Research*, 96(1): 205–216
- Kucuksayacioglu F, Ulusoy G (2018). A hybrid genetic algorithm application for a bi-objective, multi-project, multi-mode, resource-constrained project scheduling problem. Working Paper #36791. Istanbul: Sabanci University
- Leyman P, Vanhoucke M (2016). Payment models and net present value optimization for resource-constrained project scheduling. *Computers & Industrial Engineering*, 91(1): 139–153
- Li K Y, Willis R J (1992). An iterative scheduling technique for resource-constrained project scheduling. *European Journal of Operational Research*, 56(3): 370–379
- Liu H, Wang Y (2009). A method for multi-project with resource constraints based on greedy strategy. In: *Proceedings of the Fifth International Conference on Autonomic and Autonomous Systems*. Valencia: IEEE, 22–27
- Mika M, Waligora G, Weglarz 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
- Myers R H, Montgomery D C, Anderson-Cook C M (2016). *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. New Jersey: John Wiley & Sons
- Najafi A A, Niaki S T A, Shahsavar M (2009). A parameter-tuned genetic algorithm for the resource investment problem with discounted cash flows and generalized precedence relations. *Computers & Operations Research*, 36(11): 2994–3001
- Ning M, He Z, Jia T, Wang N (2017). Metaheuristics for multi-mode cash flow balanced project scheduling with stochastic duration of activities. *Automation in Construction*, 81: 224–233
- Özdamar L, Ulusoy G (1996). A note on an iterative forward/backward scheduling technique with reference to a procedure by Li and Willis. *European Journal of Operational Research*, 89(2): 400–407
- Pich M T, Loch C H, Meyer A D (2002). On uncertainty, ambiguity, and complexity in project management. *Management Science*, 48(8): 1008–1023
- Riise A, Mannino C, Burke E K (2016). Modelling and solving generalised operational surgery scheduling problems. *Computers & Operations Research*, 66: 1–11
- Sacks R, Seppänen O, Priven V, Savosnick J (2017). Construction flow index: A metric of production flow quality in construction. *Construction Management and Economics*, 35(1–2): 45–63
- Shahsavar A, Najafi A A, Niaki S T A (2015). Three self-adaptive multi-objective evolutionary algorithms for a triple-objective project scheduling problem. *Computers & Industrial Engineering*, 87(1): 4–15
- Singh A (2014). Resource constrained multi-project scheduling with priority rules & analytic hierarchy process. *Procedia Engineering*, 69: 725–734
- Sivrikaya-Şerifoğlu F (1997). *A New Uniform Order-Based Crossover Operator for Genetic Algorithm Applications to Multi-component Combinatorial Optimization Problems*. Dissertation for the Doctoral Degree. Istanbul: Boğaziçi University
- Smith-Daniels D E, Aquilano N J (1987). Using a late-start resource-constrained project schedule to improve project net present value. *Decision Sciences*, 18(4): 617–630
- Speranza M G, Vercellis C (1993). Hierarchical models for multi-project planning and scheduling. *European Journal of Operational Research*, 64(2): 312–325
- Sprecher A, Hartmann S, Drexel A (1997). An exact algorithm for project scheduling with multiple modes. *Operations-Research-Spektrum*, 19(3): 195–203
- Talbot F B (1982). Resource-constrained project scheduling with time-resource tradeoffs: The non-preemptive case. *Management Science*, 28(10): 1197–1210
- Ulusoy G, Özdamar L (1995). A heuristic scheduling algorithm for improving the duration and net present value of a project. *International Journal of Operations & Production Management*, 15

(1): 89–98

Ulusoy G, Sivrikaya-Şerifoğlu F, Şahin Ş (2001). Four payment models for the multi-mode resource constrained project scheduling problem with discounted cash flows. *Annals of Operations Research*, 102(1–4): 237–261

Vanhoucke M (2009). A genetic algorithm for net present value maximization for resource constrained projects. In: Cotta C, Cowling P, eds. *Evolutionary Computation in Combinatorial Optimization*. Berlin, Heidelberg: Springer, 13–24

Wang L, Zheng X L (2018). A knowledge-guided multi-objective fruit fly optimization algorithm for the multi-skill resource constrained project scheduling problem. *Swarm and Evolutionary Computation*, 38: 54–63

Wang W X, Wang X, Ge X L, Deng L (2014). Multi-objective optimization model for multi-project scheduling on critical chain. *Advances in Engineering Software*, 68(1): 33–39

Wauters T, Kinable J, Smet P, Vancroonenburg W, Vanden Berghe G, Verstichel J (2016). The multi-mode resource-constrained multi-project scheduling problem. *Journal of Scheduling*, 19(3): 271–283

Węglarz J, Józefowska J, Mika M, Waligóra G (2011). Project scheduling with finite or infinite number of activity processing modes—A survey. *European Journal of Operational Research*, 208(3): 177–205

Xiao J, Wu Z, Hong X X, Tang J C, Tang Y (2016). Integration of electromagnetism with multi-objective evolutionary algorithms for RCPSP. *European Journal of Operational Research*, 251(1): 22–35

Xu J, Feng C (2014). Multi-mode resource-constrained multiple project scheduling problem under fuzzy random environment and its application to a large-scale hydropower construction project. *The*

Scientific World Journal, 463692

Zitzler E (1999). *Evolutionary Algorithms for Multi-objective Optimization: Methods and Applications*. Dissertation for the Doctoral Degree. Zürich: Swiss Federal Institute of Technology

Zitzler E, Thiele L (1998). Multi-objective optimization using evolutionary algorithms—A comparative case study. In: *Proceedings of the Fifth International Conference on Parallel Problem Solving from Nature*. Amsterdam: Springer, 292–301

Appendix A: CPU times

In Table A1, we list the computational times of NSGA-II and the proposed BFP on the archive and injection procedures. A hybrid-NSGA-II algorithm is obtained by combining these procedures. The CPU times in Table A1 demonstrate the results for the solution of the instances with $\min C_{\max}/\max NPV$. The BFP on the archive procedure apparently requires less than a second because it is not a repetitive process. In addition, its solution time only depends on the number of non-dominated solutions. The computational times of the injection procedure are less than 1/3 of NSGA-II over the instances considered, but it has considerable positive impact on the algorithm (Tables 6–8). The reported CPU times are obtained for a subset of test problems covering the spectrum of a different number of activities for comparison. The algorithms are implemented in C# and run on a PC with 8 GB RAM and 2.7 GHz Dual-Core Intel Core i5.

Table A1 Sample of CPU times for NSGA-II, injection, and BFP on the archive

Instances	Number of activities	CPU times (seconds)		
		NSGA-II	Injection	BFP on the archive
A11_11	140	819	175	0.13
A12_11	140	730	262	0.21
A13_11	140	721	155	0.10
A21_11	140	970	303	0.16
A22_11	140	980	279	0.09
A23_11	140	923	325	0.05
A31_11	140	1046	382	0.09
A32_11	140	1165	313	0.08
A33_11	140	1212	394	0.06
B1010_11	100	249	23	0.13
B1012_11	120	505	66	0.12
B1014_11	140	744	181	0.10
B1016_11	160	1317	276	0.22
B1018_11	180	2026	304	0.11
B1020_11	200	2508	391	0.09
B1030_11	300	11857	2670	0.34

Appendix B: Performance metrics

Figures B1, B2, and B3 illustrate the hypervolume, maximum spread, and size of the set of non-dominated solutions obtained with NSGA-II and hybrid-NSGA-II

performed on different problem sets. The horizontal axis in each figure corresponds to the test problems in sets A, B, and C, respectively. The hybrid-NSGA-II results in better solutions in terms of these performance measures over problem sets A, B, and C.

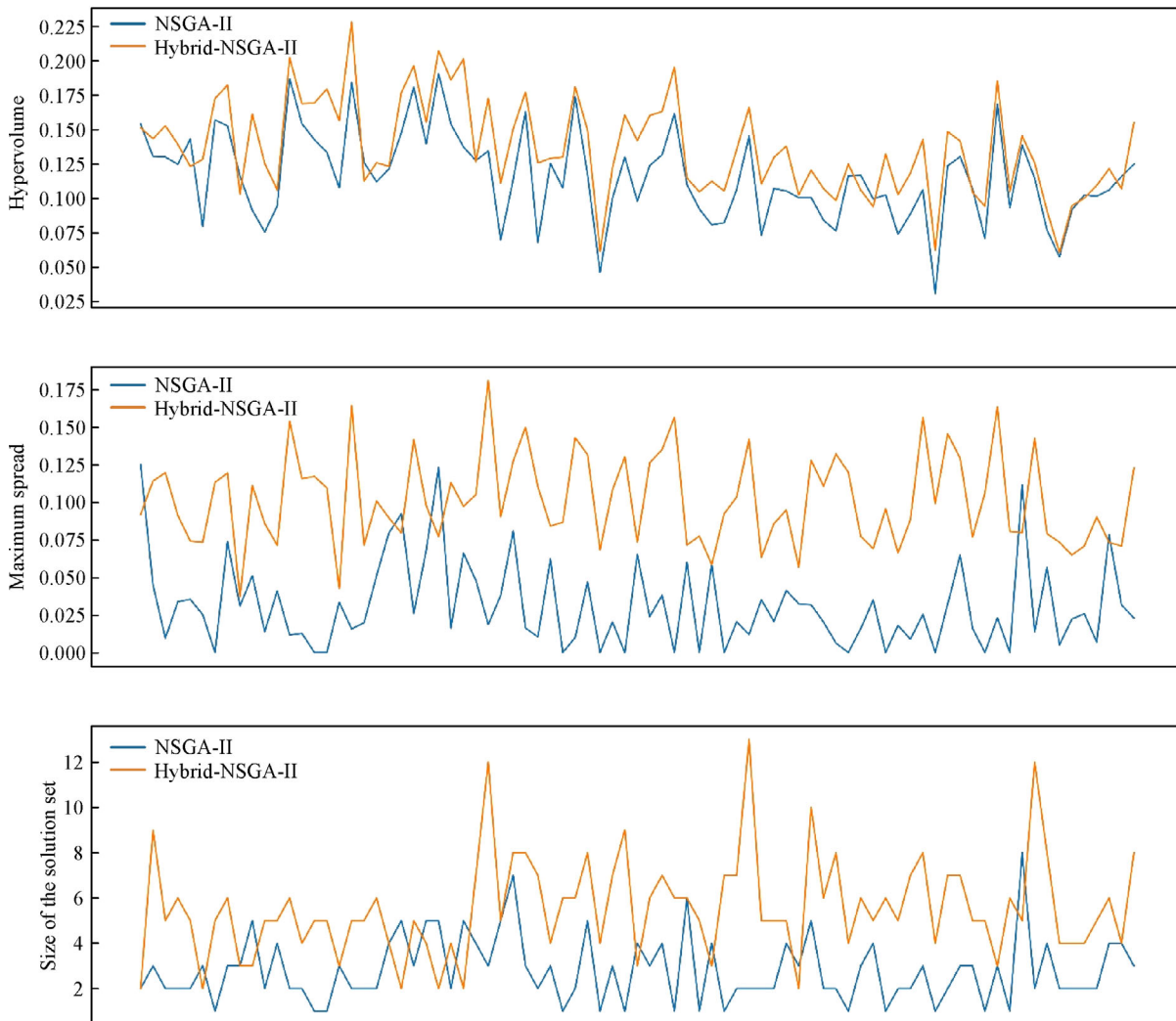


Fig. B1 Comparison of NSGA-II and hybrid-NSGA-II for problem set A.

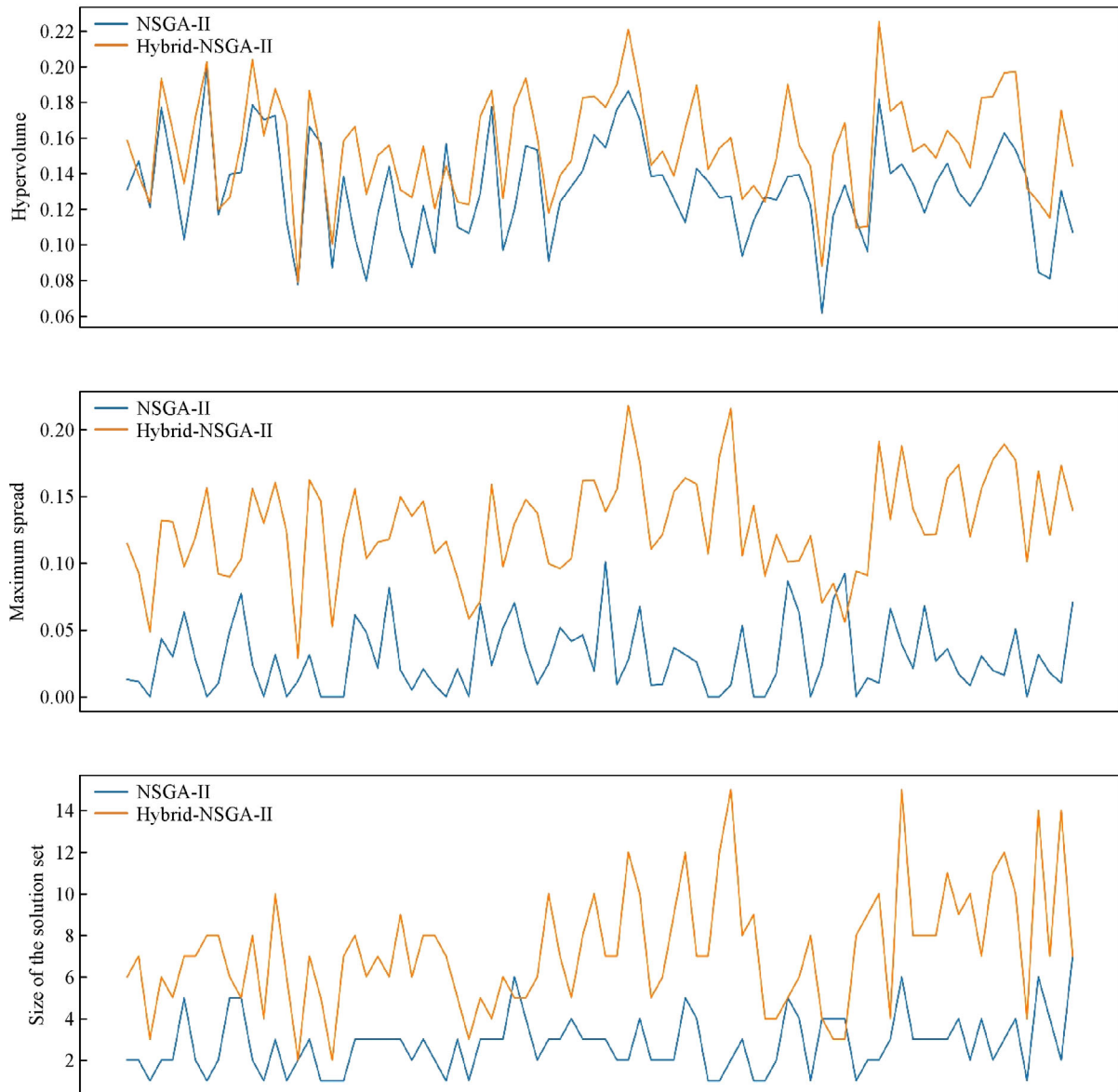


Fig. B2 Comparison of NSGA-II and hybrid-NSGA-II for problem set B.

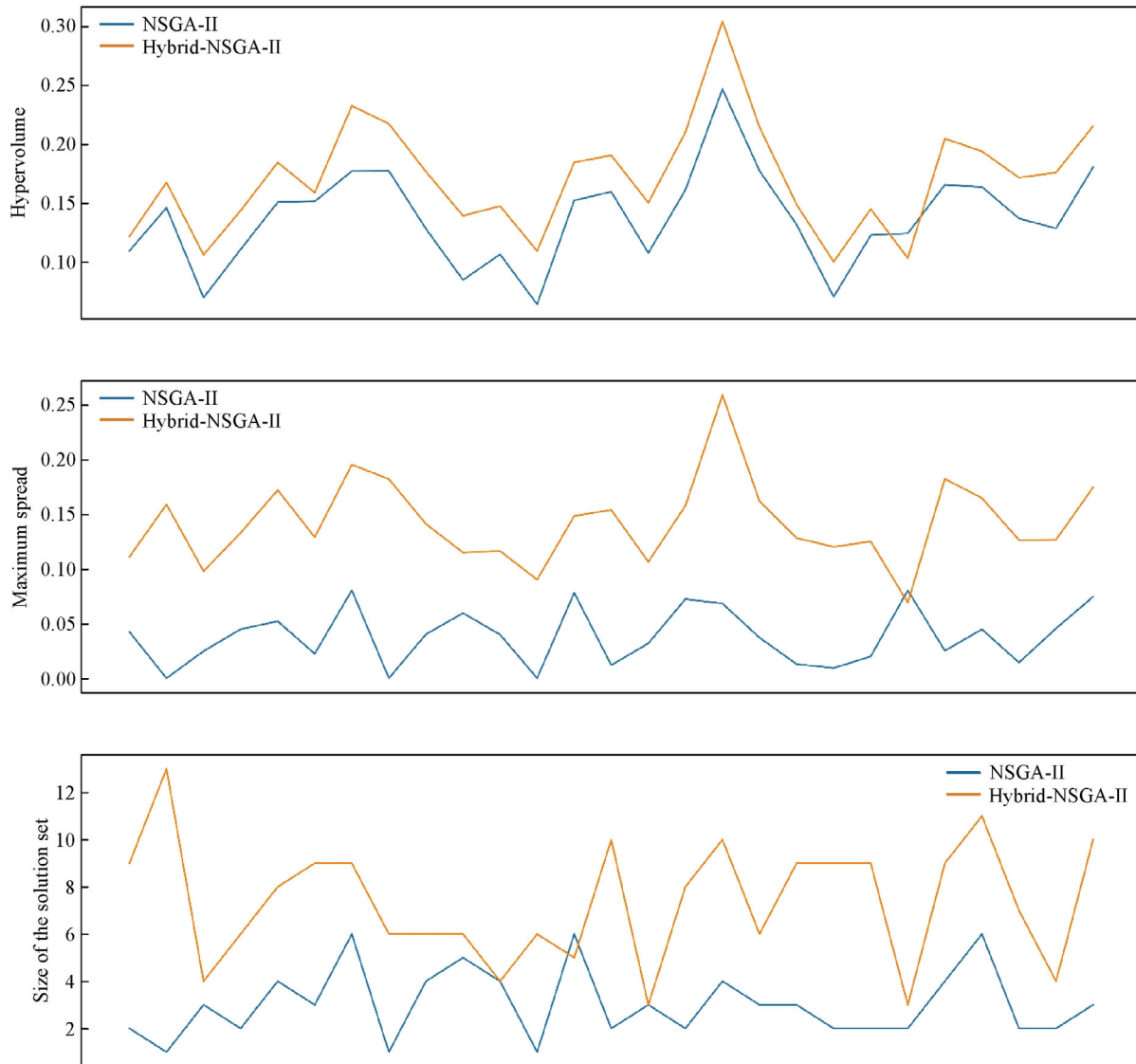


Fig. B3 Comparison of NSGA-II and hybrid-NSGA-II for problem set C.