

Sameh Al-SHIHABI, Mohammad AIDURGAM

Multi-objective optimization for the multi-mode finance-based project scheduling problem

© Higher Education Press 2020

Abstract The finance-based scheduling problem (FBSP) is about scheduling project activities without exceeding a credit line financing limit. The FBSP is extended to consider different execution modes that result in the multi-mode FBSP (MMFBSP). Unfortunately, researchers have abandoned the development of exact models to solve the FBSP and its extensions. Instead, researchers have heavily relied on the use of heuristics and meta-heuristics, which do not guarantee solution optimality. No exact models are available for contractors who look for optimal solutions to the multi-objective MMFBSP. CPLEX, which is an exact solver, has witnessed a significant decrease in its computation time. Moreover, its current version, CPLEX 12.9, solves multi-objective optimization problems. This study presents a mixed-integer linear programming model for the multi-objective MMFBSP. Using CPLEX 12.9, we discuss several techniques that researchers can use to optimize a multi-objective MMFBSP. We test our model by solving several problems from the literature. We also show how to solve multi-objective optimization problems by using CPLEX 12.9 and how computation time increases as problem size increases. The small increase in computation time compared with possible cost savings make exact models a must for practitioners. Moreover, the linear programming-relaxation of the model, which takes seconds, can provide an excellent lower bound.

Keywords multi-objective optimization, finance-based scheduling, multi-mode project scheduling, mixed-integer linear programming, CPLEX

Received September 12, 2019; accepted December 17, 2019

Sameh Al-SHIHABI (✉)
Industrial Engineering Department, University of Jordan, Amman 11942, Jordan
E-mail: shihabi.sameh@spjain.org

Mohammad AIDURGAM
Systems Engineering Department, King Fahd University of Petroleum & Minerals, Dhahran 31261, Kingdom of Saudi Arabia

1 Introduction

The finance-based scheduling problem (FBSP) is introduced by Elazouni and Gab-Allah (2004). The FBSP captures the problem in which contractors rely on a credit line (CL) to finance project expenditures if project expenses exceed sponsor payments. Elazouni and Gab-Allah (2004) developed an integer programming (IP) model to minimize the project duration, such that the amount borrowed does not exceed the CL.

This pioneering work has attracted the attention of practitioners and researchers alike; consequently, cash-flow dynamics are modified to capture real-world situations. For example, expenses related to mobilization, advanced payments, and fixed costs are not considered by Elazouni and Gab-Allah (2004), which is important in practice. However, these costs are modeled by Elazouni and Metwally (2005). Note that the IP solution of Elazouni and Gab-Allah (2004) is not extended by Elazouni and Metwally (2005); to account for these changes, a Genetic Algorithm (GA) is developed for solving the modified problem.

Researchers abandoning exact methods and adopting heuristics and meta-heuristics to solve all FBSP variants is a norm. Table 1 summarizes the FBSP-related publications. As shown in column 2 of Table 1, researchers have modeled a single project or a portfolio of projects, whereas column 3 presents the targeted objective function. Solution methodologies used to solve the suggested models are shown in column 4. Clearly, a discontinuation exists in the developments of exact IP models because either heuristics or meta-heuristics techniques are used to solve the FBSP variants, except for the study of Elazouni and Gab-Allah (2004) and constraint programming (CP) of Liu and Wang (2008; 2010).

The CP models of Liu and Wang (2008; 2010) have ignored mobilization costs and advanced payments, as Elazouni and Gab-Allah (2004). Moreover, these models do not account for the case of a positive cash balance for which paying interests is no longer needed.

Table 1 Summary of FBSP-related publications

Paper	Single/Portfolio	Objective	Solution methodology
Elazouni and Gab-Allah (2004)	Single	Min. <i>duration</i>	IP
Elazouni and Metwally (2005)	Single	Min. <i>duration</i>	GA
Liu and Wang (2008)	Single	Max. <i>profit</i>	CP
Abido and Elazouni (2009)	Single	Min. <i>duration</i>	GA
Elazouni (2009)	Portfolio	Multiple	Heuristic
Afshar and Fathi (2009)	Single	Multiple	GA & Fuzzy Sets
Fathi and Afshar (2010)	Single	Multiple	GA
Liu and Wang (2010)	Portfolio	Max. <i>profit</i>	CP
Abido and Elazouni (2011)	Portfolio	Multiple	Heuristic
Elazouni and Abido (2011)	Single	Multiple	Heuristic
Alghazi et al. (2012)	Single	Min. <i>duration</i>	Frog-leaping Algorithm, GA, Simulated Annealing
Alghazi et al. (2013)	Single	Min. <i>duration</i>	GA
Elazouni et al. (2015)	Single	Min. <i>duration</i>	Frog-leaping Algorithm, GA, Simulated Annealing
Gajpal and Elazouni (2015)	Single	Min. <i>duration</i>	Heuristic
Al-Shihabi and AlDurgam (2017)	Single	Min. <i>duration</i>	Max-Min Ant System
Alavipour and Arditi (2018a)	Single	Min. <i>cost</i>	LP
Alavipour and Arditi (2018b)	Single	Min. <i>cost</i>	LP

Linear programming (LP) models are used to find the best financing alternatives for projects (Alavipour and Arditi, 2018a; 2018b). The LP models have assumed fixed schedules for which the optimum financing alternative is found.

Another relevant problem that has attracted many researchers and has an impact on the project cash-flow is the time-cost trade-off problem (TCTP) (Siemens, 1971; Kerzner and Kerzner, 2017). TCTP models allow activities to have different durations and costs. Thus, activities can accelerate, crash, and finish projects early (Laptali et al., 1997; Nkasu and Leung, 1997). TCTP models attempt to either minimize the project duration given a maximum project budget, which is also called “budget problem”, or to minimize the project budget given a maximum project duration, which is referred to as “deadline problem” (Yang, 2007).

The TCTP that assumes a linear relationship between time and cost can be solved using LP (Perera, 1980). However, the relationship between time and cost can be considered discrete, leading to the discrete TCTP (DTCTP) or multimodal TCTP (MMTCTP), that is, strongly NP (non-deterministic polynomial)-hard (De et al., 1995; 1997). Therefore, similar to any other NP-hard optimization problem, researchers have attempted to use heuristics (Vanhoucke and Debels, 2007), meta-heuristics (Sonmez and Bettemir, 2012), and exact methods (De et al., 1995; Burns et al., 1996; Moussourakis and Haksever, 2004; Chassiakos and Sakellaropoulos, 2005) to solve this problem. For bi- and multi-objective optimization problems, researchers have depended on meta-heuristics and heuristics (Zheng et al., 2004; Doerner et al., 2008;

Ammar, 2011).

Considering that crashing activities change project cash flows with respect to their amounts and timings, contractors must validate their TCTP solutions concerning the CL. Therefore, researchers have simultaneously considered the MMTCTP and FBSP. Moreover, a new problem is created that we henceforth call it the multi-mode FBSP (MMFBSP). Table 2 summarizes the literature related to the MMFBSP, where, again, exact solution techniques are missing. To the authors’ knowledge, no exact model captures the multi-objective MMFBSP. Alavipour and Arditi (2019a; 2019b) consider different schedules, for which they try to find the best financial alternatives. The authors use a GA and LP hybrid to solve their model: The GA schedules the project activities, whereas the LP algorithm finds the best financing options.

The accuracy of meta-heuristic or heuristic approaches, against exact methods, is not addressed in the literature on FBSP. Heuristic approaches are usually used to obtain quick solutions. Yet, a small deviation in a megaproject cost can mean millions of dollars. Consequently, spending extra time to guarantee the solution optimality is not an option for contractors — it is a must. Moreover, even if an exact algorithm is not run to optimality, it can still provide proper bounds, against which can benchmark heuristic and meta-heuristic solutions. With the new advances of commercial solvers, such as CPLEX 12.9 (Nodet, 2019), multi-objective optimization problems can be solved. In summary, exact models and solutions are needed for two reasons:

- Finding solutions with certain optimality;
- Finding bounds against which we can check the

Table 2 Summary of MMFBSP-related publications

Paper	Single/Portfolio	Objective	Solution methodology
Elazouni and Metwally (2007)	Single	Min. <i>duration</i> Max. <i>profit</i>	GA
Ali and Elazouni (2009)	Portfolio	Min. <i>duration</i> Max. <i>profit</i>	GA
Elazouni and Abido (2014)	Single	Multi-objective	Heuristic
El-Abbasy et al. (2016)	Portfolio	Multi-objective	GA
El-Abbasy et al. (2017)	Portfolio	Multi-objective	GA
Alavipour and Arditi (2019b)	Single	Max. <i>profit</i>	GA & LP
Alavipour and Arditi (2019a)	Single	Max. <i>profit</i>	GA & LP

quality of solutions obtained using meta-heuristic methods.

Heuristics are informed search techniques that systematically explore state space according to Pearl (1984), whereas meta-heuristics, such as GA, are high-level strategies for guiding searches, as discussed by Glover (1986). GA, which is the most used meta-heuristic to solve FBSP and MMFBSP, needs infinite time to converge to global optimum solutions, as proven by Rudolph (1994). In all the publications that are listed in Tables 1 and 2, a small number of instances are solved. Furthermore, according to the famous Free Lunch Theorem (Wolpert and Macready, 1997), concluding the accuracy of heuristic or meta-heuristic algorithms by using small sets of problems is dangerous.

In this study, we introduce an exact multi-objective model for the MMFBSP. We also demonstrate how a state-of-the-art exact solver similar to CPLEX 12.9 can be used to solve multi-objective optimization problems.

The mathematical model that captures the MMFBSP is a mixed integer linear program (MILP). The model attempts to minimize project duration, and we solve this model by using CPLEX 12.9, embedding standard branch and bound, and cutting plane algorithms. To avoid the development of a nonlinear model, we adopt an iterative solution technique that is presented by Elazouni and Gab-Allah (2004).

As stated by Nodet (2019), “with CPLEX 12.9, you can define multiple objectives for a single problem and combine them as a hierarchy, through blending, or with combinations of both.” Thus, CPLEX 12.9 can directly solve multi-objective optimization problems. If two or more objectives use the same measure, then these measures can be blended; otherwise, users can specify a hierarchy that shows the importance of the objectives. For contractors who have vague ideas about which objectives are more important than the other, CPLEX 12.9 can find alternative solutions using the `populate()` method.

This work has several contributions. The multi-objective MILP model fills the gap left by researchers concerning the modeling and solving of the FBSP and MMFBSP by using the exact methods. We also show how CPLEX can be used to solve multi-objective optimization problems. Unlike previous models that only consider either execution or

bidding stages of contracts, our model relates bidding to executing.

This study is organized as follows: Section 2 describes the business model of the MMFBSP, whereas Section 3 translates the business model to an MILP model. Section 4 demonstrates how CPLEX 12.9 can be used to solve multi-objective optimization problems. Section 5 presents several experiments to validate our model, show how to solve multi-objective problems by using CPLEX 12.9, and study the time complexity of problems. Section 6 discusses conclusions and future studies.

2 Business model

We divide the project life into two stages. The first stage is the bidding stage, and we show how a contractor calculates the bidding price (*BP*). The second stage is the executing stage, and we demonstrate how the contractor executes the project, given the contract and bank agreements.

2.1 Bidding

To find *BP*, the contractor must calculate activity and project duration costs. Each activity $i \in N$ can be executed based on mode $j \in M_i$, where M_i is the set of execution modes of activity i , and the contractor should choose only one of these modes to execute an activity. The execution modes are different in terms of time and cost. The execution time and cost of activity i , based on mode j is denoted by $T_{i,j}$ and $C_{i,j}$, respectively. In the following description, we use week to measure time. However, all the derivations can be adjusted to other time units. The time index used to denote the time units is k .

To find *BP*, the contractor selects one execution mode for each activity and finds the sum of costs of these modes, DC^{sum} , as shown in Eq. (1). Note that j^{bid} represents the selected mode in calculating *BP*. From this cost, the contractor calculates the total variable costs (VC^{sum}), as presented in Eq. (2), where O_V is the overhead multiplier.

$$DC^{\text{sum}} = \sum_{i \in N} C_{i,j^{\text{bid}}}, \quad (1)$$

$$VC^{\text{sum}} = O_V \times DC^{\text{sum}}. \quad (2)$$

Unlike the activities' direct and overhead costs, the contractor must pay a weekly fixed overhead fee, O_F , while executing the project. We use W^{bid} to denote the bidding duration; thus, the sum of the fixed overhead fees (FC^{sum}) can be found, as shown in Eq. (3). W^{bid} is usually found using the critical path method (CPM) (Horowitz, 1967). The contractor should also pay the mobilization cost (MOB) at the start of the project to move to the project location.

$$FC^{\text{sum}} = O_F \times W^{\text{bid}}. \quad (3)$$

The contractor usually looks for a profitability percentage, O_P , with respect to the above-discussed costs. To find the profit value (PV), the contractor multiplies O_P by all previously mentioned costs, as shown in Eq. (4) (Clough et al., 2015).

$$PV = O_P \times (DC^{\text{sum}} + VC^{\text{sum}} + FC^{\text{sum}} + MOB). \quad (4)$$

The contractor must pay a bonding cost, BC , as a form of guarantee to the sponsor if the contractor fails to fulfil the project clauses. This cost is usually 1%–2% of the sum of PV and all costs. The bonding cost multiplier is denoted by O_B , and the bonding value is as shown in Eq. (5). BP is the sum of all previous costs and profit value, as presented in Eq. (6).

$$BC = O_B \times (PV + DC^{\text{sum}} + VC^{\text{sum}} + FC^{\text{sum}} + MOB), \quad (5)$$

$$BP = BC + PV + DC^{\text{sum}} + VC^{\text{sum}} + FC^{\text{sum}} + MOB. \quad (6)$$

2.2 Executing

If the contractor wins the bid, then BP becomes the contract price (CP), and W^{bid} becomes W^{execute} . To finance the project, the contractor considers two financing sources: Payments from the sponsor and loans from a line of credit (LOC) account. We start this section by describing contractor expenses, followed by a description of sponsor payments. We then show how withdrawals from an LOC are used to finance the contractor's financial deficit.

2.2.1 Expenses

Before the start of the project, the sponsor should pay the mobilization and bonding costs, as shown in case 1 of Eq. (7). Expenses due to activity i in week $k \geq 1$ if executed according to mode j is $\frac{C_{ij}}{T_{ij}}$. The total expenditure due to all activities for $k \geq 1$, TE_k , is shown in case 2 of Eq. (7). TE_k for $k \geq 1$ includes activities' direct and indirect costs and the weekly fixed cost. All these costs are

related to the project. As we discuss later, we also have financing costs, which are the interests that must be paid to the bank.

$$TE_k = \begin{cases} MOB + BC & (k = 0), \\ (1 + O_V) \times \sum_{i \in N} \frac{C_{ij}}{T_{ij}} + O_F & \\ (k = 1, 2, \dots, W^{\text{execute}}). \end{cases} \quad (7)$$

2.2.2 Sponsor payments

The contractor and sponsor calculate a mark-up multiplier (MU), which relates CP to DC^{sum} , as shown in Eq. (8). MU is used by the contractor and sponsor to value the conducted work. Note that DC^{sum} depends on the selected modes during the bidding stage. However, these modes may change when executing the project, as discussed later.

$$MU = \frac{CP}{DC^{\text{sum}}}. \quad (8)$$

A reimbursement period (R), which is also called the project period, is part of the contract clauses. The contractor is only allowed to invoice the sponsor at weeks corresponding to $R, 2R, \dots, \left\lceil \frac{W^{\text{execute}}}{R} \right\rceil R$, where W^{execute} is the project execution duration. The invoices cover the project earned value (EV) during R .

To find EV due to activity $i \in N$ if part of it is executed in week k ($EV_{i,k}$), the contractor must map the completed work direct cost to the bidding cost, as presented in Eq. (9). Such an equation shows that the bidding cost is equally divided by the execution mode duration. The sum of all costs during R , due to all activities, represents the project period earned value (PEV), which the contractor uses to invoice the sponsor, as shown in Eq. (10). Equation (10) finds the sum of the EV of all the activities from the start of the period at $R + 1$ weeks earlier than the invoice date to the invoice date.

$$EV_{i,k} = \frac{C_{ij}^{\text{bid}}}{T_{ij}} \left(k = 1, 2, \dots, \left\lceil \frac{W^{\text{execute}}}{R} \right\rceil \right), \quad (9)$$

$$PEV_k = \begin{cases} \sum_{t=k-R+1}^k \sum_{i \in N} EV_{i,t} \\ \left(k = R, 2R, \dots, \left\lceil \frac{W^{\text{execute}}}{R} \right\rceil R \right), \\ 0, \text{ otherwise.} \end{cases} \quad (10)$$

To invoice the sponsor in week k , I_k , the contractor marks up PEV , as presented in Eq. (11). The sponsor does not immediately pay the contractor; rather, the sponsor delays the payment (P_k) for L_P weeks, as shown in case 2

of Eq. (12), where RP is the retainage percentage, i.e., amount kept by the sponsor to be paid at the end of the project. The second term of case 2 of Eq. (12) shows the deduction value of the advanced payment (AP), which the sponsor pays to the contractor at the start of the project, as presented in case 1 of Eq. (12). After the last invoice payment by L_R weeks, the contractor receives the retained amounts from the invoices, as shown in case 3 of Eq. (12).

$$I_k = \begin{cases} MU \times PEV_k \left(k = R, 2R, \dots, \left\lceil \frac{W^{\text{execute}}}{R} \right\rceil R \right), \\ 0, \text{ otherwise,} \end{cases} \quad (11)$$

$$P_k = \begin{cases} AP \ (k = 0), \\ (1 - RP) \times I_{k-L_P} - \frac{AP}{\left\lceil \frac{W^{\text{execute}}}{R} \right\rceil R}, \\ \left(k = R + L_P, 2R + L_P, \dots, \left\lceil \frac{W^{\text{execute}}}{R} \right\rceil R + L_P \right), \\ RP \times CP \left(k = \left\lceil \frac{W^{\text{execute}}}{R} \right\rceil R + L_P + L_R \right), \\ 0, \text{ otherwise.} \end{cases} \quad (12)$$

Due to L_P , the contractor's cumulative expenses are usually higher than his cumulative income as described by Peterson (2009) and shown in Fig. 1. The contractor depends on an LOC account to finance the project expenses.

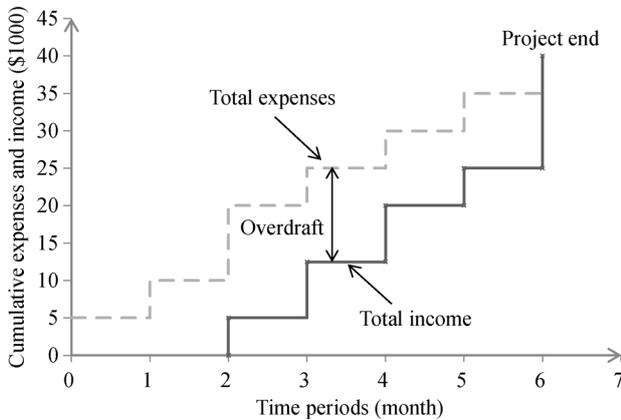


Fig. 1 Cumulative expenses versus income profile.

2.2.3 LOC

LOC is an arrangement between a financial institution,

usually a bank, and a customer who establishes the maximum loan amount the customer can borrow. The borrower can access funds from the LOC any time as long as the borrower does not exceed the maximum amount (or credit limit) set in the agreement and meet any other requirements, such as making timely minimum payments. The contractor negotiates with the bank about how loans are compounded and when interests are paid. Assuming that interests are compounded weekly, we denote the weekly interest as r_W . We also assume that interests are paid every V weeks. The bank withdrawals depend on the bank account balance, weekly expenses, and payments. The bank account balance (B_k) can be found in Eq. (13). Before the start of the project, the balance is simply the difference between AP and TE_0 , as shown in case 1 of Eq. (13). Starting from week one, $k = 1$, until the last invoice payment in week $\left\lceil \frac{W^{\text{execute}}}{R} \right\rceil R + L_P$, the bank balance is shown in case 2 of Eq. (13). This case shows that the balance in week k is equal to that in the previous week plus any payment received at the end of the previous week minus the current week expenses and current week interest payments (IB_k) to the bank if it is due. The balance at the end of the project, once the retained invoice payments are returned to the contractor, is shown in case 3 of Eq. (13). An LOC means that B_k should not exceed CL , as presented in Eq. (14).

$$B_k = \begin{cases} AP - TE_0 \ (k = 0), \\ B_{k-1} + P_{k-1} - TE_k - IB_k \\ \left(k = 1, 2, \dots, \left\lceil \frac{W^{\text{execute}}}{R} \right\rceil R + L_P \right), \\ B_{\left\lceil \frac{W^{\text{execute}}}{R} \right\rceil R + L_P} + RP \times CP \\ \left(k = \left\lceil \frac{W^{\text{execute}}}{R} \right\rceil R + L_P + L_R \right), \end{cases} \quad (13)$$

$$B_k \geq -CL \ (k = 1, 2, \dots, W^{\text{execute}}). \quad (14)$$

To find interest payments in week k , IB_k , we must consider the following three cases:

- $B_{k-1} \geq TE_k$. For this case, the cash balance left from the previous week can cover the current week expenditures.
- $B_{k-1} \leq 0$. For this case, the cash balance left from the previous week cannot cover any of the current week expenditures.
- $0 \leq B_{k-1} \leq TE_k$. For this case, the cash balance from the previous week is positive but fails to cover all the current week expenses. Consequently, the contractor must withdraw $TE_k - B_{k-1}$ for the LOC.

We assume that the contractor pays all the current week expenses at the beginning of the week. Thus, interests due

on week $k = V, 2V, \dots, \left\lceil \frac{W^{\text{execute}}}{V} \right\rceil V$ are calculated, as shown in Eq. (15).

$$IB_k = \begin{cases} \sum_{t=k-V+1}^k -\min\{0, B_{k-1} - TE_k\} \times r_w \\ \left(k = V, 2V, \dots, \left\lceil \frac{W^{\text{execute}}}{V} \right\rceil V \right), \\ 0, \text{ otherwise.} \end{cases} \quad (15)$$

3 Optimization model

The MILP model checks the feasibility of finishing the project in W^{execute} weeks. We assume a value for project duration to avoid the formulation of a mixed-integer nonlinear programming (MINLP) model, as discussed later in this section.

3.1 Formulation

The objective function of this formulation is the minimization of the project duration. In Eq. (16), π is a milestone event, which has zero duration that signals the end of the project. The basic decision variable of the model is $x_{i,j,k}$, which is a binary decision variable that has a value of 1 if activity $i \in N$ starts in week k and is executed according to mode $j \in M_i$, 0 otherwise. Only one $x_{i,j,k}$ is equal to 1 per activity, as shown in Eq. (17). That is, an activity must be executed on the basis of one mode and start in one week.

The project ends once all activities finish, as shown in Eq. (18). We also impose that activities without successors finish in W^{execute} weeks, as presented in Eq. (19), where Q_i is the set of successor activities to activity i . This condition is needed to determine the number of invoices or payments, from which we calculate the deduction values that should be subtracted from sponsor payments, as shown in case 2 of Eq. (12) of the business model and case 2 of Eq. (25) of the MILP model. If this condition is not added to the MILP formulation, then the model becomes

MINLP due to the $\frac{AP}{\left\lceil \frac{W}{R} \right\rceil}$ term. For the FBSP, researchers

have ignored AP in their formulations (Elazouni and Gab-Allah, 2004; Elazouni et al., 2015). Precedence relationships are shown in Eq. (20).

The two cases of total expenditures that are discussed in Eq. (7) of the business model are captured by Eq. (21) of the MILP model. To find expenditures in week k , we must consider the possibilities that the activity start in week k or earlier. For example, assume that activity A has two modes, 1 and 2, having a duration of $T_{A,1} = 4$ and $T_{A,2} = 5$, respectively. If we want to calculate the expenditures due

to this activity in week 4, then we must consider all the possibilities to start this activity between $k = 1$ and $k = 4$ based on modes 1 and 2. The summation $\sum_{t=\max(1, k-T_{ij})}^k$ captures these possible starts. Note that if $k - T_{ij} \leq 0$, then the summation starts at $k = 1$.

Equation (22) shows how to calculate $EV_{i,k}$ in terms of decision variables $x_{i,j,k}$. The earned value, according to Eq. (22), is estimated in terms of the $C_{i,j,bid}$ of the activities. To invoice the sponsor, PEV_k , where $k = R, 2R, \dots, \left\lceil \frac{W^{\text{execute}}}{R} \right\rceil R$, is found using Eq. (23). The invoice values are shown in Eq. (24), the payments from the sponsor are presented in Eq. (25), and bank balance is shown in Eq. (26).

In the business model, interests are paid if $B_{k-1} - TE_k < 0$, as exhibited in Eq. (15). The first step in calculating interests is to check if the cash balance of the previous week minus the current week expenditures is positive or negative. For this reason, we define binary indicator variable α_k , such that this variable is equal to 1 if the cash balance of the previous week minus the current week expenditures is negative, 0 otherwise. The value of α_k is found using Eqs. (27) and (28). We use auxiliary variable IL_k to obtain a value of 0 if $B_{k-1} - TE_k > 0$, otherwise $IL_k = B_{k-1} - TE_k$, as shown in Eqs. (29–32). M is a large number in Eqs. (27) and (28), in addition to Eqs. (30–32). The value of IB_k , which is the interest due to the bank, is found using Eq. (33). Equation (34) enforces B_k not to exceed CL , similar to Eq. (14) in the business model.

$$\text{Min } \pi \quad (16)$$

s.t.

$$\sum_{k=1}^W \sum_{j \in M_i} x_{i,j,k} = 1 \quad (\forall i \in N), \quad (17)$$

$$\sum_{k=1}^W \sum_{j \in M_i} (k + T_{ij}) \times x_{i,j,k} \leq \pi \quad (\forall i \in N), \quad (18)$$

$$\sum_{k=1}^W \sum_{i \in N} \sum_{j \in M_i} (k + T_{ij}) \times x_{i,j,k} \geq 1 \quad (\forall \{i \in N | Q_i = \phi \text{ and } k + T_{ij} = W^{\text{execute}}\}), \quad (19)$$

$$\sum_{k=1}^W \sum_{j \in M_i} (k + T_{ij}) \times x_{i,j,k} \leq \sum_{k=1}^W \sum_{j \in M_q} k \times x_{q,j,k} \quad (\forall \{i, q \in N | q \in Q_i\}), \quad (20)$$

$$TE_k = \begin{cases} MOB + BC \quad (k = 0), \\ \sum_{i \in N} \sum_{j \in M_i} \sum_{t=\max(1, k-T_{ij})}^k \frac{C_{i,j}}{T_{ij}} \times x_{i,j,t} \\ (\forall i \in N \text{ and } k = 1, 2, \dots, W^{\text{execute}}), \end{cases} \quad (21)$$

$$EV_{i,k} = \sum_{t=\max(1,k-T_{ij})}^k \sum_{j \in M_i} \frac{C_{ij}^{\text{bid}}}{T_{ij}} \times x_{ij,t} \quad (k = 1, 2, \dots, \left\lceil \frac{W^{\text{execute}}}{R} \right\rceil), \quad (22)$$

$$PEV_k = \begin{cases} \sum_{t=k-R+1}^k \sum_{i \in N} EV_{i,k} \\ \left(k = R, 2R, \dots, \left\lceil \frac{W^{\text{execute}}}{R} \right\rceil R \right), \\ 0, \text{ otherwise,} \end{cases} \quad (23)$$

$$I_k = \begin{cases} MU \times PEV_k \left(k = R, 2R, \dots, \left\lceil \frac{W^{\text{execute}}}{R} \right\rceil R \right), \\ 0, \text{ otherwise,} \end{cases} \quad (24)$$

$$P_k = \begin{cases} AP \ (k = 0), \\ (1-RP) \times I_{k-L_P} - \frac{AP}{\left\lceil \frac{W^{\text{execute}}}{R} \right\rceil} \\ \left(k = R + L_P, 2R + L_P, \dots, \left\lceil \frac{W^{\text{execute}}}{R} \right\rceil R + L_P \right), \\ RP \times CP \ \left(k = \left\lceil \frac{W^{\text{execute}}}{R} \right\rceil R + L_P + L_R \right), \\ 0, \text{ otherwise,} \end{cases} \quad (25)$$

$$B_k = \begin{cases} AP - TE_0 \ (k = 0), \\ B_{k-1} + P_{k-1} - TE_k - IB_k \\ \left(k = 1, 2, \dots, \left\lceil \frac{W^{\text{execute}}}{R} \right\rceil R + L_P \right), \\ B \frac{W^{\text{execute}}}{R} + RP \times CP \\ \left(k = \left\lceil \frac{W^{\text{execute}}}{R} \right\rceil R + L_P + L_R \right), \end{cases} \quad (26)$$

$$B_{k-1} - TE_k \leq (1 - \alpha_k) \times M \ (k = 1, 2, \dots, W^{\text{execute}}), \quad (27)$$

$$B_{k-1} - TE_k \geq -\alpha_k \times M \ (k = 1, 2, \dots, W^{\text{execute}}), \quad (28)$$

$$IL_k \geq 0 \ (k = 1, 2, \dots, W^{\text{execute}}), \quad (29)$$

$$IL_k \geq -(B_{k-1} - TE_k) + (\alpha_k - 1) \times M \quad (k = 1, 2, \dots, W^{\text{execute}}), \quad (30)$$

$$IL_k \leq -(B_{k-1} - TE_k) + (1 - \alpha_k) \times M \quad (k = 1, 2, \dots, W^{\text{execute}}), \quad (31)$$

$$IL_k \leq \alpha_k \times M \ (k = 1, 2, \dots, W^{\text{execute}}), \quad (32)$$

$$IB_k = \begin{cases} \sum_{t=\max(1,k-V+1)}^k IL_t \times r_W \\ \left(k = V, 2V, \dots, \left\lceil \frac{W^{\text{execute}}}{V} \right\rceil V \right), \\ 0, \text{ otherwise,} \end{cases} \quad (33)$$

$$B_k \geq -CL \ (k = 1, 2, \dots, W^{\text{execute}}), \quad (34)$$

$$\alpha_k \in \{0, 1\}. \quad (35)$$

3.2 Algorithm

Given that the model assumes a value for W^{execute} , the MILP model must be solved several times by starting with $W^{\text{execute}} = W^{\text{min}}$. If the model fails to find a feasible solution for this project duration value, then we increment W^{execute} by one week to become $W^{\text{execute}} = W^{\text{min}} + 1$ and try again to determine if the model finds a feasible solution. This process is repeated until a feasible solution is found. The contractor can specify a bound for the possible project duration extension, which we denote as J , as shown in Fig. 2. If the algorithm fails to find a feasible solution in $W^{\text{execute}} + J$ weeks, then the problem is considered infeasible.

This iterative solution algorithm is first suggested by Elazouni and Gab-Allah (2004). Their starting solution is the CPM solution of the problem, assuming that no credit limit constraint exists. Following the same steps of Elazouni and Gab-Allah (2004), we find that the number of iterations increases for tight CL, and the time taken to check infeasible solutions drastically increases for large instances. To minimize time, we also start with the CPM solution; however, we use the LP-relaxation of the

problem, which considers CL, and iteratively solve the problem until we have a feasible solution of the LP-relaxation problem, as shown by the first loop of Fig. 2. Starting with this solution, we switch to the MILP model, as presented by the second loop of Fig. 2.

For the value of J , heuristic or meta-heuristics can provide good upper bound for $W^{execute}$, for example, the GA of El-Abbasy et al. (2016; 2017).

3.3 Interest calculation

The interest calculations of Elazouni and Gab-Allah (2004) use $R = L_P = V$. Having $R = L_P = V$ can make the cash balance monotonically decreasing, except for weeks when the sponsor pays an invoice, that is, every R weeks. Figure 3 illustrates a hypothetical situation in which $R = L_P = V = 4$. According to Elazouni and Gab-Allah (2004), interests that should be paid to the bank in week $k = V, 2V, \dots, \left\lceil \frac{W}{V} \right\rceil V$ are due to the negative cash balance of the previous interest payment period, IB_k^{prev} , and the compounded interests due to weekly loans, IB_k^{week} , as shown in Eq. (36). Region I in Fig. 3 represents IB_k^{prev} , whereas regions II, III, IV, and V represent IB_k^{week} , for weeks $k = 1, 2, 3, 4$, respectively.

$$IB_k = IB_k^{prev} + IB_k^{week} \left(k = V, 2V, \dots, \left\lceil \frac{W}{V} \right\rceil V \right). \quad (36)$$

Based on Elazouni and Gab-Allah (2004)'s model,

IB_k^{prev} and IB_k^{week} are calculated according to Eqs. (37) and (38), respectively. The interest calculations of our model do not have any restriction regarding the values of R, L_P , and V . For instance, our model can handle the case of $R \neq L_P$, as shown in the hypothetical case illustrated in Fig. 4.

$$IB_k^{prev} = \begin{cases} B_{k-V} \times \left((1 + r_W)^V - 1 \right) \\ \left(k = V, 2V, \dots, \left\lceil \frac{W}{V} \right\rceil V \right), \\ 0, \text{ otherwise,} \end{cases} \quad (37)$$

$$IB_k^{week} = \sum_{t=k-V+1}^k \max\{TE_t - \min\{B_{k-1} - TE_k\}\} \times (1 + r_W)^{(k-t+1)} - 1 \quad (k = V, 2V, \dots, W). \quad (38)$$

4 Multi-objective optimization

In addition to project duration, contractors may also consider other objectives, such as minimum CL, maximum profit, or minimum interest cost. In this section, we discuss two approaches to use CPLEX 12.9 in solving multi-objective optimization problems. The first approach is to use the new multi-objective optimization features of CPLEX 12.9 (Nodet, 2019). The second approach is to

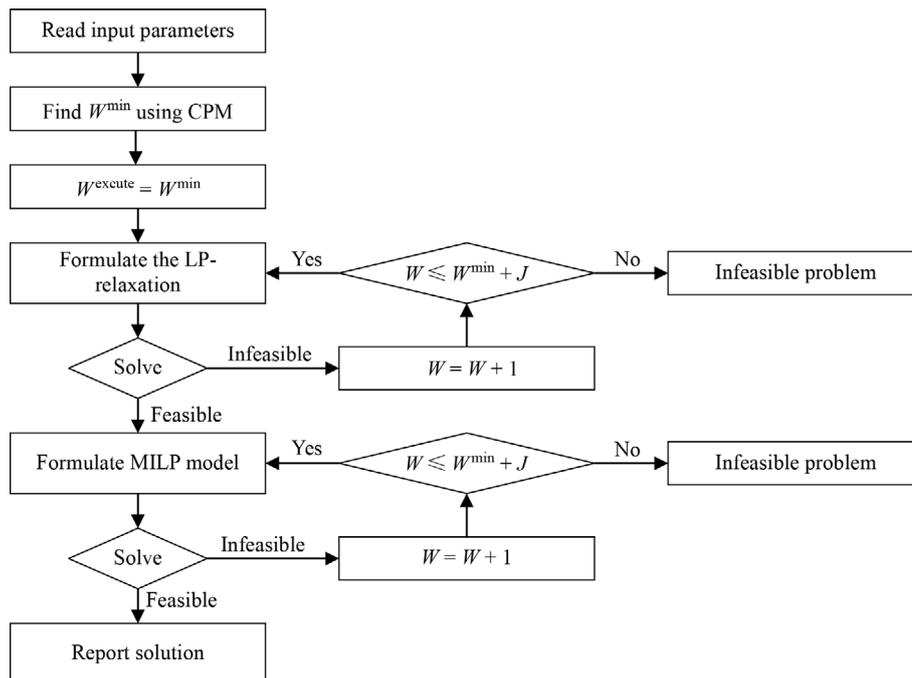


Fig. 2 Iterative algorithm to minimize project duration.

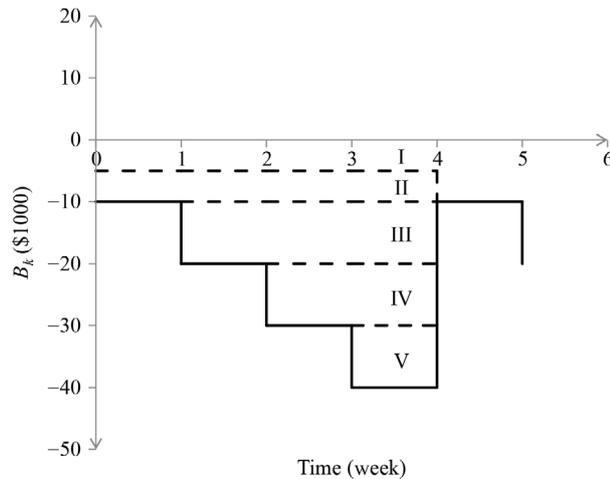


Fig. 3 Hypothetical cash balance for $R = L_p = V = 4$.

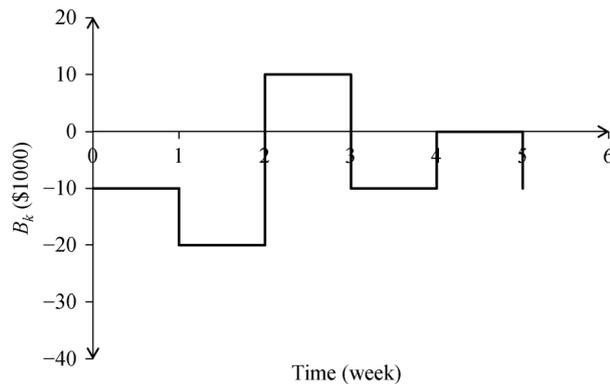


Fig. 4 Hypothetical cash balance for $R \neq L_p$.

use the `populate()` function from CPLEX to generate several solutions and choose the solution that maximizes or minimizes other objectives.

4.1 Multi-objective optimization using CPLEX 12.9

The newest version of CPLEX, CPLEX 12.9, can deal with multi-objective optimization. To use this module, the user must define two parameters.

- Priorities that define the order in which objectives are sequenced;
- Tolerances that define the range over which objective function value may be degraded to accommodate for improved optimal values of low-priority objectives.

For example, a contractor can have duration as the main priority, followed by CL, then interest cost. The priorities of these three objectives can be entered as 0, -1 and -2. Moreover, if the contractor allows a two-week delay in executing the project to have low CL and interest cost, then the duration tolerance is two weeks.

4.2 Multi-objective optimization using `populate()`

Contractors may not have a clear idea about which combination of objectives is good for them and what is the priority of these objectives. For this reason, contractors may look for a set of Pareto optimal solutions (Censor, 1977) that can be found using the `populate()` method of CPLEX 12.9. This method finds alternative solutions having the same objective value. Users can specify how many solutions they want CPLEX 12.9 to find.

Figure 5 summarizes how to solve the MMFBSP by using CPLEX 12.9 and how to look for alternative solutions with the same duration but different values of profitability or financing needs. The algorithm extends the one presented in Fig. 2, which only looks for a solution to the project time minimization. As illustrated in Fig. 5, once we find a feasible solution to the project duration problem, the `populate()` method is employed to find alternative solutions having different characteristics. The `populate()` method is left for the user to compare the different solutions and select the best. After identifying the feasible solution with the shortest duration, the `populate()` method can be used with different W values to generate solutions having different durations. Users of the `populate()` method must specify the number of needed solutions.

5 Experiments

Three sets of experiments are conducted. In the first set, we use our MMFBSP to solve two problems that are previously solved using GA. In the second set, we show the capabilities of CPLEX 12.9 in solving multi-objective optimization problems. In the last experiment, we study how computation time increases as problem size increases. The processor used to conduct all experiments in our study is the 2.5 GHz Core(TM) i5-3210M.

5.1 Comparisons

In this experiment, we validate our MMFBSP model by solving FBSP and MMTCTP examples from the literature that are solved using GA. Intuitively and logically, meta-heuristic solutions are validated by comparing their results against exact models. However, and given that researchers have preferred meta-heuristic solutions and ignored the exact ones, we assume that meta-heuristic solutions against which we compare our results are optimal.

5.1.1 MMTCTP validation

This example is first presented by Moussourakis and Haksever (2004), where a mixed-integer programming (MIP) model is presented, and a single objective function is minimized. Ammar (2011) also used the same example

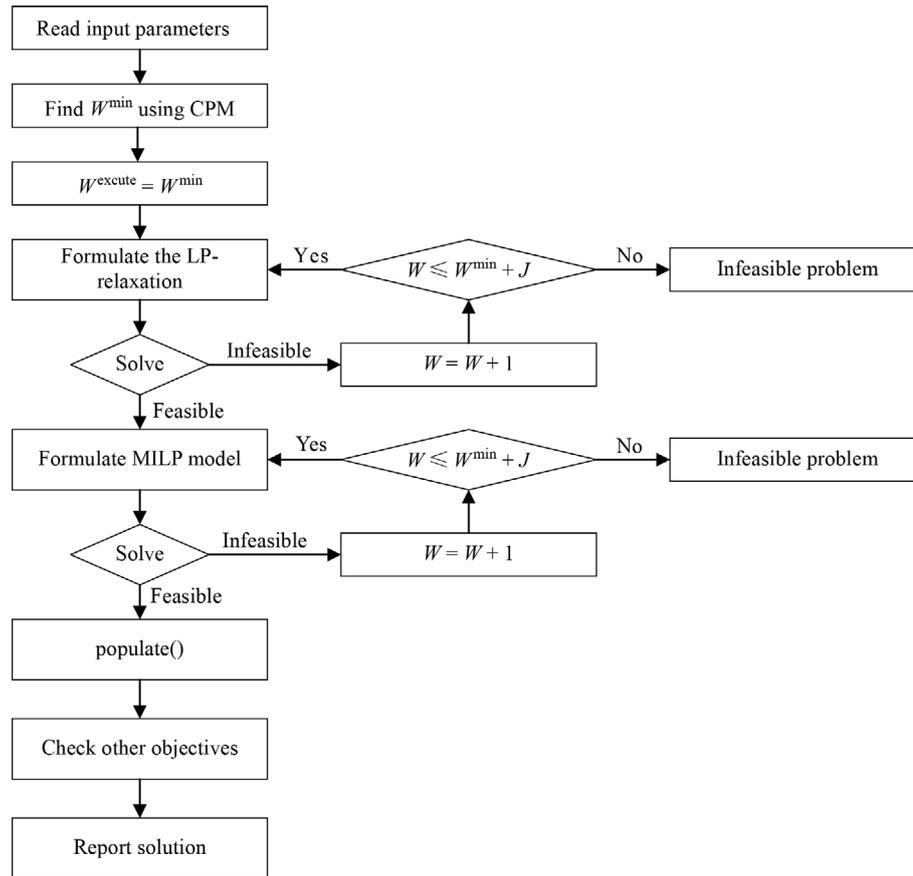


Fig. 5 Finding alternative solutions using the populate() method.

to solve a multi-objective optimization problem by using GA. Through this example, we show that our model can solve MMTCTP problems and how to use CPLEX 12.9 for solving multi-objective optimization problems easily. We attempt to replicate the results presented by Ammar (2011), where optimum time and cost combinations are found.

The example has seven activities, and each activity has different execution modes, as shown in Table 3. For instance, activity D can be executed in two modes: A discrete one in five weeks and based on a linear function between 12 and 18 weeks. Thus, activity D can be completed in 5, 12, 13, 14, 15, 16, 17, and 18 weeks, resulting in eight modes. The project network is represented in Fig. 6.

Ammar (2011) shows the optimal costs that correspond to different project durations. Execution times are not reported. For this example, $W^{\min} = 60$. The range of durations reported is between 60 and 76. Given that $CL = \infty$ and $B = \infty$, we can find a feasible solution for each duration.

In Table 4, we show the optimum solutions obtained using our model. Such solutions are the same as the ones reported by Ammar (2011). We also present the number of alternative solutions found for each project minimum duration and cost by using the populate() method in

column 3. The reason behind this number is due to the activities' slack times. According to Ammar (2011), the GA, or any other heuristic or meta-heuristic algorithm, cannot recognize all these alternative solutions.

5.1.2 FBSP validation

One of the examples that has been cited by many researchers (Alghazi et al., 2012; 2013) is the 30-activity model shown in Fig. 7. The project illustrated in the same figure consists of six main activities: A, B, C, D, E, and F. We obtain as many copies of these activities as shown in Fig. 7, where we have five copies of each activity. The precedence relationships are also illustrated in Fig. 7. The costs of activities A, B, C, D, E, and F are \$1700, \$1500, \$1800, \$1900, \$1600, and \$2000, respectively. Alghazi et al. (2013) solved this problem by assuming the data shown in Table 5, where APR is the annual percentage rate. In describing their example, the authors did not add variable overheads to each activity. Instead, they calculated the value of the total overheads and equally distributed that value over the project duration. Moreover, they paid taxes that are equally distributed over the project duration. Hence, in replicating the authors' model, we assume that $O_V = 0$ and use $O_F = \$789.3$.

Table 3 Contractor cost parameters of the MMTCTP example

Activity	Option	Duration (week)		Cost (\$)		Time-cost function	Number of modes
		Upper	Lower	Lower	Limit		
A	1	28	22	100	130	Piecewise linear	15
	2	22	18	140	210		
	3	28	22	100	130		
B	1	20	14	50	80	Linear with gaps	10
	2	10	8	120	300		
C	1	15	15	75	75	Discrete points	3
	2	8	8	240	240		
	2	4	4	500	500		
D	1	18	12	70	220	Linear and discrete points	8
	2	5	5	360	360		
E	1	26	18	40	80	Piecewise linear	22
	2	18	14	200	240		
	3	14	6	240	260		
F	1	25	15	120	300	Linear	11
G	1	7	7	0	0	Discrete points	11

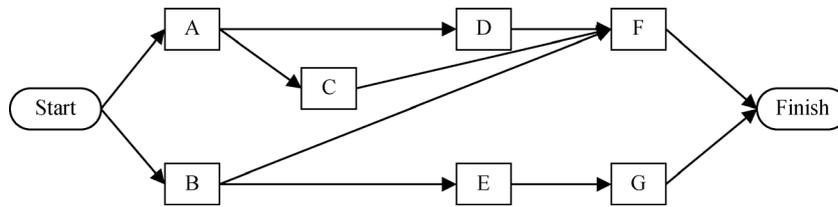


Fig. 6 Network representing the precedence relationships of the MMTCTP example.

Table 4 Solutions of the MMTCTP example

Duration (week)	Budget (\$)	Number of alternatives
76	911.0	22
75	910.0	22
74	909.0	23
73	908.0	25
72	907.0	25
71	906.0	906
70	905.0	21
69	904.0	21
68	903.0	21
67	902.0	21
66	901.0	21
65	900.0	20
64	917.0	21
63	933.0	21
62	944.5	22
61	956.0	23
60	973.0	24

After replicating the model, we can generate several solutions that have a slight difference in terms of the final profitability. The single solution presented by Alghazi et al. (2012; 2013) has a profitability of \$28013.4. However, we obtain several solutions that have resulted in different profitability ranging from \$27885.6 to \$28036.8. The difference between these values is due to the interest costs.

5.2 Multi-object optimization approaches

In this experiment, we demonstrate how to use the previously discussed two multi-objective optimization techniques. We use the 30-activity model used in the FBSP experiment, but we change the input parameters of this experiment. Table 6 shows the bidding, contracting, and financing parameters of this experiment, respectively, where r_D is the effective daily interest rate.

5.2.1 Unknown priorities

We use the `populate()` method to find 1000 solutions for

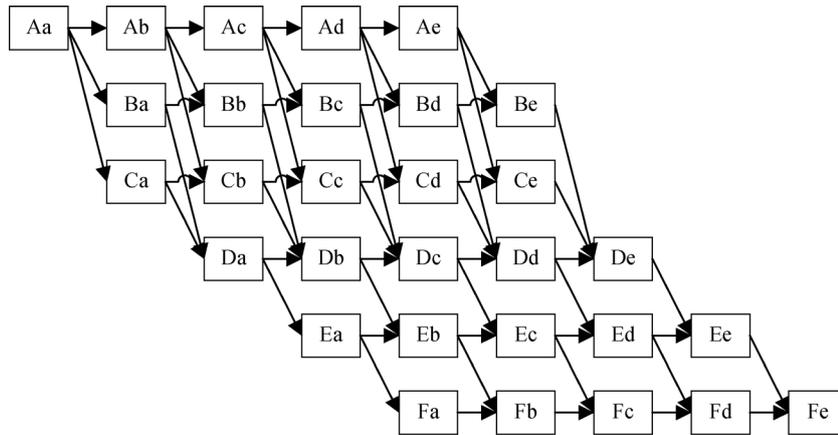


Fig. 7 Network representing the FBSP example.

Table 5 Contract and loan parameters of the FBSP example

Contract parameters		Loan parameters	
Data type	Value	Data type	Value
<i>BC</i>	\$1958	<i>APR</i>	58.4%
<i>MOB</i>	\$7618	<i>CL</i>	\$4000
<i>AP</i>	\$19779	<i>Comp</i>	daily
<i>L_P</i>	1 week	<i>V</i>	weekly
<i>L_R</i>	0 week	<i>r_w</i>	0.8%
<i>MU</i>	1.492		
<i>R</i>	1 week		
<i>RP</i>	5%		

different project durations. For $W = 34$ weeks, Fig. 8 shows the different solutions with respect to their *CL* and profit. Contractors can use this graph to define their optimal front, and then execute the project on the basis of their own criteria. This experiment is repeated for different project durations and is summarized in Table 7.

5.2.2 Known priorities

In this experiment, we assume that the contractor has preferences with respect to the objective functions. Using

$W = 34$ weeks, we first solve the problem, assuming that profitability is more important than *CL*. Then, we shift priorities by minimizing *CL* first, and then maximizing the profit. For the first case, the maximum profit is \$92405, whereas the minimum *CL* is \$29856. For the second case, the minimum *CL* is \$29481, whereas the maximum profit for this *CL* is \$91615. Figure 8 does not show these two solutions.

5.3 Computation time

In this experiment, we study the time needed to find feasible solutions for a set of instances having different sizes. As a test set, we extend the model used in the previous example by having many copies of each activity, as explained by Alghazi et al. (2012; 2013). For example, to create a 60-activity project, we generate 10 copies of each main activity. We generate four project networks having 30, 60, 120, and 240 activities. We also use the same bidding, contracting, and financing parameters shown in Table 6.

Approaches and times to find solutions are presented in Table 8. Column 1 of Table 8 shows the instance size, whereas column 2 presents the CPM solutions if the fastest execution modes are used. The time to find a CPM solution is 0 s for all the instances. Starting with the CPM solution,

Table 6 Default bidding, contracting, and financing parameters of the experimental study

Bidding parameters		Contracting parameters		Financing parameters	
Data type	Value	Data type	Value	Data type	Value
<i>O_F</i>	\$100/day	<i>AP</i>	$1.1 \times (MOB + BC)$	<i>CL</i>	\$20000
<i>MOB</i>	\$10000	<i>L_P</i>	4 weeks	<i>Comp</i>	daily
<i>O_B</i>	2%	<i>L_R</i>	0 week	<i>r_D</i>	1%
<i>O_P</i>	50%	<i>MU</i>	calculated	<i>V</i>	weekly
<i>O_V</i>	10%	<i>R</i>	4 weeks		
		<i>RP</i>	20%		

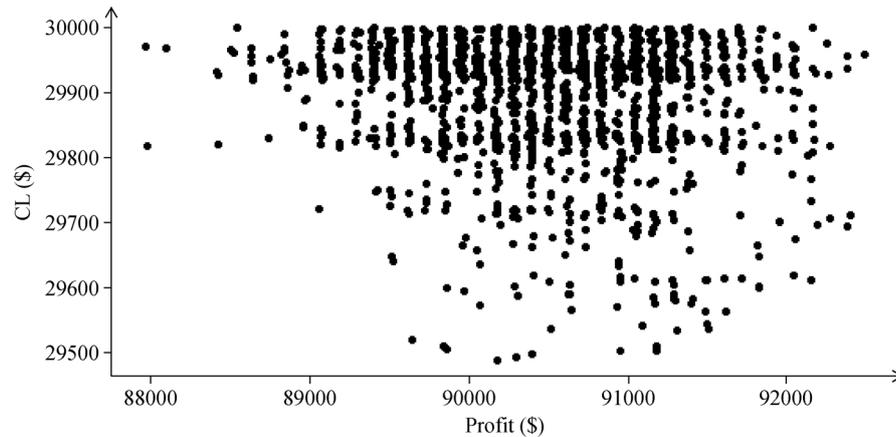


Fig. 8 Solutions having different CL and profits for the 30-activity problem.

Table 7 Experiment results of the 30-activity project example

Duration (week)	Min. CL (\$)	Max. Profit (\$)	Min. Interest (\$)
34	29377	92610	636
35	28907	92930	618
36	26930	93543	577
37	27125	92250	556
38	26173	92375	548
39	24104	93659	516
40	24624	93425	489

we apply the solution algorithm as illustrated in Fig. 2, but use the LP-relaxation of the problem. Columns 3 and 4 show the feasible LP-relaxation solutions and the time needed to find these solutions, respectively. The time to find the first feasible LP-relaxation solution is negligible, except for the 240-activity instance that takes 157 s, which is less than 3 min.

Starting with the LP-relaxation solution, we again apply our algorithm but use the MILP model. Columns 5 and 6 of Table 8 show the MILP solution and the time to find this solution, respectively. The computation time is less than 2 min for all the instances, except for the 240-activity instance, which takes nearly 41 min. In total, solving the 240-activity takes less than 45 min. Note that CPLEX 12.9 can do parallel computing to reduce the computation time.

Finally, the LP-relaxation has provided a good lower bound for the problem. Optimal MILP solutions take one week more than the optimal LP-relaxation solutions.

Table 8 Computation time experiment results for different project size

Instance size	CPM solution (week)	LP-relaxation solution (week)	LP-relaxation time (s)	MILP solution (week)	MILP solution time (s)
30	29	33	0	34	2
60	49	55	0	56	102
120	89	92	10	93	110
240	169	171	157	172	2433

Contractors can use the LP-relaxation to check solutions found by heuristics and meta-heuristics.

6 Conclusions and future research

This study presents an MILP model to solve the multi-objective MMFBSP. Using the bidding parameters, the model shows the best project execution schedule. To avoid having a nonlinear model, the MILP is embedded in an iterative algorithm that attempts to find a feasible project duration by iteratively increasing the project executing time. Given that project duration may not be the only criteria considered in scheduling the project, we demonstrate how CPLEX 12.9 can be used to solve multi-objective optimization problems. Unlike heuristic and meta-heuristic solutions, our model guarantees solution optimality.

Users who are not interested in solving several optimization models having different objectives can use two features of CPLEX 12.9. Users who are certain about what objectives they are targeting and the importance of these objectives can directly use this information in CPLEX 12.9. Users who have a vague idea about what objectives they are looking for or the importance of such objectives can find several alternative solutions using the `populate()` method.

We validate our model by solving the MMTCTP and FBSP examples from the literature. Then, we show how the `populate()` method can help in finding alternative solutions from which the user can select the best. Last, we

study how computation time increases as problem size increases. The time needed to solve a problem having three solution alternatives is less than 45 min, without benefiting from the parallel computing capabilities of CPLEX 12.9. Moreover, the LP-relaxation of the problem provides an excellent lower bound against which results of heuristic and meta-heuristic solutions can be compared. LP-relaxation time is almost negligible.

An improved model that maintains linearity but avoids the iterative solution technique is needed. Researchers have previously abandoned exact methods and adopted heuristic and meta-heuristics to solve the FBSP and its extension. Thus, additional exact models are also needed to capture all FBSP extensions, such as considering more than one financing option or portfolio of projects. Meta-heuristic solutions must be tested against large, real project instances to check their accuracy. Doing so is significant for contractors to judge if spending extra time for an optimal solution is a good investment.

Acknowledgements We would like to thank IBM Company for making CPLEX free for research. The second author would like to acknowledge the support provided by the Systems Engineering Department, King Fahd University of Petroleum & Minerals, Dhahran-KSA, to conduct this research.

References

- Abido M, Elazouni A M (2011). Multi-objective evolutionary finance based scheduling: Entire projects' portfolio. *Journal of Computing in Civil Engineering*, 25(1): 85–97
- Abido M A, Elazouni A (2009). Improved crossover and mutation operators for genetic-algorithm project scheduling. In: *IEEE Congress on Evolutionary Computation*. IEEE, 1865–1872
- Afshar A, Fathi H (2009). Fuzzy multi-objective optimization of finance based scheduling for construction projects with uncertainties in cost. *Engineering Optimization*, 41(11): 1063–1080
- Al-Shihabi S T, AlDurgam M M (2017). A max-min ant system for the finance-based scheduling problem. *Computers & Industrial Engineering*, 110: 264–276
- Alavipour S M R, Arditi D (2018a). Impact of contractor's optimized financing cost on project bid price. *International Journal of Project Management*, 36(5): 808–818
- Alavipour S M R, Arditi D (2018b). Optimizing financing cost in construction projects with fixed project duration. *Journal of Construction Engineering and Management*, 144(4): 04018012
- Alavipour S M R, Arditi D (2019a). Maximizing expected contractor profit using an integrated model. *Engineering, Construction, and Architectural Management*, 26(1): 118–138
- Alavipour S M R, Arditi D (2019b). Time-cost tradeoff analysis with minimized project financing cost. *Automation in Construction*, 98: 110–121
- Alghazi A, Elazouni A, Selim S (2013). Improved genetic algorithm for finance-based scheduling. *Journal of Computing in Civil Engineering*, 27(4): 379–394
- Alghazi A, Selim S Z, Elazouni A (2012). Performance of shuffled frog-leaping algorithm in finance-based scheduling. *Journal of Computing in Civil Engineering*, 26(3): 396–408
- Ali M M, Elazouni A (2009). Finance-based CPM/LOB scheduling of projects with repetitive non-serial activities. *Construction Management and Economics*, 27(9): 839–856
- Ammar M A (2011). Optimization of project time-cost trade-off problem with discounted cash flows. *Journal of Construction Engineering and Management*, 137(1): 65–71
- Burns S A, Liu L, Feng C W (1996). The LP/IP hybrid method for construction time-cost trade-off analysis. *Construction Management and Economics*, 14(3): 265–276
- Censor Y (1977). Pareto optimality in multi-objective problems. *Applied Mathematics & Optimization*, 4(1): 41–59
- Chassiakos A P, Sakellariopoulos S P (2005). Time-cost optimization of construction projects with generalized activity constraints. *Journal of Construction Engineering and Management*, 131(10): 1115–1124
- Clough R H, Sears G A, Sears S K, Segner R O, Rounds J L (2015). *Construction Contracting: A Practical Guide to Company Management*. Hoboken, NJ: John Wiley & Sons
- De P, Dunne E J, Ghosh J B, Wells C E (1995). The discrete time cost tradeoff problem revisited. *European Journal of Operational Research*, 81(2): 225–238
- De P, Dunne E J, Ghosh J B, Wells C E (1997). Complexity of the discrete time-cost tradeoff problem for project networks. *Operations Research*, 45(2): 302–306
- Doerner K F, Gutjahr W J, Hartl R F, Strauss C, Stummer C (2008). Nature-inspired metaheuristics for multi-objective activity crashing. *Omega*, 36(6): 1019–1037
- El-Abbasy M S, Elazouni A, Zayed T (2016). MOSCOPEA: Multi-objective construction scheduling optimization using elitist non-dominated sorting genetic algorithm. *Automation in Construction*, 71(2): 153–170
- El-Abbasy M S, Elazouni A, Zayed T (2017). Generic scheduling optimization model for multiple construction projects. *Journal of Computing in Civil Engineering*, 31(4): 04017003
- Elazouni A (2009). Heuristic method for multi-project finance-based scheduling. *Construction Management and Economics*, 27(2): 199–211
- Elazouni A, Abido M (2011). Multi-objective evolutionary finance-based scheduling: Individual projects within a portfolio. *Automation in Construction*, 20(7): 755–766
- Elazouni A, Abido M (2014). Enhanced trade-off of construction projects: Finance-resource-profit. *Journal of Construction Engineering and Management*, 140(9): 04014043
- Elazouni A, Alghazi A, Selim S Z (2015). Finance-based scheduling using meta-heuristics: Discrete versus continuous optimization problems. *Journal of Financial Management of Property and Construction*, 20(1): 85–104
- Elazouni A, Gab-Allah A (2004). Finance-based scheduling of construction projects using integer programming. *Journal of Construction Engineering and Management*, 130(1): 15–24
- Elazouni A, Metwally F (2005). Finance-based scheduling: Tool to maximize project profit using improved genetic algorithms. *Journal of Construction Engineering and Management*, 131(4): 400–412
- Elazouni A M, Metwally F G (2007). Expanding finance-based scheduling to devise overall-optimized project schedules. *Journal*

- of Construction Engineering and Management, 133(1): 86–90
- Fathi H, Afshar A (2010). GA-based multi-objective optimization of finance-based construction project scheduling. *KSCE Journal of Civil Engineering*, 14(5): 627–638
- Gajpal Y, Elazouni A (2015). Enhanced heuristic for finance-based scheduling of construction projects. *Construction Management and Economics*, 33(7): 531–553
- Glover F (1986). Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, 13(5): 533–549
- Horowitz J (1967). *Critical Path Scheduling: Management Control Through CPM and PERT*. New York: Ronald Press
- Kerzner H, Kerzner H R (2017). *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*. Hoboken, NJ: John Wiley & Sons
- Laptali E, Bouchlaghem N, Wild S (1997). Planning and estimating in practice and the use of integrated computer models. *Automation in Construction*, 7(1): 71–76
- Liu S S, Wang C J (2008). Resource-constrained construction project scheduling model for profit maximization considering cash flow. *Automation in Construction*, 17(8): 966–974
- Liu S S, Wang C J (2010). Profit optimization for multi-project scheduling problems considering cash flow. *Journal of Construction Engineering and Management*, 136(12): 1268–1278
- Moussourakis J, Haksever C (2004). Flexible model for time/cost tradeoff problem. *Journal of Construction Engineering and Management*, 130(3): 307–314
- Nkasu M, Leung K (1997). A resources scheduling decision support system for concurrent project management. *International Journal of Production Research*, 35(11): 3107–3132
- Nodet X (2019). CPLEX Optimization Studio 12.9 is available. IBM Developer
- Pearl J (1984). *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Reading, MA: Addison-Wesley Publishing
- Perera S (1980). Linear programming solution to network compression. *Journal of the Construction Division*, 106(3): 315–326
- Peterson S J (2009). *Construction Accounting and Financial Management*. Englewood, New Jersey: Pearson Prentice Hall
- Rudolph G (1994). Convergence analysis of canonical genetic algorithms. *IEEE Transactions on Neural Networks*, 5(1): 96–101
- Siemens N (1971). A simple CPM time-cost tradeoff algorithm. *Management Science*, 17(6): B-354–B-363
- Sonmez R, Bettemir O H (2012). A hybrid genetic algorithm for the discrete time-cost trade-off problem. *Expert Systems with Applications*, 39(13): 11428–11434
- Vanhoucke M, Debels D (2007). The discrete time/cost trade-off problem: Extensions and heuristic procedures. *Journal of Scheduling*, 10(4–5): 311–326
- Wolpert D H, Macready W G (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1): 67–82
- Yang I T (2007). Using elitist particle swarm optimization to facilitate bicriterion time-cost trade-off analysis. *Journal of Construction Engineering and Management*, 133(7): 498–505
- Zheng D X, Ng S T, Kumaraswamy M M (2004). Applying a genetic algorithm-based multi-objective approach for time-cost optimization. *Journal of Construction Engineering and Management*, 130(2): 168–176