*Journal of Optimization in Industrial Engineering 10 (2012) 19-27*

# An Integrated Model of Project Scheduling and Material Ordering: A Hybrid Simulated Annealing and Genetic Algorithm

Nima Zoraghi[a], Amir Abbas Najafi[b], Seyed Taghi Akhavan Niaki[c,*]

[a]*M.Sc., Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran*

[b]*Assistant Professor, Department of Industrial Engineering, K.N. Toosi University of Technology, Tehran, Iran*

[c]*Professor, Department of Industrial Engineering, Sharif University of Technology, Tehran, Iran*

Received 5 September, 2011; Revised 16January, 2012; Accepted 25 February, 2012

**Abstract**

This study aims to deal with a more realistic combined problem of project scheduling and material ordering. The goal is to minimize the total material holding and ordering costs by determining the starting time of activities along with material ordering schedules subject to some constraints. The problem is first mathematically modelled. Then a hybrid simulated annealing and genetic algorithm is proposed to solve it. In addition, some experiments are designed and the Taguchi method is employed to both tune the parameters of the proposed algorithm and to evaluate its performance. The results of the performance analysis show the efficiency of the proposed methodology.

*Keywords:* Project scheduling; Material ordering; Hybrid simulated annealing;Taguchi design.



## 1. Introduction and Literature Review

As project management has become an important field in numerous modern industries, a careful management of projects is an absolute necessity to preserve the competitiveness of companies. A big challenge for the project manager is to achieve the project goals that have been outlined in the project plan. The project scheduling is the core of the project plan and involves finding a schedule for activities of the project subject to some limitations such as precedent constraints.

Material planning is another important element in a project plan. The material required for the project must be available at the right time. It is the responsibility of a project manager to make sure the employees have the material required to perform their activities.In fact, a big challenge for a project manager is to coordinate the project scheduling and the material ordering in a project plan. Traditionally, these two issues have been treated independently. In other words, the project is first scheduled and then the demand profiles of all the materials are generated from the obtained schedule. This approach usually results in a non-optimal solution for the material planning where projects are frequently delayed and as a result, resources are wasted due to material shortages. Thus, the best solution is obtained when the project scheduling and the material ordering are performed simultaneously.

In this paper, the evolution of the integrated problem is reviewed and the impact of activities starting time as a decision variable on the materials plan is investigated. Based on this strategy, project scheduling is first carried out and then, by taking the activity schedule (activities starting times) as a known parameter, a material ordering plan is determined. These strategies provide scheduling flexibility that can lead to further reduction in the project's total cost or make-span. Taking this approach also helps project managers to make trade-offs between the cost elements such as material ordering and holding costs.

Aquilano and Smith (1980) introduced the integrated problem of project scheduling and material ordering (PSMO). They developed a model consisting of material and inventory levels scheduling that integrates the critical path method and the material requirement planning. They presented a set of formal CPM/MRP algorithms that may be used to compute the early and late start schedules as well as the critical sequence. In their techniques, the CPM is initially designed to schedule projects only subject to precedence constraints. Later, additional techniques were introduced to consider the constraints upon various aspects of resource availability.

In a subsequent research, Smith-Daniels and Aquilano (1984) presented an improvement over their original treatment and proposed a heuristic scheduling based on the least slack rule and extended this model for scheduling large projects where requirements for both renewable and non-renewable

* Corresponding author E-mail address: Niaki@Sharif.edu

resources are incorporated. In order to investigate the effects of capital and materials costs in an integrated form, Smith-Daniels and Smith-Daniels (1987a) presented a project-scheduling problem where the net present value of a project is maximized subject to capital and material constraints. Through a series of examples, they showed the necessity of considering materials costs and constraints that lead to lower total project costs and recommended developing heuristic approaches due to the computational complexities involved.

In another study, Smith-Daniels et al. (1987b) proposed a mixed integer 0-1 programming to formulate an integrated problem of project scheduling and materials ordering with a fixed activity duration that provides an optimal schedule of project activities and materials orders. They showed that an optimal solution might be found by decomposing the problem into the derivation of the project schedule and the subsequent derivation of materials lot sizes and that the latest starting time schedule provides an optimal solution to the problem.

Dodin and Elimam(2001)extended Smith-Daniels and Smith-Daniels (1987b)'s work to include the two factors of reward and penalty policy for the project and variable activity duration. Sajadiehetal. (2009)proposed a genetic algorithm to solve a PSMO problem where the ordering times of materials for each activity were determined independently.

In the present study, the PSMO problem is extended in a way that the ordering times of materials are corporately determined by considering all activity requirements. This approach makes the total cost reduce significantly. In addition, an efficient hybrid algorithm is proposed to find the solution, especially for large-scale problems. The rest of the paper is organized as follows: In Section 2, the problem is precisely defined, the assumptions and notations are introduced, and the mathematical model of the PSMO problem is presented. The details of a hybrid meta-heuristics algorithm are provided in Section 3. Parameters calibration of the algorithm is carried out in Section 4. In Section 5, comparisons and computational results of the proposed algorithm are investigated. Finally, the conclusion comes in Section 6.

## 2. Problem Definition, Assumptions and Notations

In a typical PSMO problem, assuming that the project consists of$n$activities, $(i = 1,2, \dots, n)$, each activity is carried out without interruption. The model is imposed by zero-lag finish-to-start precedence constraints on the sequencing of activities and is shown by an activity-on-node network with no loop. Activities 1and$n$are dummies and represent the project start and completion, respectively. An activity$i$ has a fixed duration$d_i$. Further, for an activity$i$, there is a set of predecessor activities$P_i$. The execution of the ith activity requires $f, (f = 1,2, \dots, F)$,types of materials (non-renewable) over its duration. The resource usage over an activity is taken uniform and a typical activity$i$ uses$U_{if}$units of material$f$per period. In addition,$A_f$and$H_f$denotethe ordering cost and the holding cost per unit of the$f^{th}$materialper unit time, respectively. The type and the quantity of all materials must be determined at the beginning of each period. The activities are to be scheduled such that the make-span of the project does not exceed a given deadline $(DD)$. The capacity of the warehouse is considered unlimited and the lead-time is assumed to be inappreciable. The problem is to find the starting time of the project activities and the materials ordering schedule such that the precedence relation of activities are satisfied and the total material holding and ordering cost is minimized.

According to the assumptions and notations introduced, the PSMO problem is formulated as a mixed integer programming model as follows:

$$Min\ Z = \sum_{f=1}^{F} \sum_{t=0}^{DD-1} A_f \lambda_{ft} + \sum_{f=1}^{F} \sum_{t=0}^{DD} H_f I_{ft} \quad (1)$$

subject to

$$\sum_{t=ES_j}^{LS_j} tX_{jt} + d_j \le \sum_{t=ES_i}^{LS_i} tX_{it} \quad ; \quad \forall j \in P_i \quad (2)$$

$$\sum_{t=ES_i}^{LS_i} X_{it} = 1 \quad ; \quad i = 1,2, \dots, n \quad (3)$$

$$\sum_{t=ES_n}^{LS_n} tX_{nt} \le DD \quad (4)$$

$$I_{ft} = I_{f(t-1)} + Q_{ft} - \sum_{i=1}^{n} \sum_{w=Max(t-d_i+1,ES_i)}^{Min(t,LS_i)} U_{if} \times X_{iw} \ ;$$

$$f = 1,2, \dots, F \quad , \quad t = 1,2, \dots, DD \quad (5)$$

$$I_{f0} = Q_{f0} (6)$$

$$Q_{ft} \le \lambda_{ft} \times M \quad (7)$$

$$X_{it} = \{0,1\} \ ; \ i = 1,2, \dots, n \ , \ t = ES_i, \dots, Ls_i \quad (8)$$

$$\lambda_{ft} = \{0,1\} \ ; \ f = 1,2, \dots, F \ , \ t = 0,1, \dots, DD-1 \quad (9)$$

$$Q_{ft} \ge 0 \ ; \ f = 1,2, \dots, F \ , \ t = 0,1, \dots, DD-1 \quad (10)$$

$$I_{ft} \ge 0 \ ; \ f = 1,2, \dots, F \ , \ t = 0,1, \dots, DD \quad (11)$$

where the decision variables are defined as:

$X_{it}$: A binary variable, equals one if activity$i$is started in period$t$ and zero otherwise

$\lambda_{ft}$: A binary variable, equals one if material$f$is ordered in period$t$and zero otherwise

$Q_{ft}$: The ordered quantity of material $f$in period$t$

$I_{ft}$: The inventory level of material$f$in period$t$

Moreover, $ES_i$is the earliest starting time, $Ls_i$is the latest start time of activity$i$, and $M$represents a large number expressed as$M = \sum_{f=1}^{F} \sum_{i=1}^{n} U_{if} \times d_i$.

The proposed PSMO is a new model, which is very different from the one developed by Dodin and Elimam (2001).The objective function (1) minimizes the total costs of the problem. It consists of two parts: the material ordering costs and the material holding costs. Inequality (2) enforces the precedence relations between activities. Equation (3) states that every activity must be started only once. Constraint (4) ensures that the project ends by the latest allowable completion time. Constraints (5) and (6) balance the levels of the materials over the project execution. Inequality (7) denotes the relationship between $\lambda_{ft}$ and $Q_{ft}$. Constraints (8)-(11) denote the domains of the variables.

Since Sajadieh et al. (2009)showed the PSMO belongs to the class of the NP-hard problems and since the above-mentioned model is a generalization of the PSMO problem, it is NP-hard as well. Therefore, an efficient meta-heuristic algorithm is required to solve the extended PSMO model. In the next section, a hybrid simulated annealing and genetic algorithm is developed for this purpose.

## 3. A Solution Procedure

One of the most interesting recent trends in meta-heuristic algorithms has been the hybridization of different techniques in solving optimization models. An important branch of hybridization is the enhancement of meta-heuristics with additional techniques to improve run-times, results, or both. In this section, an efficient hybrid meta-heuristic algorithm is proposed to solve the optimization model given in Section 2. The detailed framework of the algorithm is presented below.

### *3.1. The proposed hybrid algorithm*

Simulated annealing (SA) is one of the best meta-heuristics that was introduced by Kirkpatrick et al.(1983).This algorithm has been widely employed by several researchers to solve NP-hard problems of large sizes. It attempts to solve hard combinatorial optimization problems through a controlled randomization. The proposed hybrid algorithm to solve the extended PSMO model consists of two loops. In the first loop, SA attempts to find a schedule for activities. In the second loop, a genetic algorithm (GA) is applied to find the best materials ordering policy for the obtained schedule such that the minimum total holding and ordering cost of materials is obtained.

While in a usual scheme, the SA starts with the generation of an initial solution (one point), the SA part of the proposed hybrid algorithm of this research starts with the generation of several initial solutions (multi-point). Providing a suitable solution for the initialization of the algorithm is an important task and has a significant influence on the output. In this study, the initial solution is generated by the critical path method (CPM), through which the earliest start time($ES_i$) and the latest start time ($LS_i$)of the activities are obtained. Then the results of forward and backward pass computations allow for the calculation of float values of the network activities. Assuming the earliest possible and the latest allowable start time of the single end node of the network are equal, the total float denotes the time an activity can be delayed without causing a delay in the project. In other words, the total float (total slack) of activity $i$,($TF_i$), is defined as:

$$TF_i = LS_i - ES_i \tag{12}$$

Moreover, the floating time of an activity at a given schedule is equal to the difference between the activity's starting time at the schedule and its earliest starting time, that is, the floating time of activity $i$ at a given schedule $H$, $F_H(i)$, is obtained as follows:

$$F_H(i) = ST_H(i) - ES_i \tag{13}$$

where $ST_H(i)$ denotes the starting time of activity $i$ at schedule $H$. We note that the used floating time of an activity should be less than or equal to its total floating time, i.e.

$$0 \le F_H(i) \le TF_i \tag{14}$$

An initial solution is denoted by a vector of $n-2$ elements $F = (F(2), F(3), \dots, F(n-1))$, where the position of each element corresponds to the number of a non-dummy activity and its value denotes the floating time of the activity. As a result, initial solutions (schedules) can be generated randomly from the feasible region of vector $F$. In order to evaluate the objective function for a given feasible solution, both the starting times of all activities and the order quantities of the required materials are needed. Based on the $F$ vector, the schedule vector $ST = (ST(2), ST(3), \dots, ST(n-1))$ that contains the start time of the activities is obtained by adding up the elements of the vector $F$ and the earliest starting times of the activities (according to Eq. 13). Then, to evaluate a schedule, its near optimal material requirement planning is determined using a genetic algorithm (explained in Section 3.2).

The SA continues by generating the neighborhoods of initial solutions. The roulette wheel procedure is applied as a neighborhood search structure to generate new feasible solutions. Through this mechanism, each activity can move backward or forward to a new position based on its floating time. More specifically, a uniform random number is first generated in the interval $[2, n-1]$ for each activity. Then, the activities can move to their new positions based on their corresponding floating times and the generated random numbers. To avoid infeasible solutions, the floating times of the activities given in equation (14) must be met. As an example, the neighborhood structure of the proposed SA for generating new feasible solutions is illustrated in Figure1, in which an activity can shift forward (positive numbers in the second row) and backward (negative numbers in the second row) based on the generated uniform random number.

| Random number generated at interval [2,N-1] | [0,0.1] | [0.1,0.5] | [0.5,0.9] | [0.9,1] |
|---|---|---|---|---|
| Move activity (Shift forward or backward) | -2 | -1 | 1 | -2 |

Fig. 1. An example of the neighborhood structure of the proposed SA

Cooling scheme is one of the important parameters of the SA algorithm with the basic aim of controlling its behavior. When the SA proceeds, the temperature is gradually lowered under a certain mechanism called the cooling schedule. In this paper, a linear cooling scheme is applied to decrease the temperature.

With regard to this issue, consider the following project with seven activities:

| $A_i$ | D | Succ. | EST | EFT | LST | LFT | TF | $F_H$ |
|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 2,3,4 | 0 | 0 | 0 | 2 | 0 | 0 |
| 2 | 3 | 5 | 0 | 3 | 4 | 7 | 4 | 2 |
| 3 | 5 | 5,6,7 | 0 | 5 | 2 | 7 | 2 | 0 |
| 4 | 4 | 5,6 | 0 | 4 | 3 | 7 | **3** | **1** |
| 5 | 6 | 7 | 5 | 11 | 7 | 13 | 2 | 0 |
| 6 | 2 | 7 | 5 | 7 | 11 | 13 | 6 | 4 |
| 7 | 0 | 0 | 11 | 11 | 13 | 13 | 2 | 0 |

One initial solution of the HSA is:

| **Activities** | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|
| **$F_H$** | **2** | **0** | **1** | **0** | **4** |

A random number is generated at the interval [2,6] to select the position that requires a change. Another random number is generated between [0,1] to shift the activities based on Figure 1. If this random number is 0.4, then activity 4 must be shifted to one time earlier as follows:

| **Activities** | **2** | **3** | **4** | **5** | **6** |
|---|---|---|---|---|---|
| **$F_H$** | 2 | 0 | 1-1=0 | 0 | 4 |

Note that according to Eq. (14),activity 4 is allowed a shifting in the interval$0 \leq 1 \leq 3$.

The details of the proposed GA in finding the order quantities of materials and determining the minimum total ordering and holding costs are described in the next subsection. If the new vector results in a better value of the objective function, the new solution vector replaces the current solution. This procedure iterates until the algorithm is not able to find better solutions.

### 3.2. The interior GA algorithm

The development of the interior GA for finding the best material ordering policy of a given activity schedule is described in this subsection. The GA starts with the generation of an initial population, i.e., the first generation. The initial population is randomly generated according to the demand profiles of the materials in the activity schedule. In order to create the next generation, after computing the fitness values of the individuals, two operations are performed: crossover and mutation. Then, the population is randomly partitioned into pairs of individuals. To each resulting pair of individuals(parents), the crossover operator is applied with probability$P_{cr}$ to produce two new (children) individuals. After the crossover operation, each individual is considered for the mutation operation with probability$P_{mu}$. Finally, the algorithm stops if the specified number of generations, denoted by Gen, is created. The best individual of the last generation is the best ordering policy of the GA for the given activity schedule.

#### 3.2.1. Chromosome representation

In this research, a real mode is used to code the search points of the solution based on the ordered quantities of the materials in each period. Each individual chromosome, $Q$, is a matrix of$F$ rows (for $F$types of materials) and$(DD-1)$ columns (for periods 0 to$DD-1$), where an element (gene)$Q_{ft}$represents the ordered quantity of material$f$in period $t$. Figure2 presents the general form of a chromosome. To create the initial population, the ordered quantities in each period for a given activity schedule are randomly generated according to the demand profile of the materials. In order to evaluate the chromosome, the fitness value obtained from the objective function of the problem is evaluated.

$$\begin{bmatrix} Q_{10} & \cdots & Q_{1,DD-1} \\ \vdots & \vdots & \vdots \\ Q_{F0} & \cdots & Q_{F,DD-1} \end{bmatrix}$$

Fig. 2. A chromosome of the interior GA

#### 3.2.2. Crossover

In the crossover operation, two parents are selected by the roulette wheel strategy to create two children. The Uniform continuous crossover operator is employed in this research. specifically, consider two individuals$P^1$and$P^2$ selected for a crossover operation.First, a vector$\alpha = [\alpha_t]_{1*DD}$is drawn randomly based on the uniform distribution in the interval [0,1], where$\alpha_t$is used as the crossing point. Then, each gene of the two children$CH^1$and $CH^2$are obtained using $CH^1_{ft} = P^1_{ft}(\alpha_t) + P^2_{ft}(1-\alpha_t)$and$CH^2_{ft} = P^1_{ft}(1-\alpha_t) + P^2_{ft}(\alpha_t)$. Note that this crossover operation generates feasible solutions.

#### 3.2.3. Mutation

To describe this operation, let$Q$be the chromosome that is selected for mutation. First, an integer random number,$R$, is generated in the interval$[1, F]$to select one type of material. Then, two random numbers,$r_1$ and $r_2$such that$r_1 < r_2$ , are generated in the interval$[0, DD-1]$. Hence, $(Q_{R,r_1}, \ldots, Q_{R,r_2})$are the genes of the child considered for mutation. Next, the amount of each gene in the new chromosome$(Q^M)$ is obtained as follows:

$$Q^M_{R,r_1} = \sum_{j=r_1}^{r_2} Q_{R,j}$$
$$Q^M_{R,t} = 0 \quad ; \quad t = r_1 + 1, \ldots, r_2 \qquad (15)$$
$$Q^M_{ft} = Q_{ft} \quad ; \quad otherwise$$

This mutation can reduce the material ordering costs of the schedules.Figure3 shows an example of the mutation operation in a specific row of a chromosome.

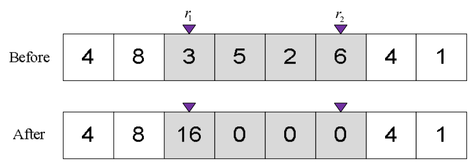


Fig. 3. An example of the mutation operation

## 4. Tuning the Parameters

It is well known that the choice of parameters and operators has a major influence on the efficiency of meta-heuristic algorithms. While the proper design of parameters and operators highly depend on the type of problems, most researchers often set the parameters and operators manually based on the reference values of the previous similar research works. However, in this study, the behavior of different operators and parameters of the proposed hybrid algorithm, HSA thereafter, is studied, and hence the algorithm is tuned to give better near optimum solutions.

There are several statistical ways of designing experiments to calibrate the algorithm, with the most frequent approach being the full factorial experiments (Montgomery (2000)). However, in this paper, the Taguchi approach is employed to reduce the number of required experiments (Ross (1989)). In the Taguchi method, orthogonal arrays are used to study a large number of decision variables with a small number of experiments. Taguchi created a transformation of the repetition data to another value called the measure of variation. The transformation is the signal-to-noise($S/N$) ratio, which explains why this type of parameter design is called robust design. In the Taguchi approach, the objective functions are categorized into three groups of "the smaller-the-better," "the larger-the-better," and "the-nominal-value-is-expected," each having a particular formula for the ($S/N$)ratio.

The most common performance measure, including the mean flow time, used in the literature to compare all the algorithms is the average of the relative deviation index($\overline{RDI}$)defined as

$$\overline{RDI} = \frac{Alg_{sol} - Min_{sol}}{Max_{sol} - Min_{sol}} \times 100 \qquad (16)$$

where$Alg_{sol}$ is the average of the solutions obtained from a given algorithm, $Min_{sol}$is the average of the best solutions obtained among all algorithms (or the best known solution, possibly optimal), and $Max_{sol}$is the average of the worst solutions obtained in each iteration. For each configuration of the algorithm, the response$\overline{RDI}$is considered in the Taguchi orthogonal design. Since the aim of this paper is to minimize$\overline{RDI}$, "the smaller-the better" type is considered for $S/N$as (Phadke(1989))

$$S/N = -10 * log\left[\left(\frac{1}{n}\right)\sum_{i=1}^{n} y_i^2\right] \qquad (17)$$

where$y_i$ refers to the value of a response.In what follows, the parameters along with their levels are first introduced. Then the proper scheme of the Taguchi method is selected. Next, the results are analyzed through the analysis of variance (ANOVA). Finally, the best combination of the parameters is selected for the tuned HSA.

There are seven parameters (factors) that may affect the performances of the proposed hybrid algorithm. The factors and their levels are presented in Table 1.The number of degrees of freedom is required to select the suitable orthogonal array. One degree of freedom for the overall mean, five degrees of freedom for Max-It with six levels, and two degrees of freedom for each of the other six three-levels factors make the total degrees of freedom 1+1×5+6×2=18(Shokrollahpour et al., 2009). Thus, the selected orthogonal array should have a minimum of 18 rows and 7 columns to accommodate the seven factors. From the standard table of orthogonal arrays, the$L_{18}$is selected as the fittest orthogonal array design that fulfills all the minimum requirements. The selected orthogonal array $L_{18}(6^1, 3^6)$is shown in Table 2, where the control factors are assigned to the columns of this matrix and the corresponding integers indicate the levels of the factors. The experiments are carried out for a set of problems, each with 18 activities. Each trial is experimented with five instances to yield more reliable information (each instance is tackled five times.) Hence, there are 25 results for each trial to perform the statistical analyses. The results are analyzed by the response$\overline{RDI}$, for which the$S/N$is obtained.To explore the relative significance of individual factors in terms of their main effects on the response, the analysis of variance (ANOVA) is conducted. The results of the analysis on the sum of squares of the factors are presented in Table 3. Since the SSX of the factor SA-Points, Pop-Size, and$T_0$is lower than the sum of squared error (SSE), they all can be pooled in the error term (Karimi et al. (2010)), resulting in the pooled ANOVA that is shown in Table 4.

Table 1
Factors and their levels

| Factors | | Levels | | | | | |
|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 |
| *Max-It* | Maximum iteration of the SA | 50 | 60 | 70 | 80 | 90 | 100 |
| *SA-Points* | Number of points (solutions) to be searched in the solution space by the HSA | 5 | 10 | 15 | – | – | – |
| $T_0$ | Initial temperature of the SA | 3000 | 4000 | 5000 | – | – | – |
| *Pop-size* | Population size of the GA | 15 | 25 | 35 | – | – | – |
| *Gen* | Maximum number of GA generation | 100 | 200 | 300 | – | – | – |
| $P_{cr}$ | Crossover probability of the GA | 0.75 | 0.85 | 0.95 | – | – | – |
| $P_{mu}$ | Mutation probability of the GA | 0.05 | 0.15 | 0.25 | – | – | – |

Table 2
'The orthogonal array$L_{18}(6^1, 3^6)$ forthe parameters of the hybrid SA

| Trial | *Max-It* | *SA-Points* | $T_0$ | *Pop-Size* | *Gen* | $P_{cr}$ | $P_{mu}$ |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 1 | 2 | 2 | 2 | 2 | 2 | 2 |
| 3 | 1 | 3 | 3 | 3 | 3 | 3 | 3 |
| 4 | 2 | 1 | 1 | 2 | 2 | 3 | 3 |
| 5 | 2 | 2 | 2 | 3 | 3 | 1 | 1 |
| 6 | 2 | 3 | 3 | 1 | 1 | 2 | 2 |
| 7 | 3 | 1 | 2 | 1 | 3 | 2 | 3 |
| 8 | 3 | 2 | 3 | 2 | 1 | 3 | 1 |
| 9 | 3 | 3 | 1 | 3 | 2 | 1 | 2 |
| 10 | 4 | 1 | 3 | 3 | 2 | 2 | 1 |
| 11 | 4 | 2 | 1 | 1 | 3 | 3 | 2 |
| 12 | 4 | 3 | 2 | 2 | 1 | 1 | 3 |
| 13 | 5 | 1 | 2 | 3 | 1 | 3 | 2 |
| 14 | 5 | 2 | 3 | 1 | 2 | 1 | 3 |
| 15 | 5 | 3 | 1 | 2 | 3 | 2 | 1 |
| 16 | 6 | 1 | 3 | 2 | 3 | 1 | 2 |
| 17 | 6 | 2 | 1 | 3 | 1 | 2 | 3 |
| 18 | 6 | 3 | 2 | 1 | 2 | 3 | 1 |

Table 3
The sum of squares

| Factor | *Max-It* | *SA-Points* | $T_0$ | *Pop-Size* | *Gen* | $P_{cr}$ | $P_{mu}$ |
|---|---|---|---|---|---|---|---|
| SSX | 204.145 | 204.139 | 204.137 | 204.1392 | 204.827 | 204.1399 | 204.636 |
| SSE | | | | 623.4 | | | |
| SST | | | | 2053.53 | | | |

Table 4
The pooled ANOVA for the S/N ratio

| Factors | DF | SS | MS | F | PX(%) | P-Value |
|---|---|---|---|---|---|---|
| *Gen* | 2 | 204.8273 | 102.41365 | 0.986 | 0.14 | 0.02 |
| $P_{mu}$ | 2 | 204.6366 | 102.31828 | 0.985 | 0.15 | 0.01 |
| *Max-It* | 5 | 204.1453 | 40.82906 | 0.393 | 15.36 | 0.005 |
| $P_{cr}$ | 2 | 204.1400 | 102.06999 | 0.982 | 0.18 | 0.03 |
| Error | 6 | 1235.8 | 205.964 | | | |
| Total | 23 | 2053.53 | | | | |

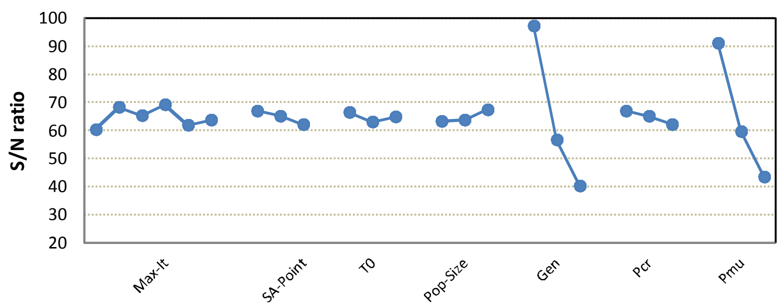


Fig. 4. The effect chart of the S/N ratios of the HSA factors

The results in Table 4 indicate that the factors Gen, Max-It, $P_{mu}$, and $P_{cr}$all are significant at 0.95 confidence level. In other words, they have significant impacts on the robustness of the proposed HSA algorithm. In order to find the optimal levels of the significant factors, the $S/N$ratios obtained at different combinations of the factor levels are depicted in Figure 4.

As Figure 4 shows, a better robustness of the algorithm is achieved when the significant parameters are set as Gen=100, $P_{mu} = 0.05$, Max-It = 80, and$P_{cr} = 0.75$. However, finding the optimal levels of the other factors including SA-Points, Pop-Size, and$T_0$ requires further investigation. In order to investigate the effects of the parameters on the relative deviation index,$\overline{RDI}$, the averages of$\overline{RDI}$for the make-span at each factor level are obtained and are plotted in Figure 5.

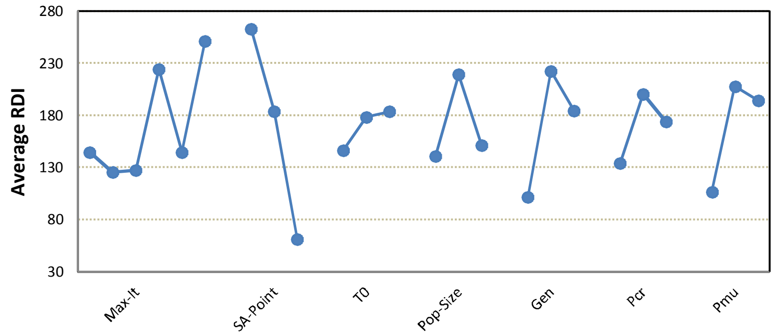


Fig. 5. The mean plot for each level of the HSA factors

Since the goal is to minimize the average of $\overline{RDI}$, based on Figure5, the optimal level of the factors Max-It, SA-Point,$T_0$,Pop-Size, Gen,$P_{cr}$, and$P_{mu}$may be obtained. Besides, the statistical significance of the factors on the$\overline{RDI}$is employed and the results are summarized in Table 5.

Based on the results in Table 5, all factors have significant effects on the$\overline{RDI}$at 95% confidence level. Besides, the SA-Point has the largest effect on the quality of the algorithm with the relative importance of 42%. Since we already showed that Gen, $P_{mu}$, Max-It, and $P_{cr}$have significant impacts on the$S/N$ ratio, only SA-Points, Pop-Size and $T_0$are considered as the adjustment factors of the $\overline{RDI}$analysis. Then, based on Figure5, the highest level of SA-Points (15), the lowest level of Pop-Size (15), and the lowest level of$T_0$(3000) result in the best performances in terms of$\overline{RDI}$.

Finally, the optimum values of the HSA parameters are obtained by considering both the $S/N$ratio and$\overline{RDI}$. The results are reported in Table 6.



Table 5
Results of the ANOVA test for the average of the relative deviation index

| Factors | Df | SS | MS | F | PX(%) | P-Value |
|---|---|---|---|---|---|---|
| SA-Point | 2 | 125156.2 | 62578.115 | 802.021 | 42.00 | 0.005 |
| Gen | 2 | 47053.2 | 23526.610 | 301.524 | 15.76 | 0.004 |
| Max-It | 5 | 44958.2 | 8991.635 | 115.240 | 14.97 | 0.003 |
| Pmu | 2 | 37652.4 | 18826.223 | 241.283 | 12.60 | 0.01 |
| Pop-Size | 2 | 22423.2 | 11211.605 | 143.691 | 1.842 | 0.02 |
| Pcr | 2 | 14471.5 | 7235.772 | 92.736 | 1.189 | 0.03 |
| T0 | 2 | 5849.9 | 2924.951 | 37.487 | 0.480 | 0.05 |
| Error | 0.295 | 2 | 0.1475 | | | |
| Total | 608.281 | 17 | | | | |

Table 6
The optimum levels and values for the parameters of the HSA algorithm

| Levels | Max-It | SA-Points | T0 | Pop-Size | Gen | $P_{Cr}$ | $P_{Mut}$ |
|---|---|---|---|---|---|---|---|
| 1 | | | ✓ | ✓ | ✓ | ✓ | ✓ |
| 2 | | | | | | | |
| 3 | | ✓ | | | | | |
| 4 | ✓ | — | — | — | — | — | — |
| 5 | | — | — | — | — | — | — |
| 6 | | — | — | — | — | — | — |
| Values | 80 | 15 | 3000 | 15 | 100 | 0.75 | 0.05 |

## 5. Experiments and Comparisons

In this section, the results of examining the proposed HSA in some test problems are reported. Since there is no exact or heuristic algorithm to get the optimal or near-optimal solution for this model, we solve the mathematical modeling of the test instances by the LINGO solver software. Although due to the nature of the problem LINGO is unable to obtain a global optimal solution for all of the test instances, we inevitably assume that the solution obtained by LINGO is a good one to compare (Najafi and Niaki, 2006).

To get a PSMO with a simple precedence relation, i.e. finish-to-start with zero time lag and no loop, the first three collections of instances with 10, 20, and 30 non-dummy activities with 1, 2, and 3 resources are generated by PROGEN for the experiments. For each combination set of the non-dummy activities and resources, 10 problems are examined. Thus, in total there are 90 test problems. To convert these problems into PSMO instances, the ordering and holding costs of the materials are generated randomly using a uniform distribution in the interval [1,25]. In order to evaluate the performance of the proposed HSA, a MATLAB computer program of the proposed HSA is first coded and then employed for the 90 created test problems. To compare the results of the HSA and LINGO, the problems are classified into small (containing 10 activities), medium (containing 20 activities), and large-scale sets (containing 30 activities) described in the following sections.

Table 7 shows the computational results of the proposed algorithm, in which columns A and B denote the number of instances LINGO and the HSA were able to find a local optimal solution within the given CPU time of 3600 seconds, respectively. The next column displays the average of the relative percentage deviation of instances, $\overline{RPD}$, where $RPD$ is obtained through Eq. (18).

$$RPD = \frac{Objective\ function\ (LINGO) - Objective\ function\ (HSA)}{Objective\ function\ (HSA)} \quad (18)$$

Further, columns D and E demonstrate the average CPU time required to obtain the solutions by the proposed HSA and LINGO in seconds, respectively. To evaluate the performance of the proposed HSA in small- sized problems, 30 problems with 10 non-dummy activities are considered. The results in Table 7 show that the proposed HSA has reached good solutions in shorter amounts of CPU time than LINGO, except for the problems with one resource. In addition, a test of hypothesis on the means of the quality of the solutions by LINGO and the HSA was employed. The results of the statistical test showed that at 95% confidence level there is no significant difference between the mean solutions of the two methods.

For the medium-sized problems, the results reveal that while there are many instances the LINGO solver is unable to solve, there is a solution found by the proposed method. Furthermore, the average relative-deviation-percentages of the instances solved by LINGO are not high. In addition, the results of a statistical test show that at 95% confidence level there is no significant statistical difference between the mean solutions obtained by LINGO and the one obtained by the proposed method. However, the amounts of CPU time for the proposed method are significantly less than those obtained by LINGO.

For the large-sized problems, once again, we observe that while there are many instances that the LINGO solver is unable to solve in 3600 seconds, there is a solution found by the HSA in a short amount of CPU time. For example, in problems with 3 resources, only one out of 10 test problems were solved by LINGO. In addition, regarding the problems that LINGO was able to find a solution for, the results of a statistical test show that there is no significant difference between the solutions obtained by LINGO and the ones obtained by the HSA at 95% confidence level.

We also compared the solutions obtained by HSA with the best randomly generated solutions after a time limit. Moreover, to test the efficiency of HSA, the best solution of the initial population (*BI*) is compared with the best solution found by the algorithm (*BA*) after a time limit. We generated 120 instances with 30, 60, 90 and 120 activities and 1 to 3 materials. The following measure is used to obtain improvement percentages:

$$\%\,Improve = 100\left(\frac{BI - BA}{BI}\right) \quad (19)$$



Table 7
The computational results and a comparison between the proposed SA and LINGO.

| No. of activities | No. of resources | No. of problems | A | B | $\overline{RPD}$(%) | D | E |
|---|---|---|---|---|---|---|---|
| 10 | 1 | 10 | 10 | 10 | -0.7 | 23.6 | 1.4 |
| | 2 | 10 | 10 | 10 | -0.8 | 25.8 | 279 |
| | 3 | 10 | 10 | 10 | -0.8 | 27.4 | 347.3 |
| 20 | 1 | 10 | 8 | 10 | -0.8 | 51.8 | 357.1 |
| | 2 | 10 | 6 | 10 | -0.8 | 55.4 | 512.2 |
| | 3 | 10 | 4 | 10 | -0.8 | 56.8 | 1070 |
| 30 | 1 | 10 | 5 | 10 | -0.8 | 71.3 | 1094 |
| | 2 | 10 | 3 | 10 | -0.8 | 73.6 | 1899 |
| | 3 | 10 | 1 | 10 | -1.2 | 77.4 | 2941 |

Table 8 shows that the algorithm improves the best unfitness value obtained from the initial population. The results in Table 8 for very large-sized problem show that the HSA is efficient to solve the problem with a logical improvement percentage. As the table indicates, when the problem size increases, the percentage improvement of the algorithm increases as well. In summary, the results of experiments on the 90 test problems of different sizes show that for 10, 20, and 30-activity problems, the mean solution obtained through the proposed HSA is almost the same as the average solution obtained by LINGO in significantly shorter amounts of CPU time. Moreover, for both 20 and 30activity problems, while there are many instances the LINGO solver is unable to find a solution, there is a solution obtained by the proposed method.

Table 8
The computational results of the large-sized problems.

| No. of activities | No. of resources | No. of Problems | Time Limit (S) | Ave. Improve % | Min. Improve % |
|---|---|---|---|---|---|
| | 1 | 10 | 30 | 14.00 | 8.67 |
| 30 | 2 | 10 | 30 | 14.65 | 4.99 |
| | 3 | 10 | 30 | 17.21 | 9.48 |
| | 1 | 10 | 50 | 17.99 | 11.20 |
| 60 | 2 | 10 | 50 | 18.34 | 11.48 |
| | 3 | 10 | 50 | 19.88 | 11.28 |
| | 1 | 10 | 80 | 18.14 | 12.84 |
| 90 | 2 | 10 | 80 | 19.23 | 12.41 |
| | 3 | 10 | 80 | 20.15 | 15.76 |
| | 1 | 10 | 100 | 20.50 | 15.49 |
| 120 | 2 | 10 | 100 | 21.74 | 16.98 |
| | 3 | 10 | 100 | 24.18 | 17.62 |

However, the results of the experiments on the 120 test problems of different sizes indicate that for 30, 60, 90 and 120-activity problems the proposed HSA is able to solve the problem with a rational improvement percentage under a time constraint. This shows that the proposed methodology is satisfactory.

## 6. Conclusions and Directions for Future Research

In this research, a class of project-scheduling problems called project scheduling with material ordering was investigated. The problem was formulated into a mixed-integer programming model. Since the problem was NP-hard, a hybrid simulated annealing and genetic algorithm approach was developed to solve it. The developed model in this paper is an extension of the PSMO problem investigated by Dodin and Elimam (2001),and develops a solution approach for small, medium, and large-scale problems. Because of the variety of the algorithm parameters, it was not economical to compare the performances regarding all parameter values. Therefore, we first made use of the Taguchi approach to calibrate them. Then, the proposed algorithm was tested on many problem instances whose mathematical models were solved by LINGO. The results of the comparison study showed the proposed methodology provides solutions as good as those obtained through LINGO and yet it takes shorter CPU times than LINGO. Further, while there were many instances for which LINGO was not able to reach the optimal solution in 3600 seconds, the proposed hybrid algorithm was able to solve them.

As a direction for future research, it may be interesting to apply some other meta-heuristics and compare them with the HSA developed in this research. Another clue for future research is the consideration of some other realistic assumptions such as supply lead-times for quantity order and renewable resource constraints. Another opportunity for research is the consideration of the problem with other optimization objectives such as minimization of total completion time, early and tardy penalties, or even multi-objective cases.

## 7. References


[1] Aquilano, N.J. and Smith, D.E.,(1980).A formal set of algorithms for project scheduling with critical path method - material requirements planning.Journal of Operations Management, 1,57-67.

[2] Dodin, B. and Elimam, A.A., (2001).Integrated project scheduling and material planning with variable activity duration and rewards. IIE Transactions, 33, 1005-1018.

[3] Karimi, N., Zandieh, M., and Najafi, A.A., (2010). Group scheduling in flexible flow shops: A hybridized approach of imperialist competitive algorithm and electromagnetic-like mechanism. International Journal of Production Research, 61, 1-13.

[4] Kirkpatrick, S., Gelatt, C. and Vecchi, M., (1983). Optimization by simulated annealing. Science, 220, 671-680.

[5] Montgomery, D.C., (2000).Design and analysis of experiments. 5th edition, John Wiley & Sons, New York.

[6] Najafi, A.A. and Niaki, S.T.A., (2006). A genetic algorithm for resource investment problem with discounted cash flows. Applied Mathematics and Computations, 183, 1057–1070.

[7] Phadke, M.S., (1989). Quality engineering using robust design. Upper Saddle River: Prentice-Hall, London.

[8] Ross, R.J., (1989). Taguchi techniques for quality engineering. McGraw-Hill, New York.

[9] Sajadieh, M.S., Shadrokh, S. and Hassanzadeh, F., (2009). Concurrent project scheduling and material planning: A genetic algorithm approach. ScientiaIranica, 16, 91-99.

[10] Shokrollahpour, E., Zandieh, M. and Dorri, B., (2009). A novel imperialist competitive algorithm for bi-criteria scheduling of the assembly flowshop problem. *International* Journal of Production Research, 52, 1-17.

[11] Smith-Daniels, D.E. and Aquilano, N.J., (1984). Constrained resource project scheduling subject to material constraints. Journal of Operations Management, 4, 369-388.

[12] Smith-Daniels, D.E. and Smith-Daniels, V.L., (1987a). Maximizing the net present value of a project subject to materials and capital constraints. Journal of Operations Management, 7, 33-45.

[13] Smith-Daniels, D.E. and Smith-Daniels, V.L., (1987b). Optimal project scheduling with materials ordering. IIE Transactions, 19, 122-129.