

**A Hybrid Genetic Algorithm Application for a Bi-Objective, Multi-Project,
Multi-Mode, Resource-Constrained Project Scheduling Problem**

Fikri Kucuksayacigil

Industrial and Manufacturing Systems Engineering Department
Iowa State University
Ames, IA, USA

Gündüz Ulusoy

Industrial Engineering Department
Sabanci University
Orhanlı, Tuzla, 34956 Istanbul, Turkey

WORKING PAPER #36791

SABANCI UNIVERSITY

January 2018

Revised and Expanded
November 2018

A hybrid genetic algorithm application for a bi-objective, multi-project, multi-mode, resource-constrained project scheduling problem

Fikri Kucuksayacigil^a, Gunduz Ulusoy^b

^a Industrial and Manufacturing Systems Engineering Department, Iowa State University, Ames, IA, USA

^b Industrial Engineering Department, Sabanci University, Istanbul, Turkey

Corresponding author: Gunduz Ulusoy, gunduz@sabanciuniv.edu

Abstract

In this study, we considered a bi-objective, multi-project, multi-mode, resource-constrained project-scheduling problem. We adopted different objectives pairs, combinations of time-based and financial performance measures. As a solution method, we used the non-dominated sorting genetic algorithm II (NSGA-II). To improve NSGA-II, a backward–forward pass (BFP) procedure was proposed for new population generation as well as for post-processing. Different alternatives for implementing BFP were tested with the results reported for different objective function combinations. To increase diversity, an injection procedure was introduced and implemented. Both the BFP and injection procedures led to improved objective function values. Moreover, the injection procedure generated a significantly higher number of non-dominated solutions resulting in more diversity. An extensive computational study was performed. The results were further assessed from the perspective of maximum cash balance. Managerial insights were presented.

Keywords: Bi-objective genetic algorithm; Multi-objective multi-project multi-mode resource-constrained project scheduling problem; Backward–forward scheduling; Injection procedure; Maximum cash balance.

1. Introduction

With changing business paradigm over recent decades, currently more emphasis is put on project-based work. We observe an increase in the number of engineering, managerial and financial services companies and technology firms that structure themselves as project organizations. In line with these developments, the relevance and importance of effectively dealing with multiple simultaneous projects has increased. Finishing these projects on time by meeting the quality requirements and without exceeding the allocated budget is a major task, which provides a great challenge for the project owners as well as the project managers. Project planning and scheduling are the major tools used to meet this challenge. The core problem underlying project scheduling in project organizations is the resource-constrained project-scheduling problem (RCPSP). RCPSP is a complex problem shown to be NP-hard [1]. In recent decades, an extensive amount of work has been accomplished for developing exact and 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) [2-7]. There is a rich body of literature in multi-project scheduling under resource scarcity.

Besides task complexity, we have relational complexity in project management resulting from multiple stakeholders with conflicting interests, which can lead to disagreements about project goals and about priorities among tasks and features of the project outcome [8]. A means for handling relational complexity is employing multi-objective programming approach. In this study, we dealt with the bi-objective MRCMPSP problem. The most common and frequently used objective in project scheduling is the minimization of the makespan of projects ($\min C_{\max}$). This objective is crucial because it allows – among other things – the early release of renewable resources for subsequent projects and can help to prevent the possible violation of imposed deadlines [9]. Another significant objective in project scheduling is to maximize the net present value of projects ($\max NPV$). NPV has been preferred as a financial performance measure by many researchers and practitioners, because it is claimed to reflect the financial aspects of the decision environment more effectively [10]. In the case when only costs are involved, the objective turns out to be the minimization of NPV. Parallel processing of projects, i.e., concurrency, in construction environments has gained a greater acceptance since 1990s. Concurrent engineering, a term referring to parallel execution of tasks, has been used by practitioners to aim at minimization of lead times of projects [11]. Hence, in addition to C_{\max} and NPV, the project manager might also be interested in minimizing the mean flow time of individual projects ($\min MFT$) so that the mean throughput times of projects are reduced leading to a general reduction in work-in-progress as well [12]. The objective $\min MFT$ also reflects the contractors' increasing attention to further reduce non-value adding activities as well as waste of time and resources as the competition is fiercer in today's world [13]. Minimization of mean completion time for individual projects ($\min MCT$) can be considered as another relevant time-based objective. A decision maker may seek a project schedule that uses renewable resources more strategically leading to acceptable project completion times. Moreover, in the case a contractor carries out multiple projects, each of which pertains to different clients, meeting their individual time-based requirements would be a key factor of success for the contractor. MCT, therefore, can be closely associated with customer satisfaction and might lead to more favorable cash profiles. Since minimization of MCT and minimization of C_{\max} explicitly refer to project terminations as soon as possible, we did not consider deadlines for projects and penalties for their violations.

A problem of interest in project scheduling is the analysis of the trade-off between C_{\max} and NPV. The financial impact of reducing the duration of a project is essential information, which the decision maker uses in the project-scheduling phase. A study into the trade-off between C_{\max} and NPV for RCPSP was presented by [14]. In that formulation, a soft deadline constraint was imposed allowing a project deadline violation at a certain penalty cost. All the payments and receipts throughout the duration of an activity were discounted up to the completion time of the

activity to represent the cash flow associated with it. The objective function was the sum of the discounted cash flows of the activities and the penalty cost. Since both C_{\max} and NPV were included in the objective function, it can be considered as a multi-objective optimization model. Khalili et al. [15] considered the bi-objective problem of $\min C_{\max}$ and $\max NPV$ simultaneously for RCPSP by approximating the Pareto front. Two meta-heuristic algorithms were employed for solving the bi-objective RCPSP: multi-population genetic algorithm (GA) [16] and two-phase sub-population GA [17].

C_{\max} and NPV intuitively conflict, but they can be mutually supporting under certain conditions. Smith-Daniels and Aquilano [18] demonstrated it in a case where the resources were of a renewable type and a lump sum payment was made at the termination of the project. Activity costs were dependent on activity durations and were incurred at the start of activities. Similarly, the mutual support of these two objectives under certain circumstances was investigated by [19]. They considered two different models. In the first one, activity related cash outflows took place at activity start times and a lump sum payment occurred at the completion of the project. Activity related costs depend on the activity's total resource demand required to complete it. The second model was 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 of relevance when managing multiple projects. The reason is that 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 in itself, possibly with different owners and different project managers. Hence, it is important to have measures to follow individual projects in a multi-project environment.

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

Contribution of this study is threefold: (i) We coped with a niche area in MRCPSPP, which is its extension to include multi-project and multi-objective aspects. The literature review below will reveal that there are a few studies in this area. We proposed $\min MFT$, $\min MCT$, $\max NPV$ and $\min C_{\max}$ as objectives in this complicated problem structure. We further analyzed our results from the perspective of maximum cash balance (i.e., the maximal cumulative gap between cash inflow and outflow) for multi-project scheduling environment. (ii) We used NSGA-II in this paper, but we proposed two different improvements, one for local search and post-processing using BFP procedure to find better solutions and the other one to enlarge the set of non-dominated solutions by an injection procedure. (iii) We revealed important managerial insights concerning the preferences among objectives used for multi-project scheduling. Moreover, we elaborated on the effects of changing renewable resource capacities on schedules, i.e., increasing

or decreasing activity progress rates. Lastly, we analyzed cash balance diagrams of different schedules to find the possible interactions between renewable resource capacities and cash balance.

In the next sections, we first present the relevant studies from the literature, followed by the mathematical programming formulation of our problem. Then, we explain the adopted solution methodology and extension of that methodology with BFP and injection procedures. It is followed by an extensive computational study that discusses the results regarding the impacts of BFP and the injection procedures as well as the relationships between 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

Recently, there have been efforts to bring theory and practice closer together in project scheduling in order to deal with the real-life concerns of project practitioners. This has drawn the attention of researchers to the modeling and solution of – among others – MRCMPSP and multi-objective RCPSPs. Lately, [20] reported about the results of the nine algorithms for multi-project scheduling in the final competition of MISTA 2013 Challenge. All algorithms submitted were heuristic procedures in which some included exact components. The primary objective was to minimize the total project delay. The secondary objective was to minimize the total project duration, which was employed as a tie-breaker. To the best of our knowledge, there are a few studies, which 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, where the decision maker guides the search through the feasible solutions by choice of parameters, such as the weights of the multiple objectives involved.

All the papers reported below treat single mode problems unless otherwise stated, i.e., they are multi-objective RCMPSPs.

The paper by Liu and Wang [21] is an example of the first approach. They 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 were also assigned weights to represent the importance of these projects to the decision maker. They implemented a greedy search algorithm to find effective solutions under resource constraints. Xu and Feng [22] developed a particle swarm optimization algorithm for MRCMPSP under a fuzzy random environment. Weighted combination of overall C_{\max} and individual prioritized project C_{\max} , project cost

consisting of fixed, variable, and crashing costs of activities and quality of projects were accepted as objectives and these were combined into a single formulation with a weighted sum approach. Wang et al. [23] proposed a cloud GA to solve multi-objective RCMPSP with time, cost, quality and robustness being the objectives. The objective function was defined as the weighted sum of the utility function of each objective. A practical application of project scheduling was studied in [24], in which operational surgery scheduling problem was modeled as MRCMPSP with generalized time constraints and with application-specific additional constraints and solved with an iterative search algorithm. Having considered an arriving patient as a project, the authors took into account many performance measures represented as weighted sum such as the number of unscheduled patients, patient waiting times, child early objective (children surgery had better be scheduled in the morning), and finish early in the day (e.g., C_{\max}).

We grouped the following papers under the second approach. Kim and Schniederjans [25] proposed a heuristic method utilizing an artificial intelligence approach. The developed software allowed the user to schedule projects simultaneously. The objectives were meeting due dates for projects, maintaining a designated production level, minimizing the work-in-progress time and maximizing the workshop stability (i.e., minimizing the number of revisions to a schedule). Chen [26] developed a 0-1 goal-programming formulation with the objectives of minimizing the deviation of each project from its deadline, the total project cost and the cost of each critical project. The proposed algorithm was implemented for different maintenance projects in a copper mine in China. For MRCPSPP with minimization of C_{\max} and total tardiness as the objectives, Tasan and Gen [27] proposed a solution procedure using GA. Lova et al. [28] proposed a multi-objective heuristic method to schedule the activities in two phases. The algorithm minimized a time-related objective in the first phase (mean project delay or multi-project duration increase). In the second phase, the objective was chosen from project partitioning, in-process inventory, resource leveling, or idle resources. Lova and Tormos [29] considered mean project delay and overall C_{\max} as two objectives and employed a combination of random sampling with backward–forward heuristics. Elazouni and Abido [30] implemented the Strength Pareto Evolutionary Algorithm for finance-based project portfolios by considering the individual profits of projects as conflicting objectives to be maximized. Xu and Zhang [31] proposed a hybrid GA with fuzzy logic controller in order to solve the problem under a fuzzy environment. The overall C_{\max} of the projects and total tardiness penalties were considered as the objectives. Florez et al. [32] maximized workforce stability in a multi-project environment in addition to minimizing C_{\max} and the cost of projects. Having developed a mixed integer formulation, the authors proposed a ϵ -constraint method and implemented it for a real construction project. Gang et al. [33] solved multi-mode, 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, 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 in such a way so as to minimize the duration of the project they manage. Singh [34]

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 were considered as objectives. Can and Ulusoy [35] created a hierarchical model for the problem, as proposed earlier by [36], and regarded each project as a macro activity and solved the problem to maximize NPV. Then they implemented a post-processing scheme to minimize C_{\max} . They developed both an exact solution method and a GA for solving the problem. Shahsavar et al. [37] considered three objectives in a resource constrained multi-project problem setting. The objectives were the minimization of the overall C_{\max} of the projects, the minimization of the total cost associated with the resources, and the minimization of the variability of the resource usage. To generate non-dominated solutions, they employed three self-adaptive GAs. One hundred and eighty problems were solved by an evaluation using five performance metrics.

It appears that there has been no attempt made for the solution of multi-objective MRCMPSPs employing the interactive multi-objective approach. Gagnon et al. [38] introduced a triple objective model for RCPSP considering C_{\max} , resource availability cost and the amount of each resource type allocated as objectives. They used the tabu search to obtain non-dominated solutions. All non-dominated solutions found during the search were stored in a dominance tree and they were available to the project manager for examination.

Our exhaustive literature review points out that RCPSP has been studied with its several modifications (transfer times of resources, dynamic arrival of projects, etc.), mostly likely due to the demands by industry partners. We observed that multi-skill RCPSP gains an increasing attention from research practitioners. Moreover, stochasticity in problem parameters and reactive/proactive scheduling turns out to be another aspect drawing attentions. Decentralized scheduling and rework possibility of activities are other significant research themes encountered in the literature survey. We can infer from the literature review that solutions techniques based on Pareto front is much more pervasive than those combining objectives into a single expression. We also learnt that numerous metaheuristic methods (and hybrid forms) have been proposed and implemented. Among others, we realized that NSGA-II shows superior performance in many cases and gains appreciation from research practitioners. The review also sheds light that NSGA-II and other evolutionary algorithms have been constantly improved by deriving new modules, by integrating heuristic methods and optimization procedures. As the review above discloses, the literature addressing the multi-objective MRCMPSP is scarce. Furthermore, the bi-objective pairs (minMFT/maxNPV) and (minMCT/maxNPV) have not been investigated before even in a single project decision environment. One of the aims of this paper is meant to fill that gap in the literature for these types of decision problems.

3. Mathematical formulation of the problem

As stated before, we focused on the bi-objective MRCMPSP in this paper. MRCMPSP is a combinatorial optimization problem, which can be described as follows: There exist $|P|$ projects, each of which have $|J_p|$ activities, excluding dummy source and sink activities. Each activity of a project has $|M_{pj}|$ execution modes, which have their own durations, d_{pjm} (activities are non-preemptive). Moreover, each activity utilizes $|R|$ different renewable resources and $|N|$ different non-renewable resources, with utilization levels of \hat{r}_{pjmr} and \hat{n}_{pjmn} , respectively. Activities of a project have precedence relations (we assume finish-to-start precedence relations with zero time lags), but we do not assume precedence relations between projects. Renewable and non-renewable resources have capacities, which should not be exceeded for a schedule to be feasible. The problem is to determine a schedule, activity start and completion times as well as their execution modes, such that precedence constraints are satisfied, resource capacities are not exceeded, and objectives of the problem are optimized simultaneously.

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}_{rt}	Capacity of renewable resource r in period t
\bar{n}_n	Capacity of non-renewable resource n
E_{pj}, L_{pj}	The earliest and latest completion times of activity j in project p
C	The set of all pairs of immediate predecessor activities, e.g., $(i, j) \in C$ 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
ρ	Discount rate

The precedence relations among activities were demonstrated with a directed acyclic activity-on-node graph. The multiple projects were represented as a composite project network in general with dummy source and sink nodes. The notation for the mathematical formulation is given in Table 1. The mathematical formulation for the problem denoted by MF was presented in Eqs. (1) to (6). This formulation is an extension of the single objective formulation given by [39].

MF

$$\text{Opt } \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_V(\mathbf{x})] \quad (1)$$

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

$$- \sum_{m=1}^{|M_{pi}|} \sum_{t=E_{pi}}^{L_{pi}} t x_{pimt} + \sum_{m=1}^{|M_{pj}|} \sum_{t=E_{pj}}^{L_{pj}} (t - d_{pjm}) x_{pjmt} \geq 0, \quad \forall (i, j) \in C \quad (3)$$

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

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

$$x_{pjmt} = \begin{cases} 1, & \text{if activity } j \text{ of project } p \text{ in mode } m \text{ ends in period } t \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

Note that E_{pj} and L_{pj} are obtained by performing forward and backward recursion on the resource unconstrained version of the problem using the mode with the smallest duration. For backward recursion, completion of the dummy sink activity is set to a known heuristic completion time, H . If such an estimate is not known, it is set to the sum of the longest durations of all activities.

The vector optimization problem for V conflicting objectives is given in Eq. (1). Eq. (2) represents the assignment constraints, which require that each activity be completed exactly once. Precedence relationships between the activities are maintained by inequality (3). Renewable and non-renewable resource limitations are enforced by inequalities (4) and (5), respectively. The case of doubly constrained resources is covered by this formulation as well [39, 40]. The decision variables x_{pjmt} are defined in Eq. (6).

Recall that there was no precedence relationship between the projects. But different types of precedence relationships can be taken into account when building the composite network. A project might precede not just another one but precede an activity or a set of activities in another project. Furthermore, there might be minimum delays between two consecutive projects. If so

desired, these possible extensions can be incorporated into MF without causing additional difficulty.

In this study, we assumed that activity costs are incurred at their completion (excluding dummy activities because there is no cost defined for them). Moreover, a lump sum payment is received at the termination of each project. Finally, 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 enabled us to calculate the NPV of a given multi-project schedule by using an appropriate discount factor.

4. Solution methodology

The approach we took here 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 is a vector of V objective functions each corresponding to a conflicting objective 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 and 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 [41]. 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 [42-44]. Note that NSGA-III was recently proposed by Deb and Jain [45] to cope with simultaneous optimization of numerous objectives; typically, more than three. Another aim to develop NSGA-III was 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 project scheduling problem. We also implemented an injection procedure to get a more diverse set of non-dominated solutions.

The parameters of NSGA-II (population size, number of generations, crossover rate and mutation rate) were determined by an extensive fine-tuning experiment. In addition to 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 ([2], [46]) to the solutions of NSGA-II as an improvement procedure. As stated by Ballestin and Blanco [47], BFP or its modifications are versatile techniques that can be employed for the solution of multi-objective RCPSPs. As pointed out above, we also applied an injection procedure to increase the diversity in the solution set of NSGA-II.

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 mode list [48, 49]. In the feasible list, activities were replaced into genes in a way that all the predecessors of an activity appeared before it. By doing so, precedence relationships between the activities were satisfied. The mode list consisted of modes assigned to activities from their mode sets.

4.3. Initial population generation

The initial population was generated by randomly creating feasible lists and corresponding mode lists. To create a feasible list, a dummy source activity was placed into the first gene. Then, for the second gene, an eligible activity set (the set of activities which are eligible to be placed into the current position) was created. An activity was randomly selected by assuming equal probabilities of selection from this set and was placed into the second gene. For the third gene, the eligible activity set was updated and this procedure was repeated until all of the genes in the feasible list were filled. As for the mode list, a mode for each activity was selected randomly from the corresponding mode set of the activity by assuming an equal chance of selection among the modes.

4.4. Scheduling the activities

Having obtained a feasible list, start and completion times were assigned to the activities by using scheduling schemes. Demeulemeester and Herroelen [9] stated that among the various scheduling schemes (the serial scheduling scheme (SSS), the parallel scheduling scheme (PSS), backward planning, and bi-directional planning) researchers had commonly preferred the first two and that both SSS and PSS demonstrated the same computational complexity for the same feasible list. However, since a schedule generated by SSS belongs to the set of active schedules [50], we preferred SSS to generate the schedules in this study.

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 there are further individuals left after eliminating this subset from the set of all individuals, the process is repeated, resulting in a Pareto front of rank-2. This process continues until all individuals are assigned to a Pareto front. The complete algorithm can be found in [41]. Since we generated the initial population randomly, some individuals may be infeasible with respect to non-renewable resource usage. In this case, we assigned a large rank value to those individuals to eliminate them in the consecutive generations of the algorithm.

For maintaining diversity, a crowding distance operator was employed in NSGA-II, particularly for binary tournament selection and population reduction [41]. The crowding distance of an individual measures how far it is from neighboring individuals on the same front in the objective space. When calculating the distance of an individual over the objective function values, a Euclidean distance definition was used. An individual with larger crowding distance is more preferable.

4.6. Forming the next generation

Three different crossover operators proposed in the literature were implemented in this study. One-point and two-point crossover procedures were defined by [51] for the single mode case. Hartmann [49] expanded one-point crossover for use in the case of multiple modes. We implemented one-point crossover as defined by [49] and a two-point crossover modified to accommodate its use for multiple modes. The other crossover mechanism implemented in this study was the multi-component uniform order-based crossover (MCUOX) proposed by [52].

A mutation operator was applied to both the feasible list and mode list. 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 relationships were satisfied. Once this process was completed, the mutation was applied to the mode list. For every position j , the mode of the activity in position j was mutated with a probability equal to the mutation rate [49]. If mutation happened, the current mode was randomly replaced by another mode, which implied that the current mode could also be preserved.

Parent selection was performed in this study with binary tournament selection (Deb et al. [41] used the same selection procedure in NSGA-II), in which rank and crowding distance values determined the winner [53]. The individual with a better rank between two individuals was selected as the parent. If there is a tie in rank values, the individual with higher crowding distance was selected.

While selecting the parents, the number of offspring to be produced depends on the type of crossover mechanism used. One-point and two-point crossovers produce two offspring from a parent. On the other hand, MCUOX creates one offspring from a parent. It is critical because we needed to produce POP offspring so that the new individual list could have a $2POP$ size, where POP denotes the size of the population.

Once we had $2POP$ individuals consisting of existing and newly created offspring, population reduction was implemented as described in [41]. The individuals were grouped according to their ranks. Then, starting with the group with rank 1, the groups were included in the next population until the size of the next population equaled POP . Note that through this procedure, the elite preservation property of NSGA-II was achieved. In the case that the last group cannot be accommodated in full into the POP , some individuals were eliminated so that the population size was reduced to POP . For this purpose, the individuals in the corresponding group were sorted in

decreasing order of their crowding distance values. Starting from the top of the list, the individuals were included in the next population until its size reached POP . This procedure was meant to enhance the diversity of the population.

In this study, an external archive was kept on the side throughout the whole solution procedure in order to keep the most recent set of non-dominated solutions. In each generation, we placed the copies of rank 1 individuals into the archive and sorted the individuals in the archive employing the non-dominated sorting procedure, thereby removing the dominated individuals from the archive.

4.7. Fine-tuning of the parameters and performance measures

The parameters of the algorithm (population size, number of generations, crossover rate and mutation rate) were determined by response surface optimization [54], in which multiple output variables were optimized based on multiple input variables. In our case, input variables were parameters of the algorithm and output variables were its performance measures. (For an alternative application of response surface methodology, see [55]). In published literature, several performance measures have been proposed to evaluate a given set of non-dominated solutions. We preferred hypervolume [56], maximum spread [57] and the size of the set of non-dominated solutions, because they do not require a reference set of non-dominated solutions. In the following sections, hypervolume and maximum spread measures will be explained in detail. Concerning the size of the non-dominated solutions measure, it is clear that the larger the set, the more preferable it is.

4.7.1. Hypervolume

Hypervolume measures the total area of rectangular shapes in the objective space, which are composed of the solutions in the approximation set and a reference point. For instance, a non-dominated solution a with two objective function values $f_1(\cdot)$ and $f_2(\cdot)$ form a rectangle defined by the points $(f_1(a), f_2(a))$ and $(0,0)$. The union of all rectangles formed by all non-dominated solutions of a Pareto front is defined as the hypervolume for the Pareto front. For the case when both objectives are minimized or maximized, $(0,0)$ can be selected as the reference point, and the smaller (larger) hypervolume represents the better situation for minimization (maximization). If the objectives improve in opposite directions, Zitzler and Thiele [56] suggested that bounds or optimum values for each objective could be taken separately to form a reference point. It was reported by the authors that hypervolume does not need scaling of the objective values.

As indicated in [56], it is better not to stick to a single performance measure. Instead, one should take advantage of several performance measures simultaneously. For instance, in addition to hypervolume, Zitzler and Thiele [56] compared two approximation sets by investigating how many solutions in the second one are dominated by the first one and vice versa.

For $\min C_{\max}/\max NPV$, suppose the approximation set appears as in Fig. 1. The circles placed on

the bottom right-hand corners of the rectangles labeled 1, 2, 3 and 4 are non-dominated solutions in the approximation set, and the circle placed on the upper left-hand corner of rectangle with label 1 is the reference point. The summation of four rectangular areas is accepted as the hypervolume measure. Since NPV is maximized and C_{\max} is minimized, a smaller hypervolume value is better. When there is only one solution in the approximation set, the area of the single rectangular shape created by the solution and the reference point corresponds to the hypervolume measure.

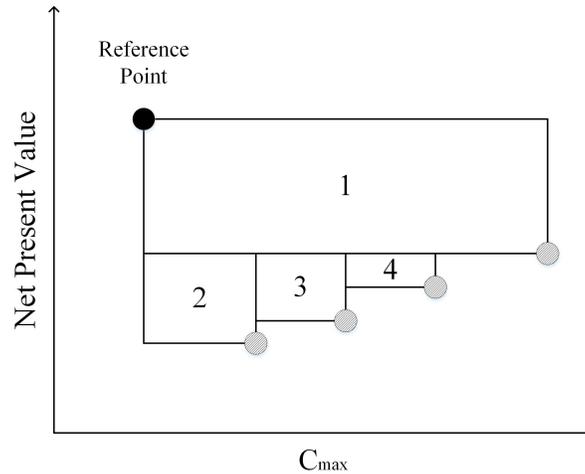


Fig. 1. Hypervolume measure for $\min C_{\max}/\max \text{NPV}$.

The crucial task was to determine the reference point. In our case, the reference value for C_{\max} was simply set to be the earliest completion time for all projects by disregarding the resource requirements. For NPV, setting a reference value is complicated because each project has an initial investment cost, a lump sum payment and execution costs for its activities. It was difficult to set a reference value quickly for NPV (in this case, we sought a value as large as possible – the so-called upper bound). We did not use an optimization model for NPV maximization because of the large size of the model. Instead, we provided a bound for NPV as follows: We considered a multi-project instance where all lump sum payments of the projects were paid at time zero. Then, we ordered all activities in increasing order of activity costs, breaking ties randomly. As for the investments, they were incurred at the end of each project. This situation represented the best hypothetical financial scenario for a project practitioner. Thus, it could be viewed as an upper bound on the NPV objective.

4.7.2. Maximum spread

Maximum spread evaluates how far the approximation set spreads across the objective space by measuring the size of the space covered by the approximation set. When the problem is bi-objective, this metric reduces 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

[57] suggested scaling of the objective values, since the magnitudes of the objectives might be quite different.

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

$$MS = \sqrt{\left(\frac{C_{max} - \overline{C_{max}}}{C_{max}^r}\right)^2 + \left(\frac{NPV - \overline{NPV}}{NPV^r}\right)^2} \quad (7)$$

where C_{max}^r and NPV^r are two large values of the corresponding objectives. They should always be larger than the numerators so that the maximum spread can stay between 0 and 1. Whereas C_{max}^r can be set as the horizon of the multi-project instance (calculated as the sum of the longest duration of each activity), NPV^r can be determined by the procedure explained in Section 4.7.1. Note that we present the formulation only for the bi-objective case, but it can be easily generalized to other multi-objective cases.

4.7.3. Fine-tuning experiments

To apply response surface optimization, 10-activity, 20-activity and 30-activity problem sets from PSBLIB were utilized [58]. Five instances from each of these problem sets were selected such that the portfolio of selected instances was a good representative of all instances. The experiments involved operator combinations and parameter combinations. By operator combination, we mean combinations of binary tournament selection and crossover types (one-point, two-point and MCUOX). Hence, we have three operator combinations. On the other hand, parameter combination implies a combination of crossover rate, mutation rate, population size and number of generations. The possible values these parameters can take are given in Table 2.

Table 2

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

Since most research, which conduct fine-tuning of GA concludes with large crossover rates and small mutation rates that result in relatively better solutions, we started with crossover and mutation rate ranges to be 0.6 and 0.01, respectively. The corresponding increments were

selected to cover sufficient search space. As for the population size and number of generations, we chose the bounds on the ranges and increments as shown in Table 2 to keep the size of the fine-tuning experiment at a reasonable level.

For each operator combination, we proceeded as follows: We replicated an instance five times using a selected parameter combination. For each replication, three performance measures were calculated and the average of five replications was taken. At the end, each instance had several average performance measures, each of which pertained to a parameter combination. Using average performance measures, response surface optimization calculates a desirability value, which represents the quality of the parameter combination. For each instance, the parameter combination with the highest desirability was selected. To select a unique parameter combination for each 10-activity, 20-activity and 30-activity instance sets, the parameter combination with the least difference in its parameter values from those of the other parameter combinations was selected.

In order to select the best operator combination, each instance was solved with the determined parameter combination. After evaluating the solution qualities for each type of crossover operator, a one-point crossover was determined to be the best type of crossover operator. For larger projects, the same fine-tuning experiment was repeated with some differences. No experiments were conducted for crossover mechanisms. Instead, the one-point crossover and binary tournament selection mechanism were accepted, *a priori*, for further implementation. In addition, crossover rate and mutation rate were not experimented with, instead those values determined to be the best for 30-activity instances were borrowed from the previous experiment. Finally, the population size and number of generations were accepted as multiples of the number of the activities existing in the project network. At the end of the experiment, for 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, for a 200-activity project network, the population size and number of generations were set at 250 and 500.

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. It shifts the activities by using their slack time. It includes two different shifting (or pass) processes. While backward pass increases 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.

BFP was applied in two different modes. The first mode is designated here as “*BFP on the Archive*”, where the archive refers 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 is 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 is designated here as “*BFP in the Intermediate Stages*”.

6. Incorporating the injection procedure into NSGA-II

One way to increase the diversity of NSGA-II is 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.

It is critical to determine how many new solutions are injected into the population and how frequently injection is performed. The number of generations G for the problem set A (refer to Section 7 for the problem sets and their descriptions) 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. Having obtained satisfactory results, we employed the same multipliers in proportion to the problem sets B and C; namely, injecting $\lceil 0.284POP \rceil$ solutions to the population after every $\lceil 0.114G \rceil$ generation, where $\lceil \cdot \rceil$ represents rounding up to the nearest integer.

7. Computational study

In order to evaluate the performance of NSGA-II with extensions, the algorithm was tested with different multi-project test instances generated in [35]. They used the single project instances presented in PSBLIB [58] and combined them into multi-project networks. Since those instances did not have any cost and payment structure for the activities, a cost assignment technique was proposed in [35]. Lump sum payments for dummy sink activities of projects and investment costs for dummy source activities of projects were defined. Since different projects with individual renewable and non-renewable resource capacities were brought together to constitute a multi-project network, renewable and non-renewable resource capacities for the multi-project network were specified. The discount rate was assumed to be 0.288% per week in this study. Note that financial parameters were set in a way that each project had a positive NPV.

To represent a variety of different environmental factors, Can and Ulusoy [35] created three problem sets of multi-project instances denoted by A, B and C. Problem set A is designed to analyze the effect of resource factor and resource strength for both renewable and non-renewable resources while fixing other factors. Resource factor and resource strength are employed here as defined by Kolisch [59]. Combinations of these four variable factors with three levels of each result in 81 instances. Set A includes multi-project cases with the same number of projects and the same number of activities but different resource requirements and resource availability levels. Each instance includes 10 projects each with 14 non-dummy activities. Problem set B includes projects with different sizes, where three levels are set for the number of projects and seven levels are set for the number of activities. The resource factor for both renewable and non-

renewable resources are fixed, whereas the resource strength is assigned two levels for each of the resource categories resulting in 84 problems. Problem set C is heterogeneous in terms of project sizes consisting of projects with different number of activities resulting in 27 instances. Three multi-project groups, each with 9 multi-projects, are formed and different levels of resource strengths are assigned. In the first group, equal numbers of small, medium and large projects are combined. In the second group, a few relatively larger projects are grouped together with a group of smaller sized projects. In the third group, a few relatively smaller projects are added to a group of relatively large projects. Further information concerning these problem sets can be found in [35].

Before implementing the algorithm, a preprocessing operation was performed to eliminate non-executable modes, redundant non-renewable resources and inefficient modes from the search space [60].

As stated before, BFP was applied to both modes “BFP on the Archive” and “BFP in the Intermediate Stages”. The objective combination $\min C_{\max}/\max NPV$ was utilized for solving the A, B and C sets of test instances with both modes of BFP. However, the objective combinations $\min MFT/\max NPV$ and $\min MCT/\max NPV$ were implemented only with the BFP mode that outperforms the other for the $\min C_{\max}/\max NPV$ objective combination.

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

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

Table 3 summarizes the results of NSGA-II and BFP on the Archive implementations (see the Appendix for CPU times). Note that a test instance resulted in multiple non-dominated solutions, and therefore multiple C_{\max} values. To report the results in Table 3, the average of these C_{\max} values ($\overline{AC_{\max}}$) were calculated first for each instance and then the average of $\overline{AC_{\max}}$ values for all the instances were computed ($\overline{\overline{AC_{\max}}}$). As for NPV, \overline{ANPV} denotes the average of NPV values for each instance and $\overline{\overline{ANPV}}$ denotes the average of \overline{ANPV} s for all the instances.

The same instances were used to run NSGA-II and BFP on the Archive. In order to conduct $\overline{AC_{\max}}$ and \overline{ANPV} comparisons, we used paired t -test (with 0.95 confidence level) whenever we could show that the difference between two data sets compared fits the normal distribution (by using Andersen-Darling test with 0.95 confidence level). Otherwise, we used Wilcoxon signed-rank test (with 0.95 confidence level), which does not require normality of the difference. p -values with (*) in Table 3 indicate that Wilcoxon signed-rank test was used to obtain the corresponding results (this applies as well to all p -values with (*) in the following Tables 7.5 through 7.8).

For the following analyses reported in Table 3 and Tables 7.5 through 7.8, null hypothesis H_0 refers to the case of equality between means of two data sets. Alternative hypothesis, H_A , on the

other hand, means one data set has a smaller/larger mean than that of the other. For \overline{AC}_{\max} comparison, it can be inferred from Table 3 that we have enough evidence to reject H_0 , which implies BFP on the Archive outperformed NSGA-II in obtaining better \overline{AC}_{\max} values for all test sets. On the other hand, a significant improvement was not observed for \overline{ANPV} except for the problem set A, which reflects a statistically significant improvement for \overline{ANPV} .

Table 3

Comparison of the performances of NSGA-II and BFP on the Archive

Test sets	# 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*	281,806	282,119	0.03*
B	84	114.75	110.69	2E-15*	332,864	333,123	0.17
C	27	108.31	104.44	9E-11	385,864	385,461	0.36*

7.2. BFP in the Intermediate Stages for $\min C_{\max}/\max NPV$

In order to run BFP in the Intermediate Stages, we needed to specify certain conditions under which it was implemented. One intuitive way was to take into account the plateau length, which is defined as the number of successive generations not contributing to finding better solutions than those already in the archive. In order to determine the best plateau length for test sets A, B and C, we ran the algorithm in advance to observe the behavior of the archive in this respect.

While running in advance, generation numbers in which the archive reaches plateau lengths 5, 7, 9, 11, 13, 15, 17, 19, 21, 23 and 25 were recorded. For example, A11_11 instance reached the plateau length 5 first when the generation number was 203.

The test instances for the same instance set were separated into subgroups (INSSUB). The instances in the same subgroups had similar complexity, which was adjusted in the data generation phase in [35]. For each subgroup, averages of the recorded generation numbers for the same plateau length were calculated. The first plateau length (starting from 5 and incrementing by 2), whose average recorded generation number is larger than half of the boundary for the number of generations (pre-determined number of generations before running the algorithm), was determined to be the best plateau length. Table 4 shows the best plateau lengths for the subgroup instances.

Table 4

Best plateau lengths for the subgroup instances.

INSSUB	Best plateau length	INSSUB	Best plateau length	INSSUB	Best plateau length
A11	13	B1014	9	B1530	5
A12	21	B1016	15	B2010	19
A13	21	B1018	19	B2012	23
A21	5	B1020	15	B2014	11
A22	17	B1030	5	B2016	13
A23	19	B1510	13	B2018	5
A31	5	B1512	23	B2020	7
A32	17	B1514	5	B2030	5
A33	19	B1516	15	C1	7
B1010	5	B1518	7	C2	15
B1012	5	B1520	7	C3	11

It was observed that the algorithm could not improve the solution quality after implementing BFP in the Intermediate Stages a number of times. Thus, the algorithm can be terminated and BFP implemented again on non-dominated solutions, though it might be claimed that the last implementation of BFP does not result in any benefit in terms of finding better solutions. In this study, we decided that if the algorithm implemented BFP in the Intermediate Stages five times and did not find better solutions, the algorithm was terminated.

Table 5 presents the implementation results of BFP in the Intermediate Stages. Note that \overline{AC}_{max} and \overline{ANPV} values in NSGA-II columns are the ones reported in Table 3.

Table 5

Comparison of the performances of NSGA-II and BFP in the Intermediate Stages

Test sets	# of instances	\overline{AC}_{max}			\overline{ANPV}		
		NSGA-II	BFP in the Intermediate Stages	p-value	NSGA-II	BFP in the Intermediate Stages	p-value
A	81	110.97	107.77	2E-13	281,806	281,152	0.31*
B	84	114.75	113.38	2E-03*	332,864	328,669	6E-07*
C	27	108.31	105.49	9E-04	385,864	381,855	7E-04

From the test results reported in Table 5 it can be inferred that BFP in the Intermediate Stages is superior to NSGA-II for all three test sets in \overline{AC}_{max} because we have enough evidence to reject H_0 . On the other hand, BFP in the Intermediate Stages does not outperform NSGA-II for ANPV because we don't have enough evidence to reject H_0 and accept H_A .

From the test results discussed above, it can be concluded that BFP on the Archive was superior to BFP in the Intermediate Stages because the latter did not improve the ANPV for the NSGA-II algorithm. Therefore, we continued with BFP on the Archive for the rest of the objective combinations.

7.3. Effects of changing the capacities of renewable resources

In this section, we show how changes in the capacities of renewable resources affect C_{\max} , project schedules and resource profiles. We randomly chose the A33 subgroup instance for this analysis. There are 9 instances in this subgroup, arranged with different resource strengths. In particular, A33_11, A33_12 and A33_13 instances had the same capacities of renewable resources, but non-renewable resource capacities increased from A33_11 to A33_13. Similarly, A33_21, A33_22 and A33_23 had the same capacities of renewable resources, but they were set at higher levels than those of the first group. These three instances had the same levels of non-renewable resources as A33_11, A33_12 and A33_13. The remaining three instances (A33_31, A33_32 and A33_33) were created in the same manner.

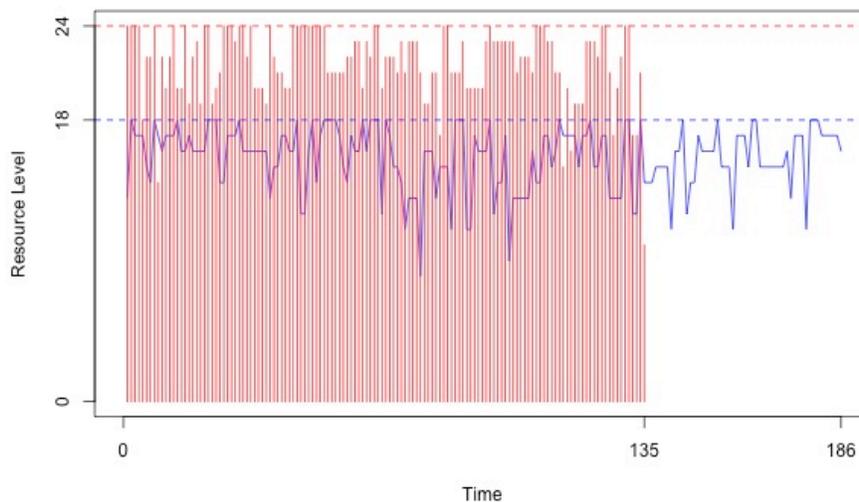


Fig. 2. Levels of renewable resource 1 for instances A33_11 and A33_21.

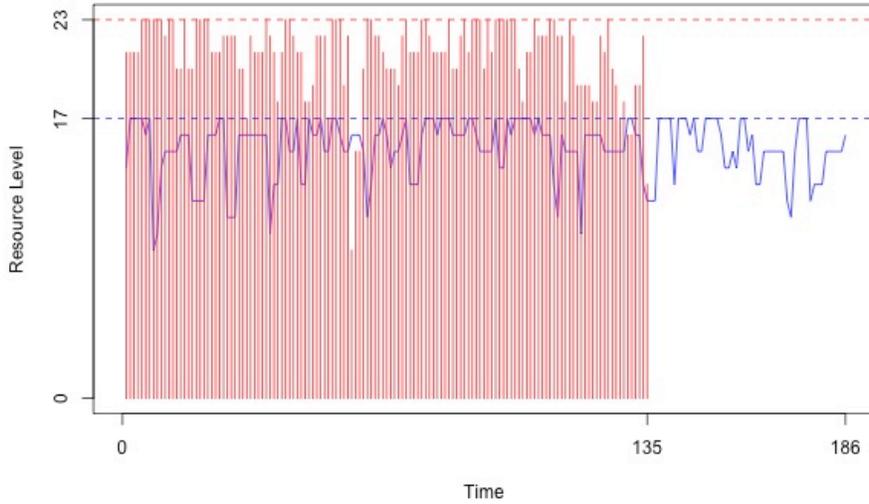


Fig. 3. Levels of renewable resource 2 for instances A33_11 and A33_21.

Fig. 2 and Fig. 3 depict the changes in the renewable resource profiles when capacities of those resources were increased simultaneously. Whereas the histogram-like resource profile (in red) is A33_13, the resource profile depicted as a line graph (in blue) pertains to A33_21. It was observed that increasing renewable resource capacities by 6 units reduced C_{max} drastically.

Fig. 4 illustrates the corresponding Gantt charts for projects in instances A33_11 (blue bars) and A33_21 (red bars). It was clearly observed that increasing renewable resource capacities reduced the completion times of all projects but one.

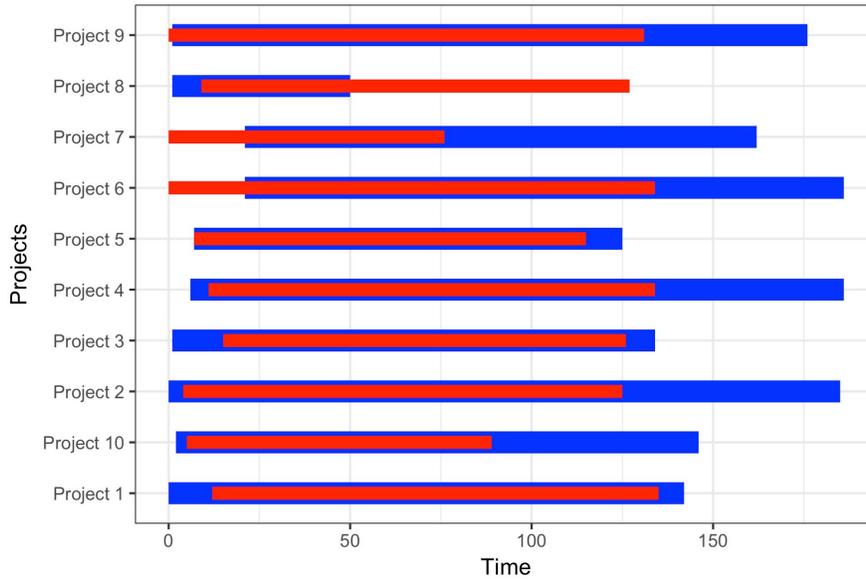


Fig. 4. Gantt charts for projects in instances A33_11 and A33_21.

Note that we did not report the effects of changes in non-renewable resource capacities on schedules. We observed that changes in their capacities did not have any impact on schedules because all schedules are feasible with respect to non-renewable resource capacities.

7.4. Effects of the injection procedure

Recall that the outcomes of Sections 7.1 and 7.2 implied superiority of BFP on the Archive over BFP in the Intermediate Stages. Therefore, we compared two sets of solutions in this section:

- (i) The solutions obtained by BFP on the Archive without implementing injection, and
- (ii) The solutions obtained by BFP on the Archive with the injection procedure.

The first solution set was in fact the set presented in Table 3 under the BFP on the Archive heading.

Table 6 reveals that the injection procedure was not effective in improving C_{\max} , since we have enough evidence to reject the associated H_0 . On the other hand, it was very effective in obtaining solutions with a higher NPV (see the Appendix for CPU times for solving the test problems using BFP on the Archive with the injection procedure).

Table 6

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

Test sets	# of instances	$\overline{AC_{\max}}$			\overline{ANPV}		
		BFP on the Archive without injection	BFP on the Archive with injection	p -value	BFP on the Archive without injection	BFP on the Archive with injection	p -value
A	81	106.96	112.99	3E-29	282,119	296,752	5E-15*
B	84	110.69	118.42	8E-26	333,123	354,743	2E-15*
C	27	104.44	110.99	7E-06*	385,461	400,699	1E-11

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

As stated before, the main idea of injecting new solutions was to maintain the diversity of the algorithm. Hence, to show that this was indeed the case, we report in Table 8 the number of non-dominated solutions obtained by both cases of without injection and with injection. It is inferred from Table 8 that the injection procedure helped to find significantly more solutions because we had enough evidence to reject H_0 .

Table 7

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

Test sets	# of instances	\overline{AMFT}			\overline{AMCT}		
		BFP on the Archive without injection	BFP on the Archive with injection	p -value	BFP on the Archive without injection	BFP on the Archive with injection	p -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 number of non-dominated solutions with and without injection.

Test sets	# of instances	Average number of non-dominated solutions		
		BFP on the Archive without injection	BFP on the Archive with injection	p -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

7.5. 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. We reported the results for those two parameters – out of \overline{AC}_{\max} , \overline{ANPV} , \overline{AMFT} , and \overline{ANPV} – that are present in the objective combination at hand. The average values for the remaining two objectives were calculated using the schedules obtained for the objective combination at hand. For example, for the objective combination of $\min MCT/\max NPV$, we calculated \overline{AMCT} and \overline{ANPV} values. This way, we could compare the impact of the objective combination under consideration of the remaining two objectives.

Table 9

Comparison of results for different objective combinations from Set A.

Objective combinations	BFP on the Archive with injections			
	\overline{ANPV}	\overline{AMFT}	\overline{AMCT}	\overline{AC}_{\max}
$\min C_{\max}/\max NPV$	296,752	60.8	76.5	112.99
$\min MFT/\max NPV$	278,305	37.7	103.7	194.1
$\min MCT/\max NPV$	302,253	49.3	69	125.1

Table 10

Comparison of results for different objective combinations from Set B.

Objective combinations	BFP on the Archive with injections			
	\overline{ANPV}	\overline{AMFT}	\overline{AMCT}	\overline{AC}_{\max}
$\min C_{\max}/\max NPV$	354,743	66.7	80.5	118.4
$\min MFT/\max NPV$	311,640	42.7	140	266
$\min MCT/\max NPV$	363,284	56.1	73.7	132.4

Table 11

Comparison of results for different objective combinations from Set C.

Objective combinations	BFP on the Archive with injections			
	\overline{ANPV}	\overline{AMFT}	\overline{AMCT}	$\overline{AC_{max}}$
minC _{max} /maxNPV	<i>400,699</i>	61.4	73.9	<i>111</i>
minMFT/maxNPV	<i>356,321</i>	38.5	131.7	259.8
minMCT/maxNPV	<i>407,492</i>	50.5	<i>64.3</i>	128

Table 9, Table 10 and Table 11 present the relevant results for each objective combination for problem sets A, B and C, respectively. The value for an objective in the objective combination investigated in that row was written in italics.

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} , on the other hand, reached its highest value for all problem sets for the objective combination minMCT/maxNPV. This was consistent with the cash flow structure adopted here with a lump sum payment at the termination of each project, as well as a positive return from each project.

One other point attracting attention was that \overline{AMCT} had its highest value for all problem sets for the objective combination minMFT/maxNPV. This result implied that in a given period the number of projects being processed in general was relatively low, allowing a higher number of resource allocations leading to smaller flow times. This further implied that the projects were distributed less densely, increasing the completion times. Interestingly, this objective combination led to the smallest ANPV values over all 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 NPV for the projects.

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

Note that there is a substantial difference between \overline{AMFT} values when objective combinations are minC_{max}/maxNPV and minMCT/maxNPV. It was essentially a result of obtaining different start and completion times for projects by changing the modes of activities. Table 12 lists an example of scheduling projects at different times for the A31_31 instance. As can be seen, project durations (flow times) with minMCT/maxNPV are much smaller than those with minC_{max}/maxNPV. Out of 140 activities in this instance, 28 activities were assigned modes with smaller durations with the minMCT/maxNPV objective combination.

Table 12

Project start and completion times for instance A31_31.

Projects	minC _{max} /maxNPV		minMCT/maxNPV	
	Start time	Completion time	Start time	Completion time
1	0	21	46	79
2	18	77	79	109
3	2	58	23	49
4	0	107	4	44
5	38	97	16	58
6	6	39	9	40
7	9	45	0	16
8	72	108	0	23
9	35	84	30	91
10	3	29	51	100

Table 13 shows another instance in which project start and completion times, as well as project durations, were different for minC_{max}/maxNPV and minMCT/maxNPV. 50 out of 270 activities had modes with smaller durations for minMCT/maxNPV when compared to modes assigned for minC_{max}/maxNPV.

Table 13

Project start and completion times for instance B1518_21.

Projects	minC _{max} /maxNPV		minMCT/maxNPV	
	Start time	Completion time	Start time	Completion time
1	28	101	2	55
2	1	73	0	31
3	3	48	0	38
4	0	48	0	78
5	0	83	0	50
6	4	115	3	80
7	0	48	0	31
8	0	111	1	80
9	0	78	0	114
10	0	114	37	105
11	0	28	5	100
12	0	36	0	29
13	0	28	51	98
14	0	83	61	116
15	0	64	0	42

Table 14 lists project start and completion times for instance C2_32. 47 out of 252 activities were assigned modes with longer durations for minMCT/maxNPV.

Table 14

Project start and completion times for instance C2_32.

Projects	minC _{max} /maxNPV		minMCT/maxNPV	
	Start time	Completion time	Start time	Completion time
1	0	25	2	57
2	0	20	20	43
3	0	48	2	24
4	0	39	0	17
5	0	63	6	42
6	0	33	0	19
7	0	50	0	12
8	0	40	0	29
9	0	23	4	56
10	2	27	7	42
11	0	14	1	31
12	1	68	6	71
13	1	58	0	24
14	0	29	0	13
15	6	65	4	36
16	1	53	0	29
17	4	49	34	89
18	0	80	24	77

7.6. A different perspective to look at the schedules

NPV has been widely preferred to point out the financial success or failure of project schedules. However, from the perspective of the contractor, it does not reveal how frequent payments are received and what the level of cash available is during on-going management of projects. For a contractor to run the business continuously without a break due to lack of cash, cash availability is a crucial financial necessity [61]. An objective reflecting this necessity is the minimization of the contractor's maximal cumulative gap between cash inflow and outflow [62]. The difference is designated as the cash balance (CB). We employ here the convention that a positive CB implies that the contractor is in need of compensation for cash –such as borrowing- whereas negative CB represents that the contractor has cash on hand. If we calculate CB for every period once a schedule is obtained, and calculate the cumulative sum of these values, we get the CB diagram for the contractor. Hence, we can determine the maximum of this cumulative series. Note that we did not take into account this performance measure in this study. In this section, we demonstrate CB diagrams for two example schedules and list relevant managerial insights.

The example compared 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 has 25 and 19 units of renewable resource capacities, respectively and A21_11 has 19 and 14 units of renewable resource capacities, respectively). All remaining properties of these two instances are kept the same for the purpose of comparison.

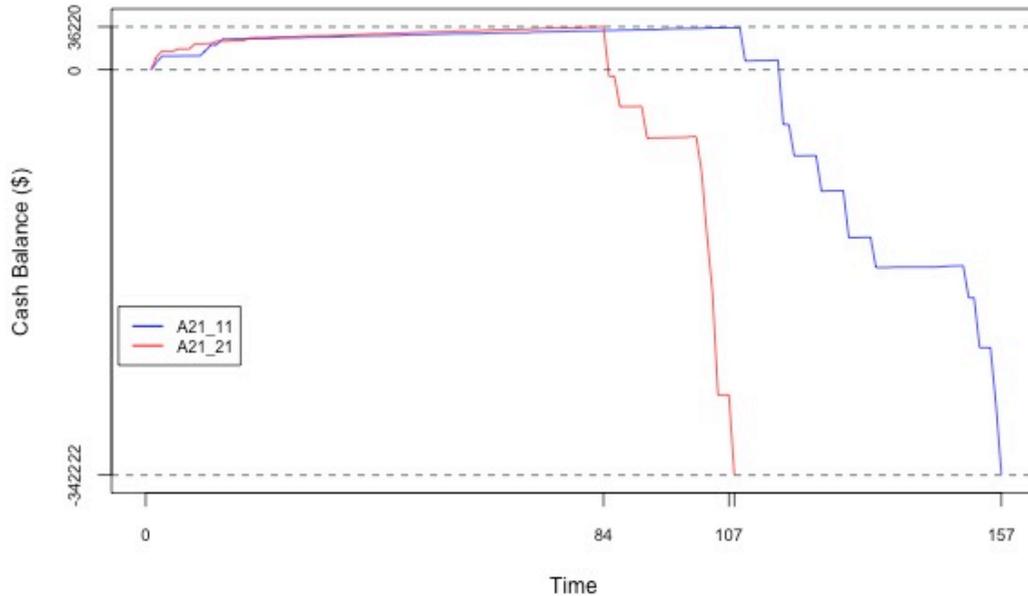


Fig. 5. CB diagram of instances A21_11 and A21_21.

Fig. 5 illustrates the CB diagrams of schedules of A21_11 and A21_21 instances. Since the latter has more renewable resource capacities, its projects were completed earlier (A21_11 and A21_21 have overall C_{max} of 157 and 107, respectively). Furthermore, it took longer for A21_11 from the completion of its first project to the last one compared to A21_21 because its tighter renewable resource capacities did not allow for more frequent completion of projects. Note that CB diagrams increased initially, resulting from the execution of activities and project initiations. CB diagrams increased up until a lump sum payment was received (reflected as decreases). We observed a more dramatic increase in CB diagram for A21_21 because number of projects initiated in this instance was larger. Notice that both CB diagrams had nearly the same slope until period 84. A21_21 scheduled 109 activities whereas A21_11 scheduled only 78 activities until period 84, which is in line with our expectation. Our detailed analysis revealed that approximately 20 percent of the non-dummy activities of instance A21_21 were executed faster by selecting different modes.

We derive two important managerial insights out of this case. First, mode change is an important strategy tool for the project contractor. By changing modes of activities, the contractor was able to reach a desired outcome. Second, by questioning why the CB diagram of A21_21 did not have a larger slope, we can conclude that larger CB has negative impact on NPV, which was one of

the objectives used for solving these instances. In other words, if more and more activities were scheduled in an earlier time, NPV would decrease. Hence, we can state maxNPV and minimization of maximum CB are to a certain degree non-conflicting objectives.

8. Conclusions

In this paper, we studied bi-objective decision problems: $\min C_{\max}/\max NPV$, $\min MFT/\max NPV$ 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 further financial performance measure referred to as cash balance (CB).

We implemented NSGA-II with a BFP extension with two modalities and an injection procedure. The results showed that BFP on the Archive modality is superior to NSGA-II in finding non-dominated solutions with better objective function values. The extension of the BFP on the Archive modality with the injection procedure resulted in solutions with higher NPV values. Moreover, it was able to find higher number of non-dominated solutions with more diversity, thus enlarging the decision space extensively providing more versatile choice for the decision-maker.

The computational study resulted in the following managerial insights:

(i) As the computational results clearly indicated, trying to reduce the mean flow time of projects by adopting $\min MFT/\max NPV$ as an objective combination had an extremely negative impact on NPV, completion times of the projects and the overall C_{\max} . Hence, unless the decision maker prefers shorter flow times for the projects above all the other objectives for some reason, this objective combination would not be recommended from the perspective of NPV.

(ii) Obtaining higher values for NPV is generally in line with finishing each project as early as possible, i.e., minimizing the mean completion times of the projects ($\min MCT$) rather than minimizing the overall C_{\max} ($\min C_{\max}$). This 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. Furthermore, more competitive due dates can be quoted to the client when completion times of the projects are reduced.

(iii) Changing the modes of activities, i.e., increasing or decreasing activity progress rates turned out to be a significant strategy tool for the contractor. This applied to the performance measure CB as well. Increasing the capacities of renewable resources decreased considerably the completion times CT of the projects and in general, the overall C_{\max} . Thus, seeking a clever policy for assigning modes to activities is definitely rewarding.

The objective combination minMCT/maxNPV still leads to acceptable values for the overall C_{\max} . It also leads to smaller MFT values compared to the min C_{\max} /maxNPV objective combination. The results obtained for the mean objective values by min C_{\max} /maxNPV are relatively close but inferior to those obtained by minMCT/maxNPV. On the other hand, they are preferable to those obtained by minMFT/maxNPV. From the above discussion, we can conclude that, given the decision-making process is restricted to NPV, C_{\max} , MFT and MCT, the decision makers can limit themselves to analyzing the schedules provided by the non-dominated solutions obtained by the objective combinations minMCT/maxNPV and min C_{\max} /maxNPV.

The problem investigated in this paper is indeed a rich one in the sense that several extensions for further research can be suggested. An interesting line of research would be to investigate different cash flow profiles other than the lump sum payment due at the termination of the project. Different payment structures can be employed for that purpose [48]. A managerial result shared above states that increasing the capacities of renewable resources considerably decreased the completion times and hence, the overall C_{\max} of the projects. Increasing capacities of resources would need an increase in the budget available. Analyzing the marginal impact of increasing the budget available and its allocation to different resource capacities on the objectives under consideration would be another research area of particular practical interest. Other objective functions not considered here, such as the minimization of maximum cash balance and minimization of mean weighted tardiness of the projects, can be employed. For considering tardiness, for example, due dates would be incorporated into the problem definition, adding a new dimension of interest.

Appendix

Table A.1

CPU times for solving the test problems using BFP on the Archive with the injection procedure

Test sets	Number of non-dummy activities	Range of CPU times (seconds)		
		Table 3	Table 5	Table 6
A	140	[435, 1101]	[267, 1106]	[467, 1164]
B	100	-	-	[171, 224]
B	120	-	-	[349, 448]
B	140	-	-	[563, 690]
B	150	-	-	[884, 1152]
B	160	-	-	[926, 1153]
B	180	-	-	[1511, 2413]
B	200	-	-	[1938, 3794]
B	210	-	-	[3032, 3962]
B	240	-	-	[5060, 7921]
B	270	-	-	[7803, 10130]
B	280	-	-	[9792, 13495]
B	300	-	-	[9255, 14286]

B	320	-	-	[16483, 22011]
B	360	-	-	[25743, 34998]
B	400	-	-	[37029, 49137]
B	450	-	-	[50582, 68042]
B	600	-	-	[115273, 158480]
C	236	[2970, 4701]	[3460, 5010]	[4528, 6868]
C	296	[6213, 9724]	[9874, 14095]	[9115, 14392]
C	300	[6540, 10434]	[2458, 4180]	[9685, 15392]

References

- [1] J. Blazewicz, J.K. Lenstra, A.R. Kan, Scheduling subject to resource constraints: Classification and complexity, *Discrete Appl. Math.* 5 (1) (1983) 11-24, [https://doi.org/10.1016/0166-218X\(83\)90012-4](https://doi.org/10.1016/0166-218X(83)90012-4).
- [2] L. Özdamar, G. Ulusoy, A note on an iterative forward/backward scheduling technique with reference to a procedure by Li and Willis, *Eur. J. Oper. Res.* 89 (2) (1996) 400-407, [https://doi.org/10.1016/0377-2217\(94\)00272-X](https://doi.org/10.1016/0377-2217(94)00272-X).
- [3] W. Herroelen, B. De Reyck, E. Demeulemeester, Resource-constrained project scheduling: A survey of recent developments, *Comput. Oper. Res.* 25 (4) (1998) 279-302, [https://doi.org/10.1016/S0305-0548\(97\)00055-5](https://doi.org/10.1016/S0305-0548(97)00055-5).
- [4] P. Brucker, A. Drexl, R. Möhring, K. Neumann, E. Pesch, Resource-constrained project scheduling: Notation, classification, models, and methods, *Eur. J. Oper. Res.* 112 (1) (1999) 3-41, [https://doi.org/10.1016/S0377-2217\(98\)00204-5](https://doi.org/10.1016/S0377-2217(98)00204-5).
- [5] R. Kolisch, R. Padman, An integrated survey of deterministic project scheduling, *Omega* 29 (3) (2001) 249-272, [https://doi.org/10.1016/S0305-0483\(00\)00046-3](https://doi.org/10.1016/S0305-0483(00)00046-3).
- [6] W. Herroelen, R. Leus, Project scheduling under uncertainty: Survey and research potentials, *Eur. J. Oper. Res.* 165 (2) (2005) 289-306, <https://doi.org/10.1016/j.ejor.2004.04.002>.
- [7] S. Hartmann, D. Briskorn, A survey of variants and extensions of the resource-constrained project scheduling problem, *Eur. J. Oper. Res.* 207 (1) (2010) 1-14, <https://doi.org/10.1016/j.ejor.2009.11.005>.
- [8] M.T. Pich, C.H. Loch, A.D. Meyer, On uncertainty, ambiguity, and complexity in project management, *Manag. Sci.* 48 (8) (2002) 1008-1023, <https://doi.org/10.1287/mnsc.48.8.1008.163>.
- [9] E.L. Demeulemeester, W.S. Herroelen, *Project Scheduling - A Research Handbook*, Kluwer Academic Publishers, Boston, 2002, 978-0-306-48142-0, <https://doi.org/10.1007/b101924>.

- [10] H. Gu, A. Schutt, P.J. Stuckey, M.G. Wallace, G. Chu, Exact and heuristic methods for the resource-constrained net present value problem, in: C. Schwindt, J. Zimmermann (Eds), Handbook on Project Management and Scheduling Vol. 1, Springer International Publishing, Switzerland, 978-3-319-05442-1, 2015, pp. 299-318, https://doi.org/10.1007/978-3-319-05443-8_14.
- [11] S.B. Ahmad, F. Svalestuen, B. Andersen, O. Torp, A review of performance measurement for successful concurrent construction, *Procedia Soc. Behav. Sci.* 226 (2016) 447-454, <https://doi.org/10.1016/j.sbspro.2016.06.210>.
- [12] W. Herroelen, R. Leus, On the merits and pitfalls of critical chain scheduling, *J. Oper. Manag.* 19 (5) (2001) 559-577, [https://doi.org/10.1016/S0272-6963\(01\)00054-7](https://doi.org/10.1016/S0272-6963(01)00054-7).
- [13] R. Sacks, O. Seppänen, V. Priven, J. Savosnick, Construction flow index: A metric of production flow quality in construction, *Constr. Manag. Econ.* 35 (1-2) (2017) 45-63, <https://doi.org/10.1080/01446193.2016.1274417>.
- [14] M. Vanhoucke, A genetic algorithm for net present value maximization for resource constrained projects, in: C. Cotta, P. Cowling (Eds.) *Evolutionary Computation in Combinatorial Optimization*, Springer-Verlag, Berlin Heidelberg, 978-3-642-01009-5, 2009, pp. 13–24, https://doi.org/10.1007/978-3-642-01009-5_2.
- [15] S. Khalili, A.A. Najafi, S.T.A. Niaki, Bi-objective resource constrained project scheduling problem with makespan and net present value criteria: Two meta-heuristic algorithms, *Int. J. Adv. Manuf. Technol.* 69 (1-4) (2013) 617-626, <https://doi.org/10.1007/s00170-013-5057-z>.
- [16] J.K. Cochran, S.M. Horng, J.W. Fowler, A multi-population genetic algorithm to solve multi-objective scheduling problems for parallel machines, *Comput. Oper. Res.* 30 (7) (2013) 1087-1102, [https://doi.org/10.1016/S0305-0548\(02\)00059-X](https://doi.org/10.1016/S0305-0548(02)00059-X).
- [17] P.C. Chang, S.H. Chen, K.L. Lin, Two-phase sub population genetic algorithm for parallel machine-scheduling problem, *Expert Syst. Appl.* 29 (3) (2005) 705-712, <https://doi.org/10.1016/j.eswa.2005.04.033>.
- [18] D.E. Smith-Daniels, N.J. Aquilano, Using a late-start resource-constrained project schedule to improve project net present value, *Decis. Sci.* 18 (4) (1987) 617-630, <https://doi.org/10.1111/j.1540-5915.1987.tb01550.x>.
- [19] G. Ulusoy, L. Özdamar, A heuristic scheduling algorithm for improving the duration and net present value of a project, *Int. J. Oper. Prod. Manag.* 15 (1) (1995) 89-98, <https://doi.org/10.1108/01443579510077241>.
- [20] T. Wauters, J. Kinable, P. Smet, W. Vancroonenburg, G. Vanden Berghe, J. Verstichel,

- The multi-mode resource-constrained multi-project scheduling problem, *J. Sched.* 19 (3) (2016) 271-283, <https://doi.org/10.1007/s10951-014-0402-0>.
- [21] H. Liu, Y. Wang, A method for multi-project with resource constraints based on greedy strategy, *Proceedings of the Fifth International Conference on Autonomic and Autonomous Systems*, Valencia, Spain, 2009, pp. 22-27, <https://doi.org/10.1109/ICAS.2009.9>.
- [22] J. Xu, C. Feng, Multi-mode resource-constrained multiple project scheduling problem under fuzzy random environment and its application to a large-scale hydropower construction project, *Sci. World J.* 2014 20p. <http://dx.doi.org/10.1155/2014/463692>.
- [23] W.X. Wang, X. Wang, X.L. Ge, L. Deng, Multi-objective optimization model for multi-project scheduling on critical chain, *Adv. Eng. Softw.* 68 (1) (2014) 33-39, <https://doi.org/10.1016/j.advengsoft.2013.11.004>.
- [24] A. Riise, C. Mannino, E.K. Burke, Modelling and solving generalised operational surgery scheduling problems, *Comput. Oper. Res.* 66 (2016) 1-11, <https://doi.org/10.1016/j.cor.2015.07.003>.
- [25] S.O. Kim, M.J. Schniederjans, Heuristic framework for the resource constrained multi-project scheduling problem, *Comput. Oper. Res.* 16 (6) (1989) 541-556, [https://doi.org/10.1016/0305-0548\(89\)90040-3](https://doi.org/10.1016/0305-0548(89)90040-3).
- [26] V.Y. Chen, A 0–1 goal programming model for scheduling multiple maintenance projects at a copper mine, *Eur. J. Oper. Res.* 76 (1) (1994) 176-191, [https://doi.org/10.1016/0377-2217\(94\)90015-9](https://doi.org/10.1016/0377-2217(94)90015-9).
- [27] S. Tasan, M. Gen, A priority-based genetic algorithm approach for solving multiple alternative project scheduling problems with resource constraints and variable activity times, *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, Singapore, 2008, pp. 2537–2542, <https://doi.org/10.1109/ICSMC.2008.4811677>.
- [28] A. Lova, C. Maroto, P. Tormos, A multi-criteria heuristic method to improve resource allocation in multi-project scheduling, *Eur. J. Oper. Res.* 127 (2) (2000) 408-424, [https://doi.org/10.1016/S0377-2217\(99\)00490-7](https://doi.org/10.1016/S0377-2217(99)00490-7).
- [29] A. Lova, P. Tormos, Combining random sampling and backward-forward heuristics for resource-constrained multi-project scheduling, *Proceedings of the Eighth International Workshop on Project Management and Scheduling*, Valencia, Spain, 2002, pp. 244-248.
- [30] A. Elazouni, M. Abido, Multi-objective evolutionary finance-based scheduling: Individual projects within a portfolio, *Autom. Constr.* 20 (7) (2011) 755-766, <https://doi.org/10.1016/j.autcon.2011.03.010>.

- [31] J. Xu, Z. Zhang, A fuzzy random resource-constrained scheduling model with multiple projects and its application to a working procedure in a large-scale water conservancy and hydropower construction project, *J. Sched.* 15 (2) (2012) 253-272, <https://doi.org/10.1007/s10951-010-0173-1>.
- [32] L. Florez, D. Castro-Lacouture, A.L. Medaglia, Sustainable workforce scheduling in construction program management, *J. Oper. Res. Soc.* 64 (8) (2013) 1169-1181, <https://doi.org/10.1057/jors.2012.164>.
- [33] J. Gang, J. Xu, Y. Xu, Multi-project resources allocation model under fuzzy random environment and its application to industrial equipment installation engineering, *J. Appl. Math.* 2013 19p. <http://dx.doi.org/10.1155/2013/818731>.
- [34] A. Singh, Resource constrained multi-project scheduling with priority rules & analytic hierarchy process, *Procedia Eng.* 69 (2014) 725-734, <https://doi.org/10.1016/j.proeng.2014.03.048>.
- [35] A. Can, G. Ulusoy, Multi-project scheduling with two-stage decomposition, *Ann. Oper. Res.* 217 (1) (2014) 95-116, <https://doi.org/10.1007/s10479-014-1555-0>.
- [36] M.G. Speranza, C. Vercellis, Hierarchical models for multi-project planning and scheduling, *Eur. J. Oper. Res.* 64 (2) (1993) 312–325, [https://doi.org/10.1016/0377-2217\(93\)90185-P](https://doi.org/10.1016/0377-2217(93)90185-P).
- [37] A. Shahsavar, A.A. Najafi, S.T.A. Niaki, Three self-adaptive multi-objective evolutionary algorithms for a triple-objective project scheduling problem, *Comput. Ind. Eng.* 87 (1) (2015) 4-15, <https://doi.org/10.1016/j.cie.2015.04.027>.
- [38] M. Gagnon, F.F. Boctor, G. d'Avignon, Multicriteria project scheduling with resource availability cost, Faculty of Business Administration, Laval University Working Paper, 2005, <http://www.fsa.ulaval.ca/sirul/2005-011.pdf>.
- [39] F.B. Talbot, Resource-constrained project scheduling with time-resource tradeoffs: The non-preemptive case, *Manag. Sci.* 28 (10) (1982) 1197-1210, <https://doi.org/10.1287/mnsc.28.10.1197>.
- [40] J. Węglarz, J. Józefowska, M. Mika, G. Waligóra, Project scheduling with finite or infinite number of activity processing modes – A survey, *Eur. J. Oper. Res.* 208 (3) (2011) 177-205, <https://doi.org/10.1016/j.ejor.2010.03.037>.
- [41] K. Deb, A. Pratap, S. Agarwal, T.A.M.T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Trans. Evol. Comput.* 6 (2) (2002) 182-197, <https://doi.org/10.1109/4235.996017>.

- [42] N. Balouka, I. Cohen, A. Shtub, Extending the multimode resource-constrained project scheduling problem by including value considerations, *IEEE Trans. Eng. Manag.* 63 (1) (2016) 4-15, <https://doi.org/10.1109/TEM.2015.2497209>.
- [43] J. Xiao, Z. Wu, X.X. Hong, J.C. Tang, Y. Tang, Integration of electromagnetism with multi-objective evolutionary algorithms for RCPSP, *Eur. J. Oper. Res.* 251 (1) (2016) 22-35, <https://doi.org/10.1016/j.ejor.2015.10.059>.
- [44] L. Wang, X.L. Zheng, A knowledge-guided multi-objective fruit fly optimization algorithm for the multi-skill resource constrained project scheduling problem, *Swarm Evol. Comput.* 38 (2018) 54-63, <https://doi.org/10.1016/j.swevo.2017.06.001>.
- [45] K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints, *IEEE Trans. Evol. Comput.* 18 (4) (2014) 577-601, <https://doi.org/10.1109/TEVC.2013.2281535>.
- [46] K.Y. Li, R.J. Willis, An iterative scheduling technique for resource-constrained project scheduling, *Eur. J. Oper. Res.* 56 (3) (1992) 370-379, [https://doi.org/10.1016/0377-2217\(92\)90320-9](https://doi.org/10.1016/0377-2217(92)90320-9).
- [47] F. Ballestín, R. Blanco, Theoretical and practical fundamentals, in: C. Schwindt, J. Zimmermann (Eds), *Handbook on Project Management and Scheduling Vol. 1*, Springer International Publishing, Switzerland, 978-3-319-05443-8, 2015, pp. 411-427, https://doi.org/10.1007/978-3-319-05443-8_19.
- [48] G. Ulusoy, F. Sivrikaya-Şerifoğlu, Ş. Şahin, Four payment models for the multi-mode resource constrained project scheduling problem with discounted cash flows, *Ann. Oper. Res.* 102 (1) (2001) 237-261, <https://doi.org/10.1023/A:1010914417817>.
- [49] S. Hartmann, Project scheduling with multiple modes: A genetic algorithm, *Ann. Oper. Res.* 102 (1) (2001) 111-135, <https://doi.org/10.1023/A:1010902015091>.
- [50] R. Kolisch, Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation, *Eur. J. Oper. Res.* 90 (2) (1996) 320-333, [https://doi.org/10.1016/0377-2217\(95\)00357-6](https://doi.org/10.1016/0377-2217(95)00357-6).
- [51] S. Hartmann, A competitive genetic algorithm for resource-constrained project scheduling, *Nav. Res. Logist.* 45 (7) (1998) 733-750, [https://doi.org/10.1002/\(SICI\)1520-6750\(199810\)45:7<733::AID-NAV5>3.0.CO;2-C](https://doi.org/10.1002/(SICI)1520-6750(199810)45:7<733::AID-NAV5>3.0.CO;2-C).
- [52] F. Sivrikaya-Şerifoğlu, A New Uniform Order-Based Crossover Operator for Genetic Algorithm Applications to Multi-Component Combinatorial Optimization Problems (PhD dissertation), Boğaziçi University, Istanbul, 1997.

- [53] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, Massachusetts, 1989, 978-0201157673.
- [54] R.H. Myers, D.C. Montgomery, C.M. Anderson-Cook, Response Surface Methodology: Process and Product Optimization Using Designed Experiments, John Wiley & Sons, New Jersey, 2016, 978-1-118-91601-8.
- [55] A.A. Najafi, S.T.A. Niaki, M. Shahsavari, A parameter-tuned genetic algorithm for the resource investment problem with discounted cash flows and generalized precedence relations, *Comput. Oper. Res.* 36 (11) (2009) 2994–3001, <https://doi.org/10.1016/j.cor.2009.01.016>.
- [56] E. Zitzler, L. Thiele, Multi-objective optimization using evolutionary algorithms—A comparative case study, *Proceedings of the Fifth International Conference on Parallel Problem Solving from Nature*, Amsterdam, The Netherlands, 1998, pp. 292-301, <https://doi.org/10.1007/BFb0056872>.
- [57] E. Zitzler, Evolutionary Algorithms for Multi-Objective Optimization: Methods and Applications (PhD dissertation), Swiss Federal Institute of Technology (ETH), Zürich, 1999, <https://sop.tik.ee.ethz.ch/publicationListFiles/zitz1999a.pdf>.
- [58] R. Kolisch, A. Sprecher, PSPLIB - A project scheduling problem library: OR software-ORSEP operations research software exchange program, *Eur. J. Oper. Res.* 96 (1) (1997) 205-216, [https://doi.org/10.1016/S0377-2217\(96\)00170-1](https://doi.org/10.1016/S0377-2217(96)00170-1).
- [59] R. Kolisch, Project Scheduling under Resource Constraints, Physica - Verlag, Heidelberg, 1995, 978-3-642-50296-5, <https://doi.org/10.1007/978-3-642-50296-5>.
- [60] A. Sprecher, S. Hartmann, A. Drexl, An exact algorithm for project scheduling with multiple modes, *OR Spectr.* 19 (3) (1997) 195-203, <https://doi.org/10.1007/BF01545587>.
- [61] E. M. Goldratt, Critical Chain, North River Press, Great Barrington, 1997, 978-0884271536, <https://northriverpress.com/critical-chain-eliyahu-m-goldratt/>.
- [62] M. Ning, H. Zhengwen, T. Jia, N. Wang, Metaheuristics for multi-mode cash flow balanced project scheduling with stochastic duration of activities, *Autom. Constr.* 81 (2017) 224-233, <https://doi.org/10.1016/j.autcon.2017.06.011>.