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

Fikri Küçüksayacigil
Iowa State University, Ames, IA, USA
fksayaci@iastate.edu

Gündüz Ulusoy
Sabanci University, Istanbul, Turkey
gunduz@sabanciuniv.edu

Abstract

Here we consider a bi-objective, multi-project, multi-mode, resource-constrained project-scheduling problem. The objectives were to minimize the makespan, minimize the mean of the flow times for individual projects, minimize the mean completion times for individual projects and maximize the total net present value of all projects. 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 post-processing and for new population generation. 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 with more diversity. A detailed fine-tuning process was conducted by employing a response surface optimization method. An extensive computational study was performed. Managerial insights are presented.

Keywords: Bi-objective genetic algorithm, Multi-project multi-mode RCPSP (MRCMPSP), Multi-objective MRCMPSP, Backward–forward scheduling, Injection.

1. Introduction

With changing business paradigms 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 (Blazewicz *et al.*, 1983). In recent decades, an extensive amount of work has been accomplished for developing exact and heuristic algorithms for solution to 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).

In this paper, we will deal with the bi-objective MRCMPSP problem. The most common and frequently used objective in project scheduling is the minimization of the makespan of projects (minC_{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 (Demeulemeester and Herroelen, 2002). Another significant objective in project scheduling is to maximize the net present value of projects (maxNPV). It has been preferred as a financial objective by many researchers and practitioners, because it is claimed to better reflect the financial aspects of the decision environment (Gu *et al.*, 2015). In the case when only costs are involved, then the objective becomes the minimization of NPV. In addition to C_{max} and NPV, the project manager might also be interested in minimizing the mean flow time of individual projects (minMFT) so that the mean throughput times of projects are reduced leading to a general reduction in work-in-progress as well. Minimization of mean completion time for individual projects (minMCT) can be considered as another relevant time-based objective. A decision maker may seek a project schedule that uses renewable resources more effectively leading to acceptable project completion times. MCT, therefore, can be closely associated with customer satisfaction and might lead to more favorable cash profiles.

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 is presented by Vanhoucke (2009). In that formulation, a soft deadline constraint is imposed allowing 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 ending 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. Since both C_{max} and NPV are included in the objective function, it can be considered as a multi-objective optimization model. Khalili *et al.*, (2013) considered the bi-objective problem of minimizing C_{max} and maximizing NPV simultaneously for RCPSP by approximating the Pareto front. Two meta-heuristic algorithms were employed for solving the bi-objective RCPSP: multi-population GA (Cochran *et al.*, 2003) and two-phase sub-population GA (Chang *et al.*, 2005).

C_{max} and NPV intuitively conflict, but they can be mutually supporting under certain conditions. Smith-Daniels and Aquilano (1987) 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

Ulusoy and Özdamar (1995). 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 deals with 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 (minC_{max}/maxNPV); (*ii*) minimization of MFT and maximization of NPV (minMFT/maxNPV) and (*iii*) minimization of MCT and maximization of NPV (minMCT/maxNPV). By dealing with three different bi-objective models we aimed to gain a wider perspective of the decision problem.

In the next sections, we first present the relevant studies from published 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. However, to the best of our knowledge, there are few studies simultaneously analyzing 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.

The paper by Liu and Wang (2009) 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. Gang *et al.* (2013) solved multi-project resource allocation problems with a bi-level approach under stochastic activity durations and costs. The developed framework minimized cost and C_{max}. Delays in project completions were accepted as objectives and they were combined into a single objective function with a weighted sum approach. Xu and Feng (2014) developed a particle swarm optimization algorithm for MRCMPSP under a fuzzy random environment. C_{max}, cost and quality of projects were accepted as objectives and these were combined into a single formulation with a weighted sum approach. Wang *et al.* (2014) proposed a genetic algorithm (GA) to solve multi-objective RCMPSP with time, cost, quality and robustness being the objectives. The objective function was defined as the weighted average of the utility function of each objective.

We grouped the following papers under the second approach. Kim and Schniederjans (1989) 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 (1994) 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. Aouni *et al.* (2015) presented a goal-programming model to provide a baseline solution for a single project under resource constraints reflecting the best compromise based on the project manager's preference structure. Three objectives were considered: the project duration, the project cost and the quantity of allocated resources. Lova *et al.* (2000) 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 (2002) considered mean project delay and overall C_{max} as two objectives and employed a combination of random sampling with backward–forward heuristics. Viana and de Sousa (2000) implemented the tabu search algorithm developed by Hansen (1997) to solve a multi-mode RCPSP with three objectives that were minimized: the makespan, the weighted lateness of activities and the violation of resource constraints. Elazouni and Abido (2011) 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 (2012) 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.* (2013) 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 an ϵ-constraint method and implemented it for a real construction project. Singh (2014) 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 (2014) created a hierarchical model for the problem, as proposed earlier by Speranza and Vercellis (1993), 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.* (2015) considered three objectives in a resource constrained multi-project problem setting. The objectives were the minimization of the overall makespan 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.* (2005) 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.

Table 1 Notation 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, $p = 1, \dots, \lvert P \rvert$
J_p, j	Set of activities in project p and activity index, $j = 1, \dots, \lvert J_p \rvert$ for project p
M_{pj}, m	Set of modes of activity j in project p and mode index, $m = 1, \dots, \lvert M_{pj} \rvert$ 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, $r = 1, \dots, \lvert R \rvert$
N, n	Set of non-renewable resources and non-renewable resource index, $n = 1, \dots, \lvert N \rvert$
$\hat{r}_{pjmr}$	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 ending time 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
$\boldsymbol{x}$	Set of decision variables
V	Number of objective functions
$f_k(\boldsymbol{x})$	k^{th} objective function, $k = 1, \ldots, V$
$\boldsymbol{f}(\boldsymbol{x})$	Vector of objective functions

As the literature review above reveals, there is no study reported that addresses the multi-objective MRCMPSP. Furthermore, the bi-objective pairs (minMFT/maxNPV) and (minMCT/maxNPV) have not been investigated before even in a single project decision environment. 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 focus on the bi-objective MRCMPSP in this paper. Activities are non-preemptive and have multiple modes. The project network is of activity-on-node (AON) type with finish-to-start precedence relation with zero time lags. The multiple projects are represented as a composite project network in general with a dummy start node and a dummy end node. The notation for the mathematical formulation is given in Table 1. The mathematical formulation for the problem denoted by MF1 is presented in Equations (1) to (6). This formulation is an extension of the single objective formulation given by Talbot (1982).

MF1

$$\text{Opt}\, \boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_V(\boldsymbol{x})] \tag{1}$$

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

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

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

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

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

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, the latest ending time of the dummy end 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 Equation (1). Equation (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 (Talbot, 1982; Węglarz *et al.*, 2011). The decision variables x_{pjmt} are defined in Equation (6).

Here we assumed 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 MF1 without causing additional difficulty.

In this study, we assume that activity costs are incurred at their completion times (excluding dummy activities because there is no cost defined for them). Moreover, a lump sum payment is received at the completion of each project. Finally, each project starts with an upfront investment. All these financial parameters enabled us to calculate the NPV of a given multi-project schedule by using an appropriate discount factor. Note that financial parameters were set in a way that each project had a positive NPV.

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 components of a are at least as good as those of b and at least one 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). 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 (Li and Willis, 1992;

Özdamar and Ulusoy, 1996) to the solutions of NSGA-II as an improvement procedure. As stated by Ballestin and Blanco (2015), BFP or its modifications are versatile techniques that can be employed for the solution of multi-objective RCPSPs. Furthermore, we applied an injection procedure to increase the diversity in the solution set of NSGA-II.

4.2. *Individual representation*

An individual is represented by a double list consisting of the precedence feasible activity list (henceforth, we will call it the feasible list) and the mode list (Ulusoy *et al.*, 2001; Hartmann, 2001). In the feasible list, activities are replaced into genes in a way that all the predecessors of an activity appear before it. By doing so, precedence relationships between the activities are satisfied. The mode list consists of modes assigned to activities from their mode sets.

4.3. *Initial population generation*

In this study, the initial population was generated by randomly creating feasible lists and corresponding mode lists. To create a feasible list, a dummy start 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, starting and ending times were assigned to the activities by using scheduling schemes. Demeulemeester and Herroelen (2002) 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 (Kolisch, 1996), 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 Deb *et al.* (2002). Since we generated the initial population randomly, some individuals might 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 (Deb *et al.*, 2002). 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 is used. An individual with larger crowding distance is more preferable.

4.6. *Forming the next generation*

Three different crossover operators proposed in published literature were implemented in this study. One-point and two-point crossover procedures were defined by Hartmann (1998) for the single mode case. Hartmann (2001) expanded one-point crossover for use in the case of multiple modes. We implemented one-point crossover as defined by Hartmann (2001) 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 Sivrikaya-Şerifoğlu (1997).

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 (Hartmann, 2001). 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.*, 2002 used the same selection procedure in NSGA-II), in which rank and crowding distance values determined the winner (Goldberg, 1989). The individual with a better rank between two individuals is selected as the parent. If there is a tie in rank values, the individual with higher crowding distance is 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 Deb *et al.* (2002). 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*

The parameters of the algorithm (population size, number of generations, crossover rate and mutation rate) were determined by response surface optimization (Myers *et al.*, 2009), 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 Najafi *et al.*, 2009). In published literature, several performance measures have been proposed to evaluate a given set of non-dominated solutions. We preferred hypervolume (Zitzler and Thiele, 1998), maximum spread (Zitzler, 1999) 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 dealt with 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 (1998) suggested that bounds or optimum values for each objective can be taken separately to form a reference point. It is reported by the authors that hypervolume does not need scaling of the objective values.

As indicated by Zitzler and Thiele (1998), it is better not to stick to a single performance measure. Instead, one should take advantage of several performance measures simultaneously. For instance, Zitzler and Thiele (1998), in addition to hypervolume, compared two approximation sets by investigating how many solutions in the second one are dominated by the first one and vice versa.

For minC_{max}/maxNPV, suppose the approximation set appears as in Figure 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.

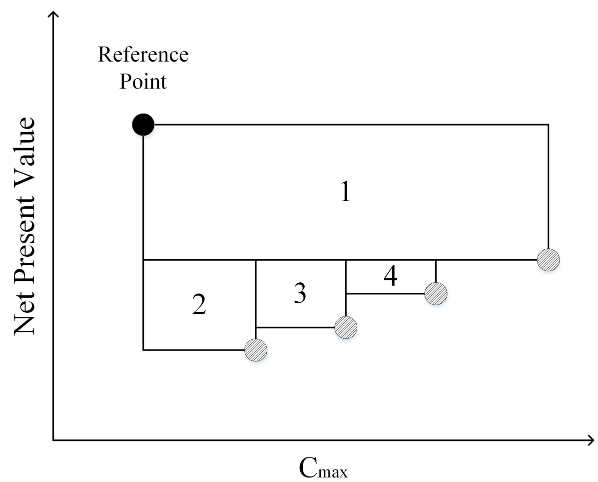

Figure 1 Hypervolume measure for minC_{max}/maxNPV

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 ending time for all projects by disregarding the resource requirements. For NPV, setting a reference value is complicated because each project has an investment cost, a lump sum payment and execution costs for its activities. It was difficult to set a reference value quickly (in this case, we sought a value as large as possible – the so-called upper bound) for NPV. 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 are 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 Figure 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, 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:

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

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 (Kolisch and Sprecher, 1997) were utilized. 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 dealing with 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 minC_{max}/maxNPV, 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 starting and ending 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 the ending and starting 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 (Figure 2).

Figure 2 presents the BFP procedure applied to a single individual. Lines 1 to 4 in Figure 2 correspond to a backward pass and lines 5 to 9 correspond to a forward pass. We emphasize that resource and precedence constraints should not be violated in lines 3 and 7. In other words, in each time period, the usage of renewable resources should not exceed their corresponding limits. Since modes of the activities do not change, it is not necessary to check non-renewable resource consumption for feasibility. As for the precedence relationships, an activity should not start earlier than the completion of all its predecessor activities.

Since NSGA-II might have many non-dominated solutions, different lists, $\mathcal{L}$, resulting from the BFP procedures applied separately to these solutions are combined. At the end, dominated solutions are removed from the combined list and the resulting individuals were presented as the output of 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”.

Algorithm: BFP Procedure on an Individual
$\mathcal{L}$: Empty list of individuals
a_o: Individual on which BFP will be applied
$\mathscr{s}_j$: Starting time of activity j of a_0
$\mathscr{f}_j$: Ending time of activity j of a_0
$T > 0$: Predetermined iteration number

While $(T > 0)$
  $\mathcal{L} = \mathcal{L} \cup \{a_0\}$
1 Sort the activities of a_0 in decreasing order of $\mathscr{f}_j$
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 $\mathscr{s}_{Dummy\ start\ activity} > 0$
    Shift all activities of $a_0{}'$ to the left so that $\mathscr{s}_{Dummy\ start\ activity} = 0$
    Label this individual as a_1 and calculate objective function values
    $\mathcal{L} = \mathcal{L} \cup \{a_1\}$
  Else if $\mathscr{s}_{Dummy\ start\ activity} = 0$
5   Sort the activities of $a_0{}'$ in increasing order of $\mathscr{s}_j$
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    $\mathcal{L} = \mathcal{L} \cup \{a_2\}$
  If there are dominated individual in $\mathcal{L} = \{a_0, a_1, a_2\}$
    Remove them from $\mathcal{L}$
  Else if there is no dominated individual
    $T = T - 1$
  Label the last individual in $\mathcal{L}$ as a_0
  Go to step 1
End

Figure 2 BFP procedure on an individual.

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 is defined as a solution in which the projects are 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 the Computational Study section 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 $\lceil 0.114G \rceil$ generations, 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 the study by Can and Ulusoy (2014). They used the single project instances presented in PSBLIB (Kolisch and Sprecher, 1997) 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 by Can and Ulusoy (2014). Lump sum payments for dummy end activities of projects and investment costs for dummy start 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.

To represent a variety of different environmental factors, Can and Ulusoy (2014) created three problem sets of instances denoted by A, B and C. Problem set A was formed 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 were employed here as defined by Kolisch (1995). Combinations of these four variable factors with three levels of each result in 81 instances. Set A included 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 included 14 projects each with 10 non-dummy activities. Problem set B focused on the effects of different number of projects and activities. In these multi-project instances, three levels were set for the number of projects and seven levels were set for the number of activities. The resource factor for both renewable and non-renewable resources were fixed, whereas the resource strength was assigned two levels for each of the resource categories resulting in 84 problems. In problem set C, a multi-project environment, which was heterogeneous in terms of project sizes, was emphasized by generating multi-project instances consisting of projects with different number of activities resulting in 27 instances. Three multi-project groups, each with 9 multi-projects, were formed and different levels of resource strengths were assigned. In the first group, equal numbers of relatively small, medium and large projects were combined. In the second group, a few larger projects were grouped together with a collection of smaller sized projects. In the third group, a few smaller projects were added to a group of relatively large projects. Further information concerning these problem sets can be found in Can and Ulusoy (2014).

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 (Sprecher *et al.*, 1997).

As stated before, BFP was applied to both modes "BFP on the Archive" and "BFP in the Intermediate Stages". The objective combination minC_{max}/maxNPV was utilized for solving the A, B and C sets of test instances with both modes of BFP. However, the objective combinations minMFT/maxNPV and minMCT/maxNPV were implemented only with the BFP mode that outperforms the other for the minC_{max}/maxNPV objective combination.

7.1. *BFP on the Archive for minC_{max}/maxNPV*

Table 3 summarizes the results of NSGA-II and BFP on the Archive implementations. 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 (AC_{max}) were calculated first for each instance and then the average of AC_{max} values for all the instances were computed ($\overline{AC_{max}}$). As for NPV, ANPV denotes the average of NPV values for each instance and $\overline{ANPV}$ denotes the average of ANPVs for all the instances.

Since the same instances were used to run NSGA-II and BFP on the Archive, a paired *t*-test (with 0.95 confidence level) was conducted for AC_{max} and ANPV comparisons. The hypotheses were as follows: For AC_{max}: [H_0: AC_{max} of NSGA = AC_{max} of BFP on the Archive; H_A: AC_{max} of NSGA > AC_{max} of BFP on the Archive]. For ANPV: [H_0: ANPV of NSGA = ANPV of BFP on the Archive; H_A: ANPV of NSGA < ANPV of BFP on the Archive]. For 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 outperforms NSGA-II in obtaining better AC_{max} values for all test sets. On the other hand, a significant improvement was not observed for ANPV.

Table 3 Comparison of NSGA-II and BFP on the Archive

Test Sets	*# of Instances*	$\overline{AC_{max}}$			$\overline{ANPV}$		
		NSGA-II	*BFP on the Archive*	*p-value*	*NSGA-II*	*BFP on the Archive*	*p-value*
A	81	110.97	106.96	3E−27	281,806	282,119	0.17
B	84	114.75	110.69	2E−28	332,864	333,123	0.20
C	27	108.31	104.44	8E−11	385,864	385,461	0.32

7.2. *BFP in the Intermediate Stages for minC_{max}/maxNPV*

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.

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

The test instances for the same instance set are separated into subgroups (INSSUB). The instances in the same subgroups had similar complexity, which was adjusted in the data generation phase in Can and Ulusoy (2014). 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.

It was observed that the algorithm couldn't 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 $\overline{\mathrm{AC}_{\mathrm{max}}}$ and $\overline{\mathrm{ANPV}}$ values in NSGA-II columns are the same as those presented in Table 3. In other words, we run the algorithm to report the results of NSGA-II in Table 3 and run it again to report the results of BFP in the Intermediate Stages in Table 5. Though it can be claimed that they are separate implementations to compare, we justify it by pointing out that the number of instances are sufficiently large to observe any differences.

Table 5 Comparison 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	0.12
B	84	114.75	113.38	5E−03	332,864	328,669	6E−07
C	27	108.31	105.49	9E−04	385,864	381,855	0.001

A paired *t*-test (with 0.95 confidence level) was conducted for AC_{max} and ANPV comparisons. The hypotheses were as follows: For AC_{max}: [H_0: AC_{max} of NSGA = AC_{max} of BFP in the Intermediate Stages; H_A: AC_{max} of NSGA > AC_{max} of BFP in the Intermediate Stages]. For ANPV: [H_0: ANPV of NSGA = ANPV of BFP in the Intermediate Stages; H_A: ANPV of NSGA > ANPV of BFP in the Intermediate Stages]. It can be inferred from Table 5 that BFP in the Intermediate Stages is superior to NSGA-II for all three test sets in 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 have enough evidence to reject H_0 and accept H_A.

From the 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.

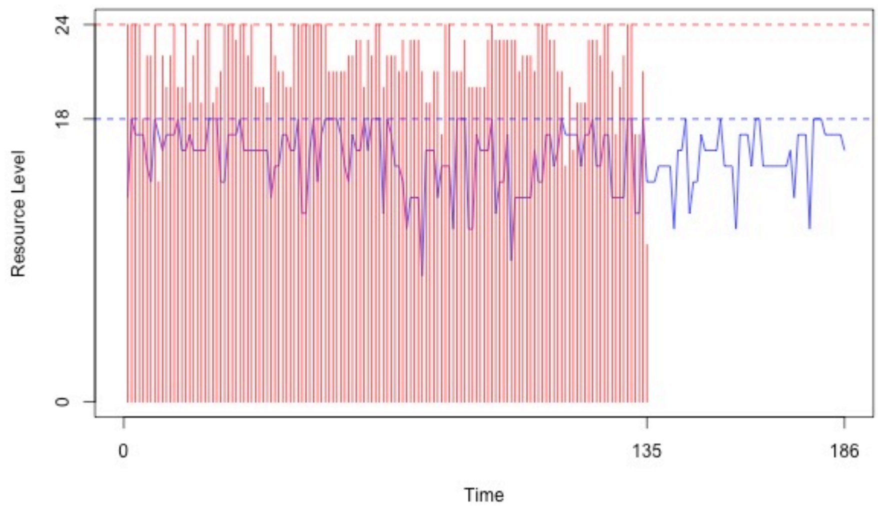

Figure 3 Levels of renewable resource 1 for instances A33_11 and A33_21

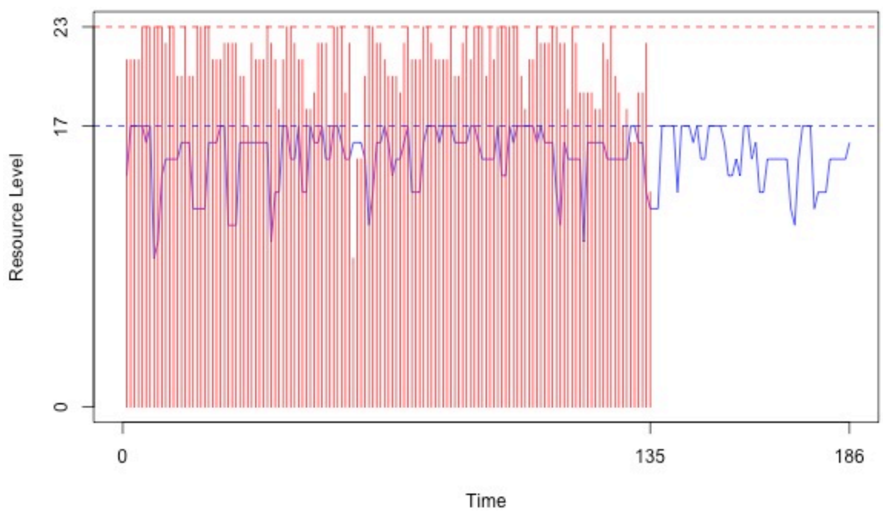

Figure 4 Levels of renewable resource 2 for instances A33_11 and A33_21

Figure 3 and Figure 4 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.

Figure 5 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.

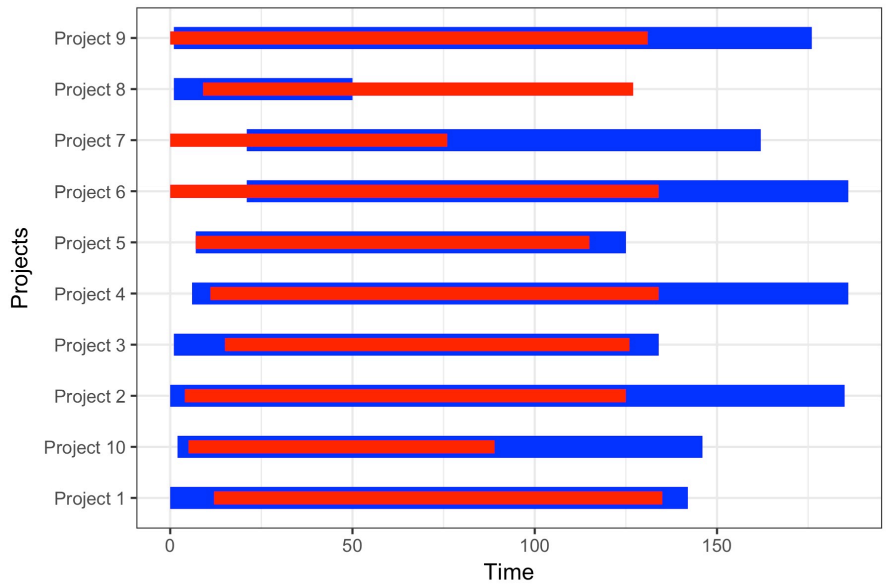

Figure 5 Gantt charts for projects in instances A33_11 and A33_21

The first solution set was in fact the set presented in Table 3 under the BFP on the Archive heading. The comparison is performed with a paired-*t* test and a significance level of 0.95. The hypotheses were as follows: For AC_{max}: [H_0: AC_{max} of BFP on the Archive with injection = AC_{max} of BFP on the Archive without injection; H_A: AC_{max} of BFP on the Archive with injection > AC_{max} of BFP on the Archive without injection]. For ANPV: [H_0: ANPV of BFP on the Archive with injection = ANPV of BFP on the Archive without injection; H_A: ANPV of BFP on the Archive with injection > ANPV of BFP on the Archive without injection].

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.

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	0.00
B	84	110.69	118.42	8E−26	333,123	354,743	4E−12
C	27	104.44	110.99	2E−08	385,461	400,699	0.00

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 hypotheses were as follows: For AMFT:

[H_0: AMFT of BFP on the Archive with injection = AMFT of BFP on the Archive without injection; H_A: AMFT of BFP on the Archive with injection < AMFT of BFP on the Archive without injection]. For AMCT: [H_0: AMCT of BFP on the Archive with injection = AMCT of BFP on the Archive without injection; H_A: AMCT of BFP on the Archive with injection < AMCT of BFP on the Archive without injection]. It was clearly seen that the injection procedure was highly effective in reducing the mean completion and flow times of projects.

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

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	0.00	88.59	76.54	0.00
B	84	91.40	66.75	5E−21	94.65	80.48	1E−23
C	27	79.32	61.40	8E−09	82.21	73.91	7E−11

As stated before, the main idea of injecting new solutions was to maintain the diversity of the algorithm. Hence, it was also necessary to report the number of non-dominated solutions obtained by the two methods. Table 8 displays the results. The hypotheses were as follows: H_0: The number of solutions obtained by BFP on the Archive with injection = The number of solutions obtained by BFP on the Archive without injection; H_A: The number of solutions obtained by BFP on the Archive with injection > The number of solutions obtained by BFP on the Archive without 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 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	3E−16
B	84	2.85	7.29	1E−21
C	27	3.37	7.37	5E−07

7.5. *Solutions for minMFT/maxNPV and minMCT/maxNPV*

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 $\overline{\text{AC}_{\text{max}}}$, $\overline{\text{ANPV}}$, $\overline{\text{AMFT}}$, and $\overline{\text{ANPV}}$ – 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 minMCT/maxNPV, we calculated $\overline{\text{AMCT}}$ and $\overline{\text{ANPV}}$ 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*			
	$\overline{ANPV}$	$\overline{AMFT}$	$\overline{AMCT}$	$\overline{AC_{max}}$
minC_{max}/maxNPV	*296,752*	60.8	76.5	*112.99*
minMFT/maxNPV	*278,305*	*37.7*	103.7	194.1
minMCT/maxNPV	*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*			
	$\overline{ANPV}$	$\overline{AMFT}$	$\overline{AMCT}$	$\overline{AC_{max}}$
minC_{max}/maxNPV	*354,743*	66.7	80.5	*118.4*
minMFT/maxNPV	*311,640*	*42.7*	140	266
minMCT/maxNPV	*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	*400,699*	61.4	73.9	*111*
minMFT/maxNPV	*356,321*	*38.5*	131.7	259.8
minMCT/maxNPV	*407,492*	50.5	*64.3*	128

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

In all problem sets, $\overline{\text{AMCT}}$, $\overline{\text{AMFT}}$, and $\overline{\text{AC}_{\text{max}}}$ reached their best values when the corresponding objective was part of the objective combination. $\overline{\text{ANPV}}$, 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. So, as the completion times of the projects decreased, the total NPV resulting from these projects increased.

One other point attracting attention was that $\overline{\text{AMCT}}$ 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 $\overline{\text{AMCT}}$, $\overline{\text{AC}_{\text{max}}}$, also had its highest value for all problem sets for the objective combination minMFT/maxNPV. A similar line of thought can be deduced for $\overline{\text{AC}_{\text{max}}}$ as that given above for $\overline{\text{AMCT}}$.

Note that there is a substantial difference between $\overline{\text{AMFT}}$ values when objective combinations are minC_{max}/maxNPV and minMCT/maxNPV. It was essentially a result of obtaining different starting and ending 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 starting and ending times for instance A31_31

Projects	*minC_{max}/maxNPV*		*minMCT/maxNPV*	
	Starting Time	*Ending Time*	*Starting Time*	*Ending 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 starting and ending times, as well as project durations, were different for minC_{max}/maxNPV and minMCT/maxNPV. Fifty out of 270 activities had modes with smaller durations for minMCT/maxNPV when compared to modes assigned for minC_{max}/maxNPV.

Table 13 Project starting and ending times for instance B1518_21

Projects	*min C_{max}/maxNPV*		*minMCT/maxNPV*	
	Starting Time	*Ending Time*	*Starting Time*	*Ending 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 starting and ending times for instance C2_32. Forty-seven out of 252 activities were assigned modes with longer durations for minMCT/maxNPV.

Table 14 Project starting and ending times for instance C2_32

Projects	*min C_{max}/maxNPV*		*minMCT/maxNPV*	
	Starting Time	*Ending Time*	*Starting Time*	*Ending 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

8. Conclusions

In this paper, we studied bi-objective decision problems: minC_{max}/maxNPV, minMFT/maxNPV and minMCT/maxNPV in the case of multi-project, multi-mode RCPSP. We implemented NSGA-II with a BFP extension with two modes 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 injection procedure proved to be very efficient in leading to a higher number of non-dominated solutions with more diversity. Moreover, it was able to find solutions with better NPV values. An extensive computational study was performed using existing sets of test problem data that had been extended to include an initial investment cost, activity costs and lump sum payment incurred at the termination of each project.

The computational study resulted in a number of managerial insights:

(*i*) Increasing the capacities of renewable resources considerably decreased the completion times and hence, the overall C_{max} of the projects.

(*ii*) As the computational results clearly indicated, trying to reduce the mean flow time of projects by adopting minMFT/maxNPV 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 low levels of flow times for the projects above all the other objectives for some reason – not part of the problem environment described here – this objective combination would not be recommended.

(*iii*) 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 (minMCT) rather than minimizing the overall C_{max} (minC_{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 customer when completion times of the projects are reduced. The objective combination minMCT/maxNPV still leads to acceptable values for the overall C_{max}, since the overall C_{max} is the completion time of the last project. It also leads to smaller mean flow time values compared to the minC_{max}/maxNPV objective combination.

(*iv*) By considering the objective combination minC_{max}/maxNPV, we observed that the results obtained for the mean objective values were relatively close but inferior to those obtained by minMFT/maxNPV. On the other hand, they are preferable to those obtained by the minMFT/maxNPV objective combination.

(*v*) Finally, 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 minC_{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 (see, e.g., Ulusoy *et al.*, 2001). An expected managerial result – that is shared above – states that increasing the capacities of renewable resources considerably decreased the completion times and hence, the overall C_{max} of the projects. Analyzing the marginal impact of increasing resource budgets for the objectives under consideration would be another research area of particularly 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. Considering tardiness, for example, would include considering due dates into the problem definition, adding a new dimension of interest.

References

Aouni B, d'Avignon G and Gagnon M (2015). Goal programming for multi-objective resource-constrained project scheduling. In: Schwindt C and Zimmermann J (eds). *Handbook on Project Management and Scheduling Vol. 1* pp. 429-442. Springer International Publishing: Switzerland.

Ballestín F and Blanco R (2015). Theoretical and practical fundamentals. In: Schwindt C and Zimmermann J (eds). *Handbook on Project Management and Scheduling Vol. 1* pp. 411-427. Springer International Publishing: Switzerland.

Blazewicz J, Lenstra JK and Kan AR (1983). Scheduling subject to resource constraints: Classification and complexity. *Discrete Applied Mathematics* **5**(1):11-24.

Brucker P, Drexl A, Möhring R, Neumann K and Pesch E (1999). Resource-constrained project scheduling: Notation, classification, models, and methods. *European Journal of Operational Research* **112**(1):3-41.

Can A and Ulusoy G (2014). Multi-project scheduling with two-stage decomposition. *Annals of Operations Research* **217**(1):95-116.

Chang PC, Chen SH and Lin KL (2005). Two-phase sub population genetic algorithm for parallel machine-scheduling problem. *Expert Systems with Applications* **29**(3):705-712.

Chen VY (1994). A 0–1 goal programming model for scheduling multiple maintenance projects at a copper mine. *European Journal of Operational Research* **76**(1):176-191.

Cochran JK, Horng SM and Fowler JW (2003). A multi-population genetic algorithm to solve multi-objective scheduling problems for parallel machines. *Computers & Operations Research* **30**(7):1087-1102.

Deb K, Pratap A, Agarwal S and Meyarivan TAMT (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* **6**(2):182-197.

Demeulemeester EL and Herroelen WS (2002). Project Scheduling - A Research Handbook. Kluwer Academic Publishers, Boston.

Elazouni A and Abido M (2011). Multi-objective evolutionary finance-based scheduling: Individual projects within a portfolio. *Automation in Construction* **20**(7):755-766.

Florez L, Castro-Lacouture D and Medaglia AL (2013). Sustainable workforce scheduling in construction program management. *Journal of the Operational Research Society* **64**(8):1169-1181.

Gagnon M, Boctor FF and d'Avignon G (2005). *Multicriteria Project Scheduling with Resource Availability Cost*. Faculty of Business Administration, Laval University Working paper.

Gang J, Xu J and Xu Y (2013). Multi-project resources allocation model under fuzzy random environment and its application to industrial equipment installation engineering. *Journal of Applied Mathematics* **2013**:19p. http://dx.doi.org/10.1155/2013/818731.

Goldberg DE (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading, Massachusetts.

Gu H, Schutt A, Stuckey PJ, Wallace MG and Chu G (2015). Exact and heuristic methods for the resource-constrained net present value problem. In: Schwindt C and Zimmermann J (eds). *Handbook on Project Management and Scheduling Vol. 1* pp. 299-318. Springer International Publishing: Switzerland.

Hansen MP (1997). *Tabu Search for Multi-objective Optimization: MOTS*. Technical Report. Institute of Mathematical Modeling, Technical University of Denmark.

Hartmann S (1998). A competitive genetic algorithm for resource-constrained project scheduling. *Naval Research Logistics* **45**(7):733-750.

Hartmann S (2001). Project scheduling with multiple modes: A genetic algorithm. *Annals of Operations Research* **102**(1):111-135.

Hartmann S and 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 and Demeulemeester E (1998). Resource-constrained project scheduling: A survey of recent developments. *Computers & Operations Research* **25**(4):279-302.

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

Khalili S, Najafi AA and Niaki STA (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.

Kim SO and Schniederjans MJ (1989). Heuristic framework for the resource constrained multi-project scheduling problem. *Computers & Operations Research* **16**(6):541-556.

Kolisch R (1995). Project Scheduling under Resource Constraints. Physica-Verlag, Heidelberg.

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 and Sprecher A (1997). PSPLIB-a project scheduling problem library: OR software-ORSEP operations research software exchange program. *European Journal of Operational Research* **96**(1):205-216.

Kolisch R and Padman R (2001). An integrated survey of deterministic project scheduling. *Omega* **29**(3):249-272.

Li KY and Willis RJ (1992). An iterative scheduling technique for resource-constrained project scheduling. *European Journal of Operational Research* **56**(3):370-379.

Liu H and 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,* pp. 22-27. IEEE.

Lova A and Tormos P (2002). Combining random sampling and backward-forward heuristics for resource-constrained multi-project scheduling. In *Proceedings of the Eighth International Workshop on PMS,* pp. 244-248.

Lova A, Maroto C and Tormos P (2000). A multi-criteria heuristic method to improve resource allocation in multi-project scheduling. *European Journal of Operational Research* **127**(2):408-424.

Myers RH, Montgomery DC and Anderson-Cook CM (2009). Response Surface Methodology: Process and Product Optimization Using Designed Experiments. John Wiley & Sons, New Jersey.

Najafi AA, Niaki STA and 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.

Özdamar L and 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.

Shahsavar A, Najafi AA and Niaki STA (2015). Three self-adaptive multi-objective evolutionary algorithms for a triple-objective project scheduling problem. *Computers and 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. PhD Dissertation, Boğaziçi University.

Smith-Daniels DE and Aquilano NJ (1987). Using a late-start resource-constrained project schedule to improve project net present value. *Decision Sciences* **18**(4):617-630.

Speranza MG and Vercellis C (1993). Hierarchical models for multi-project planning and scheduling. *European Journal of Operational Research* **64**(2):312–325.

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

Talbot FB (1982). Resource-constrained project scheduling with time-resource tradeoffs: The non-preemptive case. *Management Science* **28**(10):1197-1210.

Ulusoy G and Özdamar L (1995). A heuristic scheduling algorithm for improving the duration and net present value of a project. *International Journal of Operations and Production Management* **15**(1):89-98.

Ulusoy G, Sivrikaya-Şerifoğlu F and 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):237-261.

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

Viana A and de Sousa JP (2000). Using metaheuristics in multi-objective resource constrained project scheduling. *European Journal of Operational Research* **120**(2):359-374.

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

Węglarz J, Józefowska J, Mika M and 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.

Xu J and Zhang Z (2012). 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. *Journal of Scheduling* **15**(2):253-272.

Xu J and 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* **2014**: 20p. http://dx.doi.org/10.1155/2014/463692.

Zitzler E and Thiele L (1998). Multi-objective optimization using evolutionary algorithms—a comparative case study. In *Proceedings of International Conference on Parallel Problem Solving from Nature—PPSN V*, pp. 292-301.

Zitzler E (1999). Evolutionary algorithms for multi-objective optimization: Methods and applications. PhD Dissertation, Swiss Federal Institute of Technology (ETH), Zürich.