Taylor & Francis
Taylor & Francis Group

Engineering Optimization

ISSN: 0305-215X (Print) 1029-0273 (Online) Journal homepage: http://www.tandfonline.com/loi/geno20

Simultaneous planning of the project scheduling and material procurement problem under the presence of multiple suppliers

Babak H. Tabrizi & Seyed Farid Ghaderi

To cite this article: Babak H. Tabrizi & Seyed Farid Ghaderi (2015): Simultaneous planning of the project scheduling and material procurement problem under the presence of multiple suppliers, Engineering Optimization, DOI: 10.1080/0305215X.2015.1114772

To link to this article: http://dx.doi.org/10.1080/0305215X.2015.1114772

Engineering Optimization, 2015
http://dx.doi.org/10.1080/0305215X.2015.1114772

Simultaneous planning of the project scheduling and material procurement problem under the presence of multiple suppliers

Babak H. Tabrizi* and Seyed Farid Ghaderi

School of Industrial Engineering, Faculty of Engineering, University of Tehran, Tehran, Iran

(*Received 17 December 2014; accepted 22 October 2015*)

Simultaneous planning of project scheduling and material procurement can improve the project execution costs. Hence, the issue has been addressed here by a mixed-integer programming model. The proposed model facilitates the procurement decisions by accounting for a number of suppliers offering a distinctive discount formula from which to purchase the required materials. It is aimed at developing schedules with the best net present value regarding the obtained benefit and costs of the project execution. A genetic algorithm is applied to deal with the problem, in addition to a modified version equipped with a variable neighbourhood search. The underlying factors of the solution methods are calibrated by the Taguchi method to obtain robust solutions. The performance of the aforementioned methods is compared for different problem sizes, in which the utilized local search proved efficient. Finally, a sensitivity analysis is carried out to check the effect of inflation on the objective function value.

Keywords: project scheduling; material procurement; net present value; memetic algorithm

1. Introduction

Project scheduling has attracted noticeable attention within the past few decades as a widely used discipline, applicable to many different real-world areas. Resource scarcity in project completion has led to the resource-constrained project scheduling problem being addressed as an interesting research topic. Therefore, numerous studies have been dedicated to the problem, of which several (Herroelen, De Reyck, and Demeulemeester 1998; Brucker *et al*. 1999; Demeulemeester and Herroelen 2002; Kolisch and Hartmann 2006) can be viewed as instrumental archetypes.

This article addresses the project scheduling and material procurement problems simultaneously. Traditionally, these two issues were taken into account as separate problems. Hence, they were treated independently such that the project schedule was considered first and the required materials were ordered afterwards. However, such an approach does not necessarily lead to an optimal result, as many ordering opportunities may be lost. In other words, separate planning of project scheduling and material ordering results in the project officers not accounting for the trade-offs between different cost units, including material ordering and holding costs, purchasing costs, and associated penalties and rewards for project completion with respect to its due date (Najafi, Zoraghi, and Azimi 2011). Consequently, the lack of concurrent planning can result in a considerable increase in total project expenses.

*Corresponding author. Email: babaktabrizi@ut.ac.ir

The first article in the literature to address the integration of project scheduling and material ordering is the work by Aquilano and Smith (1980). They proposed an integrated model to consider both the critical path method and material requirement planning, including materials, lead-times and inventory-level scheduling. Smith-Daniels and Aquilano (1984) developed the previous work by proposing a heuristic solution method, based on the least slack rule, for large-sized project scheduling. They addressed all non-renewable and renewable resources, such as materials, construction components, equipment and labour, in addition to variability in the activity durations.

Smith-Daniels and Smith-Daniels (1987) developed a mixed-integer programming model to obtain an optimal project scheduling and material ordering plan for fixed duration activities. They showed that the given problem can be solved optimally when it is decomposed into a derivation of project scheduling and material ordering plan. They applied the Wagner–Whitin algorithm to determine the optimal material ordering plan for a given schedule. Erbasi and Sepil (1999) also developed a heuristic procedure to find the trade-off between material ordering expenditure and delay in project delivery. However, the purchasing of required resources was treated as a relaxed issue in all of the above models. For instance, the associated costs were calculated in terms of a single price without regard to the real-world purchasing options.

Dodin and Elimam (2001) considered an extended version of the problem by developing variable project worth, rewards/penalty for earlier/later completion of the project than the due date, and material quantity discounts. They also provided a trade-off between the incorporated costs by accounting for variable activity durations. A mixed-integer programming model was used to formulate the problem and some analytical results were presented to enhance the efficiency of the model performance by reducing the problem size. However, the proposed solution process was restricted to networks of up to 30 activities.

Schmitt and Faaland (2004) applied a heuristic to schedule recurrent construction, in which an initial schedule was generated first to dispatch the worker teams to deal with a backlog of products. Then, multiple maximal closure problems were solved to find the material release times, maximizing the net present value (NPV) of the cash flows. Sheikh Sajadieh, Shadrokh, and Hassanzadeh (2009) developed the work proposed by Dodin and Elimam (2001) by applying a genetic algorithm (GA) to obtain a near-optimal solution for larger sized problems. However, it had been assumed that the materials required to process an activity were independent of its duration, which seems impractical in real-world situations. Moreover, the activities' compression costs were considered in terms of a linear proportion, which may not be true for all cases. Finally, they checked the average improvement percentage by applying a local search for cases with different quantities of materials.

Dixit, Srivastava, and Chaudhuri (2014) pursued the uncertainty in materials lead-time in a procurement scheduling model. They applied the proposed model to a real ship-building project, in which materials procurement played a crucial role in the successful completion of the project. The addressed objective function consisted of the holding and shortage costs of the project resources. Moreover, the effect of the allocated stage budget on the project's total costs was tested. The main shortcoming of the above article is the applied fuzzy approach, which could lead to some variations in the output with respect to the ranking method. In further research, Fu (2014) investigated the material batch ordering problem for a multi-mode resource-constrained project scheduling problem. He considered the trade-off between different cost elements and proposed a hybrid algorithm, including an adapted harmony search and GA, to solve the mathematical model. The proposed system was allowed to incur shortage costs; however, the purchase was still assumed to be made by a fixed value.

On the other hand, a project scheduling problem can be regarded through a broad variety of measures, of which financial measures have received much attention in the literature. A financial-based measure, as a criterion to evaluate the project's accomplishments, can present an appropriate reflection of how the project is delivered (Afshar and Fathi 2009). However, in severely competitive business environments, a large number of projects may face failure because they have not incorporated the economic justification criteria (Chen *et al*. 2014). For instance, it has been found that more than 60% of contracts experience failure due to financial causes (Liu and Wang 2008). In other words, at a project level, even a high-profit project may turn out to be a complete failure if financial factors such as interest rates, credit limits and cash flows are not analysed carefully. Therefore, a promising method could integrate the project scheduling problem with the cash-flow management (Kumral 2010).

The existing literature shows that few articles have addressed the time value of a project's cash inflows and outflows in an integrated project scheduling and material ordering problem. Moreover, the resource supply issue has been taken into account in primitive forms, in which the required materials are acquired from a single source, for instance. The purpose of this article is to maximize the NPV of the simultaneous project scheduling and material ordering problem. Furthermore, the potential presence of multiple suppliers is taken into account to better represent real-world procurement conditions. Without loss of generality, the main contributions of this article can be stated as follows.

- The simultaneous planning of project scheduling and the material ordering problem has been addressed here with the purpose of project NPV maximization.
- The material ordering can be managed through the utilization of multiple sources, such that each one may offer a unique overall discount strategy. The issue can be accompanied by the possibility of establishing a trade-off between the procurement costs.
- A GA is applied as the solution method to deal with problems of large size, in addition to a modified version. In this regard, the Taguchi method has been applied to provide robust solutions for the proposed metaheuristics.

The remainder of the article is organized as follows. A mixed-integer mathematical programming model is developed in Section 2. Section 3 discusses the applied GA and its enhanced version, and the implementation steps are explained. The computational results are studied in Section 4, which also incorporates parameter tuning to increase the response robustness and includes the sensitivity analysis for the inflation rate. Finally, conclusions and future research directions are presented in Section 5.

2. Problem description

The proposed mathematical model presented in this section seeks to maximize the net benefit of a project execution by simultaneous project scheduling and material procurement. The problem assumptions are stated first, then the indices, parameters and decision variables are defined.

2.1. *Problem assumptions*

- An activity-on-node (AON) network is considered, including N activities, of which the first and last activities are dummies, representing the start and completion of the project. The precedence relations are all zero-lag, finish-to-start.
- The activities require M partially constrained resources.
- The material requirement of the activities is uniform within the execution duration.
- The whole material amount required to process an activity is ordered at the same time.
- The materials can be provided by different suppliers.

- The cost (*i.e.* cash outflow) of activities execution and the associated benefit (*i.e.* cash inflow) are incurred concurrently with their start and completion times (or an event completion), respectively.

2.2. *Indices*

$j = 1,2, \ldots, N$	Index of project activities
$m = 1,2, \ldots, M$	Index of required materials
$t = 0,1, \ldots, l_N$	Index of time
$s = 1,2, \ldots, S$	Index of suppliers
$k = 1,2, \ldots, K_{ms}$	Index of price discount ranges

2.3. *Parameters*

Pr_j	Set of activities preceding j.
e_j	Earliest finish time of activity j.
l_j	Latest finish time of activity j.
B_j	Benefit of activity j.
C_j	Cost of activity j.
G_{ms}	Ordering cost of material m to supplier s.
δ_{mks}	Unit cost of material m in quantity range k purchased from supplier s.
R_{jm}	Required amount of material m to perform activity j.
h_m	Holding cost of material m.
L_{ms}	Lead-time of material m ordered to supplier s.
d_j	Duration time of activity j.
α_{mks}	Limit on quantity range k of material m for supplier s.
K_{ms}	Number of quantity discount ranges for material m proposed by supplier s.
$(P/F,\ r\%,\ t)$	Single-payment present worth factor, in which r stands for the rate of interest.

2.4. *Decision variables*

x_{jt}	1 if activity j is completed at time t, and 0 otherwise.
λ_{mkst}	1 if material m is ordered within quantity range k to supplier s in period t, and 0 otherwise.
ρ_{mkstj}	1 if material m is ordered within quantity range k to supplier s for activity j in period t, and 0 otherwise.
I_{mt}	Inventory amount of material m in period t.

2.5. *Mathematical model formulation*

The proposed model is presented by a mixed-integer programming problem, as follows.

$$\begin{aligned}
\text{Max} \quad Z = {} & \sum_{j=1}^{N}\sum_{t=e_j}^{l_j} B_j x_{jt}(P/F,\ r\%,\ t) - \sum_{m=1}^{M}\sum_{s=1}^{S}\sum_{t=1}^{l_N-L_{ms}} G_{ms}\sum_{k=1}^{K_{ms}} \lambda_{mkst}(P/F,\ r\%,\ t) \\
& - \sum_{m=1}^{M}\sum_{k=1}^{K_{ms}}\sum_{s=1}^{S}\sum_{t=1}^{l_N-L_{ms}}\sum_{j=1}^{N} \delta_{mks} R_{jm} \rho_{mkstj}(P/F,\ r\%,\ t) - \sum_{m=1}^{M}\sum_{t=1}^{l_N-1} h_m I_{mt}(P/F,\ r\%,\ t) \\
& - \sum_{j=1}^{N}\sum_{t=1}^{l_N} C_j x_{jt}(P/F,\ r\%,\ t-d_j) \qquad (1)
\end{aligned}$$

subject to:

$$\sum_{t=e_i}^{l_i} tx_{it} + d_j \leq \sum_{t=e_j}^{l_j} tx_{jt}; \forall i \in \mathrm{Pr}_j \tag{2}$$

$$\sum_{t=e_j}^{l_j} x_{jt} = 1; \forall j \in 1,\ 2,\ \ldots,\quad N,\ x_{10} = 1 \tag{3}$$

$$I_{mt} = I_{m(t-1)} + \sum_{k=1}^{K_{ms}} \sum_{s=1}^{S} \sum_{j=1}^{N} R_{jm}\rho_{mks(t-L_{ms})j} - \sum_{j=1}^{N} \sum_{t'=\mathrm{Max}(t,e_j)}^{\mathrm{Min}(t+d_j-1,l_j)} \frac{R_{jm}}{d_j} x_{jt'}$$

$$\forall m = 1,\ 2,\ \ldots,\ M,\ \forall t = 1,\ 2,\ \ldots,\ l_N \tag{4}$$

$$\alpha_{m(k-1)s}\lambda_{mkst} \leq \sum_{j=1}^{N} R_{jm}\rho_{mkstj} \leq \alpha_{mks}\lambda_{mkst}; \forall m = 1,\ 2,\ \ldots,\ M,\quad \forall s = 1,\ 2,\ \ldots,\ S,$$

$$\forall t = 1,\ 2,\ \ldots,\ l_N,\quad \forall k \in 1,\ 2,\ \ldots,\ K_m \tag{5}$$

$$\sum_{k=1}^{K_m} \sum_{s=1}^{S} \lambda_{mkst} \leq 1; \forall m = 1,\ 2,\ \ldots,\ M,\quad \forall t \in 1,\ 2,\ \ldots,\ l_N \tag{6}$$

$$\sum_{k=1}^{K_m} \sum_{s=1}^{S} \sum_{t=1}^{l_N-1} \rho_{mkstj} = 1; \forall j = 1,\ 2,\ \ldots,\ N,\quad \forall m \in 1,\ 2,\ \ldots,\ M \tag{7}$$

$$\sum_{k=1}^{K_m} \sum_{s=1}^{S} \sum_{t=1}^{l_N-1} t\rho_{mkstj} + L_{ms} \leq \sum_{t=e_j}^{l_j} tx_{jt} - d_j + 1; \forall j = 1,\ 2,\ \ldots,\ N,\quad \forall m \in 1,\ 2,\ \ldots,\ M \tag{8}$$

$$x_{jt},\ \lambda_{mkst},\ \rho_{mkstj} \in [0,\ 1],\qquad I_{mt} \geq 0 \tag{9}$$

Equation (1) shows the objective function, which aims to maximize the NPV of the profits associated with the project execution. The former part corresponds to the cash inflows, including the income obtained from the execution of each activity; however, the inflows can be practised for some definite events in accordance with the provisions of the contract. The latter part represents the project costs, including material ordering, procuring, holding and executing the activities. The model constraints are shown by Equations (2)–(9), as follows. Equation (2) addresses the precedence constraints in which the successor activity cannot be completed before its predecessors, such that the completion time of an activity is equal to or larger than all its predecessors and its duration time. Equation (3) states that each activity should be completed within its relevant earliest and latest finish bounds. The inventory level of materials is calculated through Equation (4), with respect to the delivered and consumed quantities. Equation (5) shows that the purchased quantities of the materials should fall within the lower and upper discount ranges. Equation (6) is also associated with the discount interval such that the materials' procurement is met in a specific range. Equation (7) binds the procurement of each material to a definite discount interval and time period regarding a supplier. Equation (8) takes the materials lead-time into consideration, so that an activity can be completed only after all its required resources are available. Finally, the domain of decision variables is reflected in Equation (9).

3. Solution methodology

The solution methodologies are described in this section. First, the GA is addressed, including the chromosome representation. The implementation steps are presented concisely, along with a discussion on how to carry out preprocessing in individual generation. Then, an enhanced version of the GA is explained, in which a local search is applied to improve the efficiency of the algorithm.

3.1. *Genetic algorithm*

GA has proved quite efficient in dealing with discrete optimization problems as it can even search in complicated spaces (Jeong and Kim 2011; Khoo, Teoh, and Meng 2014). To initialize the GA, preprocessing is carried out to develop appropriate genotypes, *i.e.* the schedule representation, and leave out infeasible schedules. The proposed chromosome representation is as follows.

Each individual, ∇, is represented by a $(2M + 1) \times N$ matrix. The applied chromosome representation is depicted in Figure 1, in which F_N, S_{MN} and OT_{MN} stand for the activities' finish times, selected suppliers and materials' ordering times, respectively. Figure 2 shows the pseudo-code to develop the individuals, regarding the aforementioned preprocessing.

In the generation of primary individuals, an appropriate lag is considered between the virtual start node and the first activity (or activities) to enable procurement of the required materials. Moreover, a serial generation scheme has been applied to transform the individuals into the corresponding schedules.

Please refer to the supplementary material file for details of the GA application.

$$\nabla = \begin{bmatrix} F_1^{\nabla} & F_2^{\nabla} & \cdots & F_N^{\nabla} \\ S_{11}^{\nabla} & S_{12}^{\nabla} & \cdots & S_{1N}^{\nabla} \\ \vdots & \vdots & & \vdots \\ S_{M1}^{\nabla} & S_{M2}^{\nabla} & \cdots & S_{MN}^{\nabla} \\ OT_{11}^{\nabla} & OT_{12}^{\nabla} & \cdots & OT_{1N}^{\nabla} \\ \vdots & \vdots & & \vdots \\ OT_{M1}^{\nabla} & OT_{M2}^{\nabla} & \cdots & OT_{MN}^{\nabla} \end{bmatrix}$$

Figure 1. Chromosome representation of the proposed model.

Calculate the earliest and latest finish times of each activity considering the precedence relations.

Begin the algorithm; repeat the following steps for the project scheduling and material procurement problem.

  For each activity j assign a random finish time between the corresponding earliest and latest finish times.

    For each material assign an integer random number representing the selected supplier.

      For each material assign an integer random number between $[0, f_j\text{-}d_j\text{+}1\text{-}LT_{ms}]$ representing the ordering time.

      End for.

    End for.

  End for.

End.

Figure 2. Pseudo-code to generate the individuals.

3.2. *Enhanced genetic algorithm*

An enhanced version of the GA has been applied as another solution approach. Enrichment of the solution methodology with the help of a local search can strengthen the model's performance by exploiting the solution space intensively. Consequently, a variable neighbourhood search (VNS) has been applied in the evolutionary process of the GA. In other words, the proposed memetic algorithm (MA) consists of the GA as the main framework and a VNS as the local search. VNS was introduced by Mladenovic and Hansen (1997) as a metaheuristic method to systematically exploit the notion of neighbourhood change in escaping from a local optimal solution. A distinguishing feature of VNS is not to track a trajectory and, in return, to try to search noticeably distant neighbourhoods of the incumbent solution (Corz *et al*. 2012). The explored neighbour is then replaced with the incumbent solution provided that an improvement is attained. The basic steps of a VNS algorithm are summarized in Figure 3.

The integrated method can enhance the solution process by maintaining the diversity, within the convergence. It functions based on the development of more qualified solution neighbours within the individuals' evolution (Xiao *et al*. 2014). The neighbourhood structures of the applied local search should be defined in such a way that feasible solutions are produced. Here, three different types of neighbourhood structure are utilized, as described next.

Neighbourhood structure N_1. This structure intends to improve the objective function with respect to the holding costs. According to this structure, the ordering time preferably approaches the start time of its pertaining activity. For instance, let ∇_{mj}^{OT} be the randomly selected choice. The encompassed value is altered to its possible ceiling amount, *i.e.* $f_j - d_j + 1 - LT_{ms}$, to provide the minimum holding requirement.

Neighbourhood structure N_2. The purpose of this structure is to reduce the procurement costs as an important part of the objective function. Hence, the ordering times of two randomly selected alleles whose suppliers are the same, with different ordering times, are changed to a common value. However, if the common ordering time is appointed equal to the lower value, then the solution feasibility can be preserved. For instance, suppose that $\nabla_{M\times N}^{OT}$ represents part of the ordering schedule for a given individual. Let ∇_{mj}^{OT} and $\nabla_{mj'}^{OT}$ be the selected choices, whose associated suppliers are the same. Then, the ordering time of the choice with the lower value is repeated for the other one. This strategy helps the contractor to purchase in a greater amount and also improves the ordering costs.

Neighbourhood structure N_3. The neighbourhood structure N_3 seeks to facilitate the procurement process by assigning the orders to a narrower number of suppliers, to the maximum possible amount. This means that the whole supply process can be tackled more easily than under circumstances where cooperation is required with more partners. To track the issue by the monetary

Select the set of neighbourhood structures N_k, in which k=1,2,…,k_{Max}.

Use an initial solution x.

While $k \le k_{Max}$,

Generate a random point x' from the kth neighbour of x

Apply some local search method for x', represented by x''

Accept x'' and move it if it has led to an improvement ($x := x''$) and follow the searching process with the present neighbourhood structure;

Otherwise, $k \leftarrow k+1$

End while.

Figure 3. Pseudo-code of the variable neighbourhood search.

Repeat the following steps for the project scheduling and material procurement problem, respectively.

- Generate random feasible individuals until their number equals the given population size.
- Begin the algorithm and continue such that the termination criterion, maximum iteration, is met.
  - Evaluate each individual's fitness.
  - Determine the elite individuals.
  - Select pairs for mating.
    - Apply crossover operator.
    - Apply mutation operator.
  - Select a subset of individuals to undergo the improvement by the VNS method.
  - Replenish the population.
- Check for the termination criterion.
- Loop, if not terminated, otherwise

End.

Figure 4. Pseudo-code of the proposed memetic algorithm. VNS = variable neighbourhood search.

measure, moves are welcomed that do not increase the objective function value, or can even reduce it. For instance, consider the ∇th individual and let $\nabla^{S}_{M\times N}$ be the part in which the suppliers are selected. Two alleles, ∇^{S}_{mj} and $\nabla^{S}_{m'j'}$, are randomly selected and the corresponding value of the first allele is directly copied into the second one. Afterwards, the change is approved if $OF(\nabla_{\text{New}}) \leq OF(\nabla_{\text{Old}})$.

The pseudo-code of the MA is shown in Figure 4.

4. Computational experiments

The model is tested here to check its performance and applicability in practice. To do so, it is solved by *GAMS 22.1* solver and the obtained results are compared with those obtained from the metaheuristics. The comparisons are carried out on different problems to take into consideration the effect of the problem size increase. First, the underlying parameters are calibrated by the Taguchi method to provide robust solutions. These settings are performed with regard to the problem size to let the algorithms function more effectively. Then, the numerical results are discussed and a sensitivity analysis is carried out to investigate the effect of inflation on the project's NPV. All computations are performed on a Core i3 Pentium 4 PC with 2.0 GHz CPU speed and 4 GB of RAM. The metaheuristics are coded with C++ programming language.

4.1. *Experimental settings*

Taguchi (1986) presented a development of the fractional factorial experiment matrices which could reduce the number of required experiments along with the sufficient preservation of information. The orthogonal arrays in the Taguchi method make it possible to study a large number of decision variables with a small number of experiments (Montgomery 2000; Ruiz and Maroto 2006; Khoo, Teoh, and Meng 2014). This separates the factors into controllable and noise groups. The functional purpose of the method is to minimize the effect of the noise factors and determine the optimum values of the significant controllable factors, due to the robustness concept (Jeong and Kim 2011). Thus, it has been successfully applied as a systematic approach in many different optimization problems (Xu *et al*. 2013, 2014).

Taguchi uses the signal-to-noise (S/N) ratio to reflect the extant variation in the response variable. The terms 'signal' and 'noise' denote desirable (response variable) and undesirable (standard deviation) values, respectively. A higher S/N ratio points to smaller variation in the desirable value. Equation (10) shows the above ratio calculation for a maximization problem, in which y_i and n signify the ith experiment response value and number of orthogonal arrays, respectively.

$$S/N = -10 \ \log \left(\frac{\sum_{i=1}^{n} 1/y_i^2}{n} \right) \tag{10}$$

The key factors of the GA include the population size, number of iterations, mutation and crossover operators, reproduction strategy and crossover probability. On the other hand, the local search factor has been added to considerations for the MA. (Please refer to the supplementary material file for the selection of the above-mentioned factor levels.)

Here, as a general rule, the continuous compounding method has been applied to account for the time value of money as today's investments are established according to this state. Equation (11) shows the continuous compounding single sum present worth factor to be substituted for the single-payment present worth factor used in the mathematical model. P and F stand for the present equivalent and the corresponding future value, respectively. Moreover, n is associated with the relevant time period, which should be adjusted with regard to the duration of the project activities.

$$P = Fe^{-rn} \tag{11}$$

As stated earlier, the problem is investigated by the proposed solution methodologies for three different size levels, *i.e.* small, medium and large projects. Projects with 30 activities have been categorized as small-sized projects, projects with 60 and 90 activities as medium-sized cases, and projects with 120 activities as large-sized projects. To deal with the problem, different typical projects have been taken into account by the network generator in the resource-constrained project scheduling problem library (PSPLIB). The extant library facilitates the projects' network generation; however, it is necessary to use some complementary issues to develop fitted networks. Hence, Table 1 has been applied to generate the required data, according to the model requirements.

To determine the best factor levels, Figure 5 has been used as a representative to show the *S/N* ratio plot obtained by the GA for mean objective function fitness values of the small-sized projects. The results were obtained by repeating each array five times. According to the results, levels 4, 2, 2, 1, 3 and 2 are selected as the best fitted for factors A, B, C, D, E and F, respectively. All selected factor levels are shown in Table 2 according to the solution method and the problem size.

Table 1. Data generation method.

Parameter	Random distribution function
B_j	$\sim$ U [250, 350]
C_j	$\sim$ U [60, 100]
δ_{mks}	$\sim$ U [3, 8]
G_{ms}	$\sim$ U [5, 10]
h_m	$\sim$ U [1, 5]
α_{mks}	$\sim$ U [5, 15]
R_{jm}	$\sim$ U [1, 4]
d_j	$\sim$ U [1, 10]
L_{ms}	$\sim$ U [1, 15]
K_{ms}	$\sim$ U [1, 3]
r	$\sim$ U [0.04, 0.1]

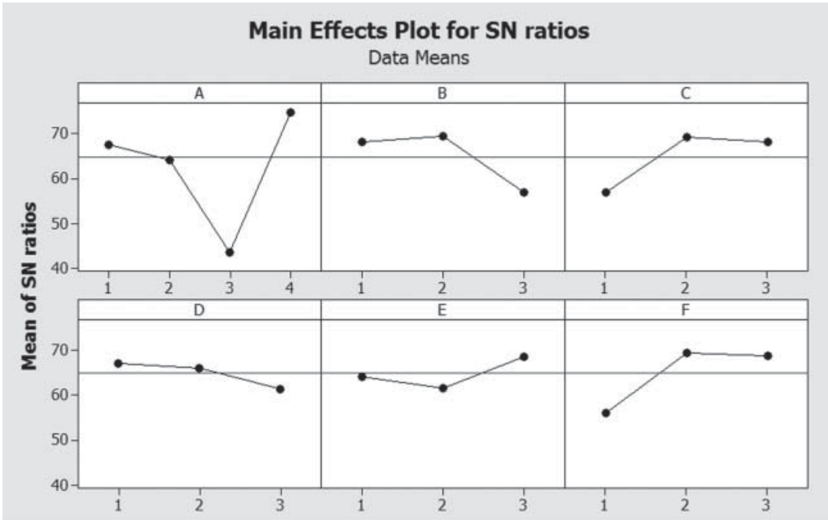

Figure 5. Main effects plot for genetic algorithm factor levels in small-sized problems. SN = signal-to-noise ratio.

Table 2. Selected factor levels for the solution methodologies.

Solution method	GA			MA		
Problem size	Small	Medium	Large	Small	Medium	Large
Selected factors	4, 2, 2, 1, 3, 2	2, 2, 2, 2, 2, 1	2, 3, 3, 3, 3, 3	4, 2, 1, 3, 1, 2, 1	2, 2, 2, 1, 3, 1, 2	4, 3, 3, 1, 1, 2, 2

Note: GA = genetic algorithm; MA = memetic algorithm.

Engineering Optimization

4.2. *Numerical results*

Initially, the performance of the mathematical formulation is considered for a typical instance, details of which can be found in the supplementary material file. Hereafter, the mathematical model is tested by the aforementioned metaheuristic solution methodologies to compare their performance with that of GAMS. The results are compared according to the objective function measures and the solution elapsed time. The objective function status (*i.e.* fitness function) is reflected according to the central and dispersion measures, in which the best and mean values of the objective function are determined for each of the instances, in addition to the standard deviation values. The results are compiled in Tables 3–6 in terms of the average values, obtained from 10 repetitions of each instance.

The computational experiments are reported according to the number of project activities, in which the solution process was intentionally interrupted for those instances with no optimal output after a specific elapsed time. The maximum allowed limit for the solution process was set to 2 h and the obtained objective function interval is reported for the circumstances exceeding the given limit, instead of the optimal value. Application of the objective function mean value can represent the convergence with the whole generation landscape, while the best objective function value can be highlighted for comparison with the ordinary branch-and-bound method.

According to Table 3, GAMS faced some restrictions in finding the optimal solutions for the last two instances, and this is reflected in the intervals found. The metaheuristic methods outperformed GAMS for instances with two or more suppliers and four or more materials. In addition, it can be seen that the MA requires more time to reach the termination criterion, as

Table 3. Results obtained for projects with 30 activities.

Problem size	GAMS		GA				MA			
				OF value				OF value		
($N \times S \times M$)	Elapsed time (s)	OF value	Elapsed time (s)	Mean	Best	StD	Elapsed time (s)	Mean	Best	StD
(30 × 1 × 1)	14	4263.7	34	4221.0	4263.7	39.6	41	4260.8	4263.7	11.2
(30 × 1 × 2)	21	5904.2	49	5856.9	5904.2	41.5	63	5898.2	5904.2	40.7
(30 × 1 × 3)	38	3911.4	73	3837.0	3911.4	40.3	84	3907.0	3911.4	29.2
(30 × 2 × 1)	18	4095.3	61	4029.7	4095.3	44.8	86	4050.2	4095.3	41.5
(30 × 2 × 2)	40	5766.9	95	5663.0	5766.9	82.8	117	5758.8	5766.9	88.6
(30 × 2 × 3)	73	4014.2	140	3953.9	4013.9	33.9	154	4007.7	4010.3	30.1
(30 × 2 × 4)	329	4934.6	229	4835.9	4926.3	60.4	280	4904.9	4928.5	68.3
(30 × 2 × 5)	484	2955.1	294	2884.1	2955.1	50.2	338	2904.8	2955.1	45.8
(30 × 2 × 6)	943	2778.0	384	2719.6	2765.9	31.3	445	2727.9	2778.0	30.2
(30 × 3 × 1)	282	4024.6	252	3952.1	4024.6	40.7	290	3968.2	4024.6	32.5
(30 × 3 × 2)	419	3975.7	275	3912.0	3962.8	44.6	297	3939.9	3969.7	33.9
(30 × 3 × 3)	2985	4022.8	338	3938.3	4022.8	68.6	394	3970.5	4022.8	61.5
(30 × 3 × 4)	6976	5019.2	396	4858.5	5011.7	41.2	470	4955.9	5018.0	67.2
(30 × 3 × 5)	7200[a]	[3100,3450]	481	3202.4	3291.3	61.0	522	3235.3	3291.3	52.7
(30 × 3 × 6)	7200[a]	[2530,2960]	570	2650.9	2701.5	32.8	637	2651.0	2712.2	29.7

Note: [a]The model was stopped for exceeding 7200 seconds.
GA = genetic algorithm; MA = memetic algorithm; OF = objective function; StD = standard deviation.

Table 4. Results obtained for projects with 60 activities.

Problem size	GAMS		GA				MA			
				OF value				OF value		
($N \times S \times M$)	Elapsed time (s)	OF value	Elapsed time (s)	Mean	Best	StD	Elapsed time (s)	Mean	Best	StD
(60 × 1 × 1)	6384	14,140.2	591	13,871.5	14,140.2	215.2	684	13,885.676	14,140.2	207.3
(60 × 1 × 2)	7200[a]	[10000,11530]	756	10,684.9	10,836.7	128.6	923	10,796.568	10,927.7	115.7
(60 × 1 × 3)	7200[a]	[10240,11350]	958	10,507.1	10,743.5	196.7	1036	10,577.67	10,760.6	170.6
(60 × 2 × 1)	7200[a]	[13220,16790]	729	14,528.7	14,886.0	330.8	861	14,581.226	14,894.0	284.8
(60 × 2 × 2)	7200[a]	[10830,11920]	894	11,114.3	11,399.3	255.4	992	11,154.606	11,428.9	241.7
(60 × 2 × 3)	7200[a]	[9860,11340]	1150	10,224.0	10,529.4	273.9	1428	10,314.77	10,546.8	216.6
(60 × 2 × 4)	7200[a]	[8430,9630]	1468	8930.4	9206.6	258.2	1664	9038.148	9222.6	167.9
(60 × 2 × 5)	7200[a]	[5820,8400]	1752	6743.7	6988.3	216.5	1937	6807.4972	6996.4	173.1
(60 × 2 × 6)	7200[a]	[5900,8150]	2018	6593.3	6818.4	204.1	2489	6651.5475	6822.1	154.5
(60 × 3 × 1)	7200[a]	[13870,14920]	914	14,238.0	14,528.6	253.2	1136	14,290.816	14,567.6	230.1
(60 × 3 × 2)	7200[a]	[11600,12750]	1466	12,011.6	12,281.9	249.6	1712	12,046.791	12,305.2	228.8
(60 × 3 × 3)	7200[a]	[10230,11700]	1964	10,664.3	10,937.8	247.6	2264	10,757.754	10,977.3	207.3
(60 × 3 × 4)	7200[a]	[8800,11000]	2490	9178.3	9433.0	230.1	2846	9239.3913	9456.9	196.9
(60 × 3 × 5)	7200[a]	[5500,7800]	2931	7263.4	7488.1	194.3	3195	7333.3953	7490.7	137.6
(60 × 3 × 6)	7200[a]	[5200,7900]	3219	6424.1	6650.3	195.2	3688	6572.526	6693.0	104.2

Note: [a]The model was stopped for exceeding 7200 seconds.
GA = genetic algorithm; MA = memetic algorithm; OF = objective function; StD = standard deviation.

it needs further attempts for the local search. On the other hand, the best objective function values of the GA and MA show very close results compared with GAMS. Likewise, the objective function mean values denote an acceptable closeness to the optimal value, as approved by the standard deviation measure. The MA could show better results than the GA from this point of view by providing less dispersed solutions for most of the instances.

Table 4 shows that GAMS is not economical for larger problems and only a solution bound has been obtained for almost all instances. In return, the metaheuristics can yield convergence in a reasonable CPU time, similarly to Table 3. The absolute superiority of the MA to the GA

Table 5. Results obtained for projects with 90 activities.

Problem size	GAMS		GA				MA			
				OF value				OF value		
$(N \times S \times M)$	Elapsed time (s)	OF value	Elapsed time (s)	Mean	Best	StD	Elapsed time (s)	Mean	Best	StD
$(90 \times 1 \times 1)$	7200[a]	[16100,19000]	3012	16,609.7	17,018.2	376.3	3254	17,152.6	17,538.5	188.4
$(90 \times 1 \times 2)$	7200[a]	[13600,17100]	3316	13,976.6	14,498.6	466.2	3480	14,528.3	14,779.6	102.4
$(90 \times 1 \times 3)$	7200[a]	[11600,15400]	3865	12,872.2	13,366.8	450.5	4028	13,126.4	13,380.7	152.2
$(90 \times 2 \times 1)$	7200[a]	[11000,14900]	3184	16,904.6	17,463.5	487.0	3387	17,459.0	17,906.7	244.5
$(90 \times 2 \times 2)$	7200[a]	[12200,16800]	3639	14,334.0	14,611.7	241.8	3822	14,653.6	14,922.3	201.7
$(90 \times 2 \times 3)$	7200[a]	[10400,15200]	4020	12,842.3	13,280.6	388.9	4339	12,957.9	13,290.2	186.7
$(90 \times 2 \times 4)$	7200[a]	–	4419	11,293.1	11,751.5	403.1	4679	11,691.8	11,979.4	128.6
$(90 \times 2 \times 5)$	7200[a]	–	4566	10,046.4	10,530.9	422.7	4920	10,427.7	10,728.1	150.9
$(90 \times 2 \times 6)$	7200[a]	–	4973	5710.3	6008.4	239.1	5266	5908.7	6085.2	114
$(90 \times 3 \times 1)$	7200[a]	[14800,19900]	3971	16,961.3	17,432.0	427.4	4175	17,455.3	17,829.8	206.3
$(90 \times 3 \times 2)$	7200[a]	[13300,18800]	4255	14,361.2	14,882.1	473.3	4449	14,742.5	15,028.1	201.7
$(90 \times 3 \times 3)$	7200[a]	[9900,16500]	4684	12,621.1	13,119.7	419.4	4977	12,916.5	13,234.2	167.1
$(90 \times 3 \times 4)$	7200[a]	–	4937	11,101.8	11,673.9	508.7	5296	11,445.2	11,750.8	165.2
$(90 \times 3 \times 5)$	7200[a]	–	5688	10,362.3	10,884.8	472.6	6058	10,790.6	11,044.7	120.4
$(90 \times 3 \times 6)$	7200[a]	–	6390	5773.5	6096.7	269.8	6627	5955.3	6139.5	88.4

Note: [a]The model was stopped for exceeding 7200 seconds.
GA = genetic algorithm; MA = memetic algorithm; OF = objective function; StD = standard deviation.

Table 6. Results obtained for projects with 120 activities.

Problem size	GA				MA			
		OF value				OF value		
$(N \times S \times M)$	Elapsed time (s)	Mean	Best	StD	Elapsed time (s)	Mean	Best	StD
$(120 \times 1 \times 1)$	5243	21,616.4	22,216.3	576.3	5624	22,315.9	22,771.4	401.6
$(120 \times 1 \times 2)$	5583	20,186.2	20,767.7	559.4	5908	20,448.7	20,844.8	358.4
$(120 \times 1 \times 3)$	5927	17,104.8	17,725.2	596.8	6578	17,948.9	18,371.5	386.9
$(120 \times 2 \times 1)$	5397	21,569.2	22,190.6	582.9	5837	21,949.5	22,420.4	428.5
$(120 \times 2 \times 2)$	5861	19,865.5	20,628.8	752.4	6740	20,391.0	20,913.9	480.3
$(120 \times 2 \times 3)$	6316	17,678.6	18,453.7	748.2	7228	18,160.5	18,664.5	461.1
$(120 \times 2 \times 4)$	6785	13,576.9	14,216.7	602.8	7937	14,302.2	14,775.0	423.7
$(120 \times 2 \times 5)$	7246	11,605.2	12,190.4	547.5	8536	12,321.3	12,702.4	354.2
$(120 \times 2 \times 6)$	7944	6,565.6	6,962.5	337.9	9775	7,065.12	7,344.2	257.7
$(120 \times 3 \times 1)$	6428	21,635.8	22,631.6	958.9	6968	22,376.8	22,950.6	512.8
$(120 \times 3 \times 2)$	6754	19,846.9	20,869.6	984	8502	20,593.3	21,208.4	582
$(120 \times 3 \times 3)$	7179	17,520.2	18,579.3	1019.8	8829	18,217.2	18,917.2	616.4
$(120 \times 3 \times 4)$	7633	13,055.6	14,083.8	1006.5	8664	14,024.7	14,593.9	538.3
$(120 \times 3 \times 5)$	8257	11,697.1	12,673	942.4	9710	12,355.5	12,897.2	495.9
$(120 \times 3 \times 6)$	9390	6,173.9	7,321.4	932.7	11,258	7,355.3	7,687.4	304.2

Note: GA = genetic algorithm; MA = memetic algorithm; OF = objective function; StD = standard deviation.

can be observed for both the concentration and dispersion measures. The applied VNS method has proved efficient in dealing with a schedule with higher NPV of execution. However, the corresponding effect can be seen on the CPU time.

Apparently, GAMS was unable to find even a solution interval for most of the instances with a project network including 90 activities. Table 5 reiterates that the proposed MA could show even more efficient performance as its results are accompanied by a narrower domain and better representatives for the best objective function value. The results indicate that utilization of a proper local search can enhance the quality of solutions for large-sized problems.

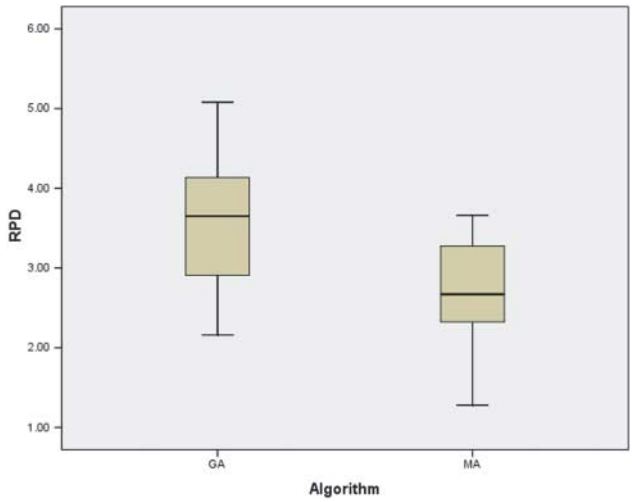

Figure 6. Relative percentage deviation (RPD) comparison of the proposed algorithms.

Finally, the projects with 120 activities are taken into account in Table 6. The GAMS output has not been reported as it could not end in any acceptable bound within the given time. On the other hand, the metaheuristics have been exceptionally enabled to be run under a 3 h elapsed time so that the convergence process can be pursued in a more qualified way. The MA provokes solutions with smaller dispersion compared with the GA. The experiments show that the MA can function more appropriately within the solution process, although the required CPU time increases.

At the end of the comparisons, the statistical validity of the results was investigated by the relative percentage deviation (RPD) measure. RPD calculates the deviation percentage of a replication with the best given value. Figure 6 shows the results of the RPD measure, in which the enhanced version of the GA proved more efficient.

4.3. *Sensitivity analysis*

This section seeks to determine the effect of inflation on the project execution benefit. Accounting for the inflation factor can illustrate how a project schedule may change. In other words, it is of great importance to find the schedules that provide the contractor with the most NPV, with regard to the given inflation rate.

To deal with this issue, the project shown in Figure A1 (see supplementary material) is used again. Let β stand for the inflation rate, for which a varied interval between 0.02 and 0.20 has been considered. The cost elements of the objective function should change in the presence of inflation. Hence, Equation (11) should be rewritten as Equation (12), as follows, and be applied to the activities' execution, procurement, holding and ordering terms. Figures 7 and 8 show the results of sensitivity analyses.

$$P = Fe^{(\beta - r)n} \tag{12}$$

Figure 7 represents the effect of inflation rate on the objective function value in accordance with the project size. Likewise, Figure 8 compares the NPV of projects of different sizes for different numbers of suppliers. As can be seen, inflation has a greater effect on the NPV for larger projects since the fluctuation domain increases. Hence, the effect of inflation on the project

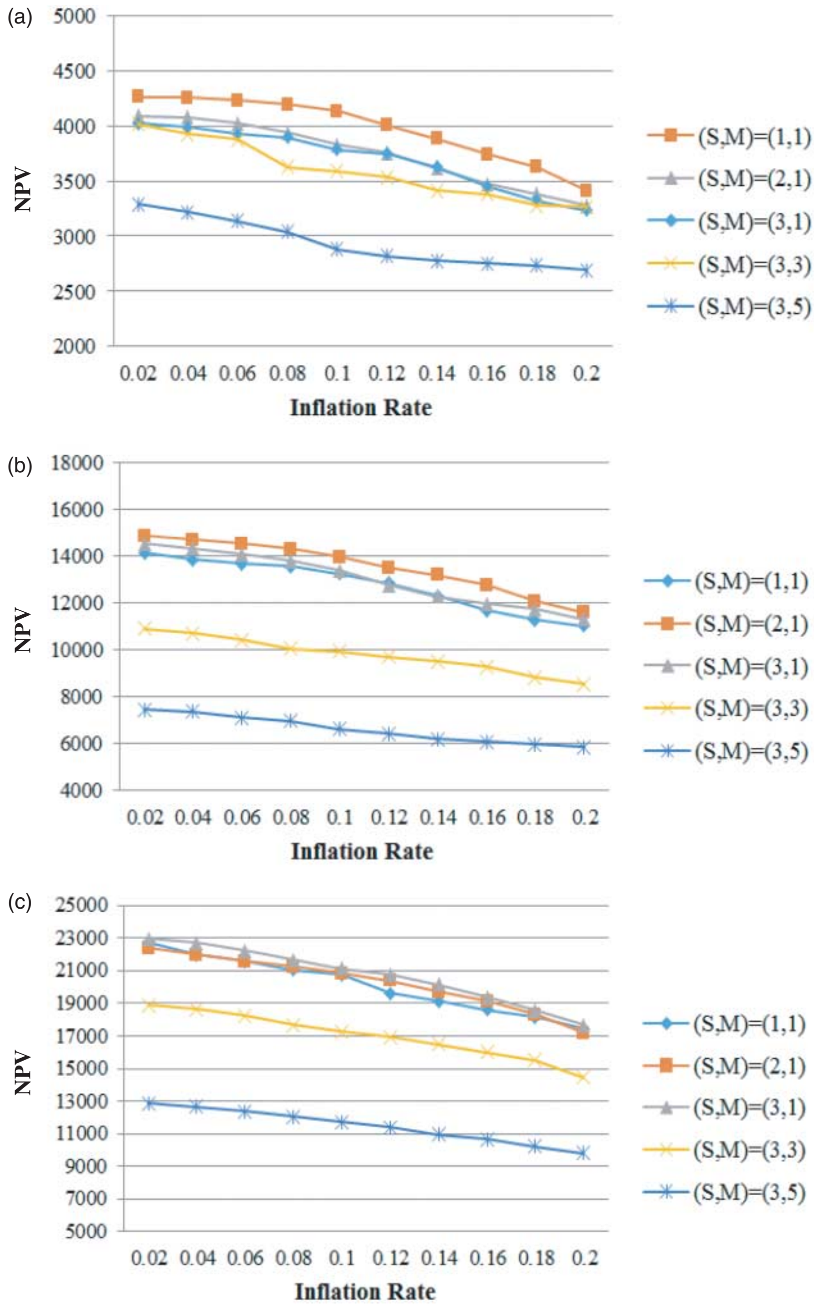

Figure 7. The project's net present value (NPV) with respect to the inflation rate: (a) small-sized projects; (b) medium-sized projects; (c) large-sized projects.

performance can be highlighted as a crucial issue. Furthermore, the results indicate that the NPV slope increases for higher values of inflation. Thus, projects implemented in environments with more unstable economic circumstances may confront more serious threats if no attention is paid to the inflationary factor. Changes in the optimum schedule of the given network have been incorporated in the supplementary material file.

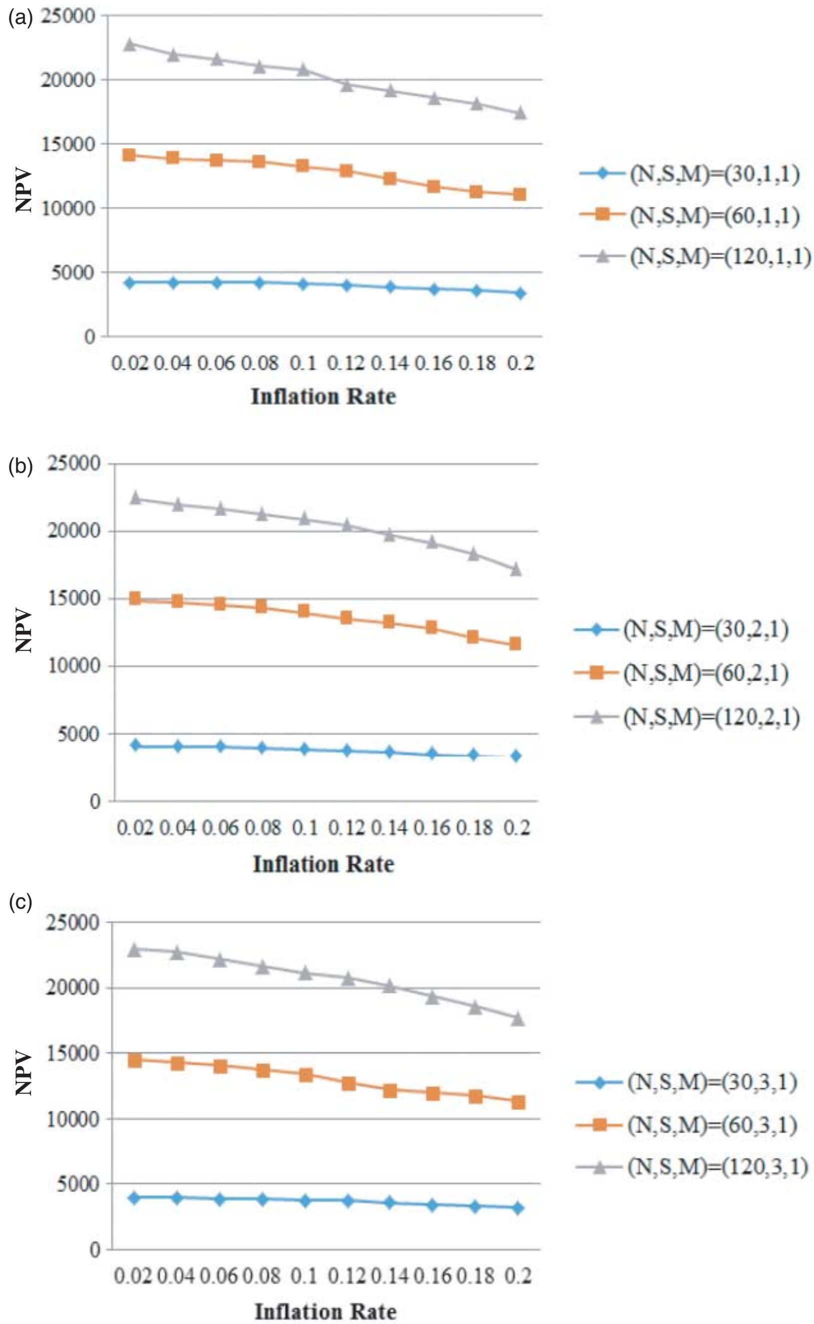

Figure 8. Comparison of the project's net present value (NPV) with different sizes for a given number of suppliers: (a) single-supplier network; (b) double-supplier network; (c) triple-supplier network.

5. Conclusions

The simultaneous planning of the project scheduling problem and material procurement was addressed in this article. The proposed mathematical model accounted for multiple suppliers, suggesting the associated overall discount strategy from which to procure the materials. Two solution methodologies were applied to consider the model performance, namely a GA and an MA. The key factors of the solution methods were tuned by the Taguchi method to provide robust solutions. This was taken into consideration for problems with different sizes. Afterwards, the mixed-integer programming model function was tested and the applicability and efficiency of the applied methods were approved. The performance comparisons were published with respect to the central and variation measures. In particular, in this regard, the MA could outperform the GA for larger problems. However, the solution elapsed time increased to some extent because of the embedded local search, which could carry out the scheduling strategies.

This article did not take some executive perspectives into account, which could be enhanced in future research studies. For instance, the effect of other local search methods on the model performance could be investigated. Likewise, dealing with the solution elapsed time is a concern and other heuristics could be tested to improve the time spent. The segregation of execution costs in terms of the need for equipment, skilled workforce and work regulations could also be considered in future research. Finally, the incorporation of uncertainty in the mathematical formulation could be regarded as another future research direction.

Disclosure statement

No potential conflict of interest was reported by the authors.

Supplemental data

Supplemental data for this article can be accessed at http://dx.doi.org/10.1080/0305215X.2015.1114772.

References

Afshar, A., and H. Fathi. 2009. “Fuzzy Multi-objective Optimization of Finance-Based Scheduling for Construction Projects with Uncertainties in Cost.” *Engineering Optimization* 41 (11): 1063–1080.

Aquilano, N. J., and D. E. Smith. 1980. “A Formal Set of Algorithms for Project Scheduling with Critical Path Method—Material Requirements Planning.” *Journal of Operations Management* 1 (2): 57–67.

Brucker, P., A. Drexl, R. Mohring, K. Neumann, and E. Pesch. 1999. “Resource Constrained Project Scheduling: Notation, Classification, Models, and Methods.” *European Journal of Operational Research* 112: 3–41.

Chen, M., S. Yan, S. S. Wang, and C. L. Liu. 2014. “A Generalized Network Flow Model for the Multi-mode Resource-Constrained Project Scheduling Problem with Discounted Cash Flows.” *Engineering Optimization* 47 (2): 165–183.

Corz, A., J. A. Gomez-Ruiz, J. I. Pelaez, E. Tenorio, and J. Veintimilla. 2012. “Design and Optimization of Symmetric Laminated Composites Using a Variable Neighbourhood Search-Based Model.” *Engineering Optimization* 44 (4): 505–520.

Demeulemeester, E. L., and W. S. Herroelen. 2002. *Project Scheduling: A Research Handbook*. Dordrecht: Kluwer.

Dixit, V., R. K. Srivastava, and A. Chaudhuri. 2014. “Procurement Scheduling for Complex Projects with Fuzzy Activity Durations and Lead Times.” *Computers and Industrial Engineering* 76: 401–414.

Dodin, B. and A. A. Elimam. 2001. “Integrated Project Scheduling and Material Planning with Variable Activity Duration and Rewards.” *IIE Transactions* 33: 1005–1018.

Erbasi, A. and C. Sepil. 1999. “A Modified Heuristic Procedure for Materials Management in Project Networks.” *International Journal of Industrial Engineering-Theory* 6 (2): 132–140.

Fu, F. 2014. “Integrated Scheduling and Batch Ordering for Construction Project.” *Applied Mathematical Modelling* 38 (2): 784–797.

Herroelen, W., B. De Reyck, and E. Demeulemeester. 1998. “Resource-Constrained Project Scheduling: A Survey of Recent Developments.” *Computers and Operations Research* 25: 279–302.

Jeong, B. J., and K. H. Kim. 2011. “Scheduling Operations of a Rail Crane and Container Deliveries Between Rail and Port Terminals.” *Engineering Optimization* 43 (6): 597–613.

Khoo, H. L., L. E. Teoh, and Q. Meng. 2014. “A Bi-objective Optimization Approach for Exclusive Bus Lane Selection and Scheduling Design.” *Engineering Optimization* 46 (7): 987–1007.

Kolisch, R., and S. Hartmann. 2006. “Experimental Investigation of Heuristics for Resource-Constrained Project Scheduling: An Update.” *European Journal of Operational Research* 174: 23–37.

Kumral, M. 2010. “Robust Stochastic Mine Production Scheduling.” *Engineering Optimization* 42 (6): 567–579.

Liu, S. S., and C. J. Wang. 2008. “Resource-Constrained Construction Project Scheduling Model for Profit Maximization Considering Cash Flow.” *Automation in Construction* 17: 966–974.

Mladenovic, N., and P. Hansen. 1997. “Variable Neighborhood Search.” *Computers and Operations Research* 24: 1097–1100.

Montgomery, D.C. 2000. *Design and Analysis of Experiments*. 5th ed. New York: Wiley.

Najafi, A. A., N. Zoraghi, and F. Azimi. 2011. “Scheduling a Project to Minimize Costs of Material Requirements.” *World Academy of Science, Engineering and Technology* 78: 134–137.

Ruiz, R., and C. Maroto. 2006. “A Genetic Algorithm for Hybrid Flow Shops with Sequence Dependent Setup Times and Machine Eligibility.” *European Journal of Operational Research* 169: 781–800.

Schmitt, T., and B. Faaland. 2004. “Scheduling Recurrent Construction.” *Naval Research Logistics* 51 (8): 1102–1128.

Sheikh Sajadieh, M., S. Shadrokh, and F. Hassanzadeh. 2009. “Concurrent Project Scheduling and Material Planning: A Genetic Algorithm Approach.” *Scientia Iranica- Transaction E: Industrial Engineering* 16 (2): 91–99.

Smith-Daniels, D. E., and N. J. Aquilano. 1984. “Constrained Resource Project Scheduling Subject to Material Constraints.” *Journal of Operations Management* 4 (4): 369–387.

Smith-Daniels, D. E., and V. L. Smith-Daniels. 1987. “Optimal Project Scheduling with Materials Ordering.” *IIE Transactions* 19 (4): 122–129.

Taguchi, G. 1986. *Introduction to Quality Engineering*. White Plains: Asian Productivity Organization/UNIPUB.

Xiao, Y., Q. Zhao, I. Kaku, and N. Mladenovic. 2014. “Variable Neighbourhood Simulated Annealing Algorithm for Capacitated Vehicle Routing Problems.” *Engineering Optimization* 46 (4): 562–579.

Xu, Y., L. Wang, S. Wang, and M. Liu. 2013. “An Effective Shuffled Frog-Leaping Algorithm for Solving the Hybrid Flow-Shop Scheduling Problem with Identical Parallel Machines.” *Engineering Optimization* 45 (12): 1409–1430.

Xu, Y., L. Wang, S. Wang, and M. Liu. 2014. “An Effective Hybrid Immune Algorithm for Solving the Distributed Permutation Flow-Shop Scheduling Problem.” *Engineering Optimization* 46 (9): 1269–1283.