Multi-objective Simulated Annealing for a Maintenance Workforce Scheduling Problem: A case Study

Nima Safaei, Dragan Banjevic and Andrew K.S. Jardine
C-MORE Lab, Department of Mechanical and Industrial Engineering, University of Toronto Canada

1. Introduction

A multi-objective simulated annealing (MOSA) algorithm is described in this chapter to solve a real maintenance workforce scheduling problem (MWSP) aimed at simultaneously minimizing the workforce cost and maximizing the equipment availability. Heavy industry maintenance facilities at aircraft service centres, railroad yards and steel companies must contend with scheduling Preventive Maintenance (PM) tasks to ensure critical equipment remains available (Quan et al., 2007). PM tasks are labour intensive and the workforce that performs those tasks are often highly-paid and highly skilled with different proficiencies, which means the PM tasks scheduling should minimize the workforce costs. Therein lies a dilemma: a small labour force would help control costs, but a small labour force cannot perform many PM tasks per hour—and equipment that is not available does not generate revenue. A long completion time is not cost effective but neither is having too many workforce costs. A proper balance would minimize labour costs while simultaneously finishing all PM tasks in a timely manner. In other words, a trade-off must be made between the workforce costs and a timely completion of all PM tasks. Hence, in most real PM tasks scheduling problems, we encounter the multi-objective optimization.

There are very few previous papers focusing on the maintenance workforce scheduling problem. Higgins (1998) formulated the railway track maintenance crew problem as a mathematical program, and then used tabu search algorithms to solve the problem. Ahire et al., (2000) examined the utility of the evolution strategies to solve a MWSP with the aim of minimizing Makespan considering multiple-skills labour and workforce availability constraints. Yanga et al., (2003) formulated an airline maintenance manpower planning problem under a one week planning cycle considering various flexible strategies such as short-term or temporary contracts, trainee, part-time and subcontracted workers. They considered workforces with different types of skills that are grouped into a number of so-called "squads" with different numbers of members (or size). The objective was to minimize the total required manpower while satisfying the demand for every time slot. Quan et al., (2007) used the evolutionary algorithms to solve a multi-objective PM task scheduling problem with the aim of simultaneously minimizing workforce costs and Makespan. Workforce costs consist of the hiring cost of workers required to complete all PM tasks on time as well as the idle time cost. Makespan refers to the total amount of time it takes to

Source: Simulated Annealing, Book edited by: Cher Ming Tan, ISBN 978-953-7619-07-7, pp. 420, February 2008, I-Tech Education and Publishing, Vienna, Austria

complete all PM tasks. Notice that these two objectives are conflicting because minimizing the workforce increases the Makespan. They assumed that workers have two different skills, i.e., mechanic and electric and each worker can perform only one skill.

The rest of the chapter is organized as follows. Section 2 presents the problem description. In Section 3, the preliminary definition and concepts of the multi-objective optimization as well as MOSA's literature are presented. The MOSA to solve the considered problem is developed in Section 4. Experimental results are presented and discussed in Section 5. Finally, Section 6 mentions the conclusion and some future work.

2. Problem description

The considered problem is related to a steel company which has recently moved to a plant wide scheduling approach, through a central department, called Central Services (CS), to respond to the maintenance requirements of manufacturing areas or Business Units (BUs). The aim of this department is to minimize the workforce costs as well as avoid long-term disruptions and shutdowns of the critical equipments within BUs. Each BU schedules their work requests and then submits them to CS which attempts to schedule the workforce on those work requests to meet the needs across the plant. Work requests represent PM tasks to return the associated equipment to the as-good-as-new condition (throughout this paper, we use the phrase 'work request' or briefly 'work' and PM task interchangeably). Given the number and variety of the work requests, and the number of workers and the variety of their skills, the CS department has found it very difficult to optimally schedule works in a reasonable time.

The CS department satisfies labour requirements through internal and external resources, as regular time, overtime, and contract. The internal resource consists of a number of specialized groups with certain proficiency/skill for PM tasks, called field groups (FGs) such as mechanical, electrical, pipefitting and lubrication proficiencies. FGs are mobile groups, variable in size (number of members), which are responsible for PM/repair tasks at BUs. The external workforce is provided by contractors. Obviously, CS prefers to use the internal workforce in regular time and overtime (including weekends) and to use the contractors when they encounter the workforce shortage. CS manages the FGs to meet the demand of BUs, and supplements them with external forces. The PM schedule for each BU may be different for different periods depending upon the variety and failure nature of the existing assets and equipments. Thus, CS always encounters a new set of work requests in each period that must be scheduled, however, the required information of the work requests is known for CS in advance. In Figure 1, the relationship between CS, BUs and labour resources are shown schematically.

2.1 Mapping the MWSP as a generalized job shop scheduling problem

The MWSP can be considered as an extended job shop scheduling problem in which each FG represents a machine type and each work request represents a job with a number of operations that must be processed on the predetermined machines according to certain precedence relations. The capacity of machines is limited in the given planning horizon. Each job has a known ready/submission time and must be completed before its due date. The conflicting objectives are the workforce cost minimization versus the BU/equipment availability maximization. The workforce cost can be interpreted as machine operating/idle costs and the BU/equipment availability can be interpreted as the flow time of the associated job (see Section 2.2 for more details). A schematic mapping of MWSP into a job shop scheduling problem with 4 FGs and 5 work requests is shown in Figure 2. Symbol "W*i*" represents work *i*. Each work may be done by different FGs according to certain precedence relations.

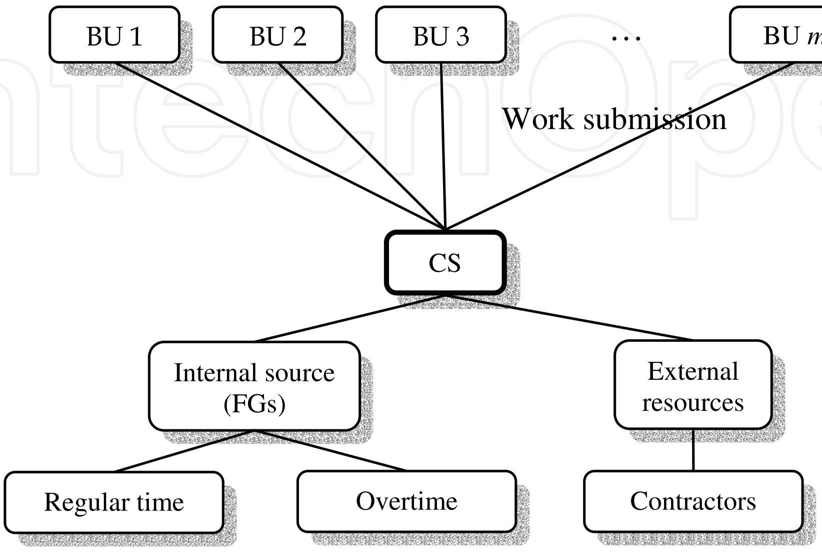


Fig. 1. Maintenance workforce management by Central Service

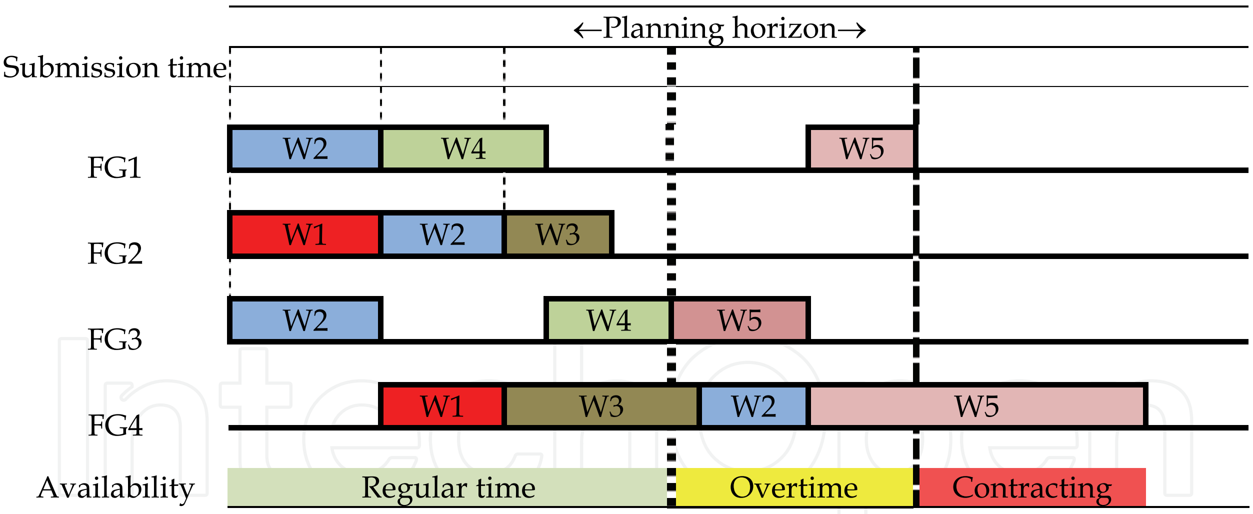


Fig. 2. Typical mapping of MWSP into job shop scheduling problem

A typical example of precedence relations associated with work 2 is shown in Figure 3. As shown in this figure, FGs 1 and 3 can operate simultaneously; however, both FGs are preceding operations for FG 2, and also FG 2 is a preceding operation for FG 4. From mathematical point of view, the precedence relations shown in Figure 3 can be presented as a 0-1 matrix as shown in Figure 4. As Figure 4 indicates, we need overtime for FGs 1, 3 and 4 to complete works 2, 3, and 5. Also, we need the external workforce as subcontracted workers for FG 4 to complete work 5. Moreover, the interference constraint between FGs causes some idle times during the operation time of FGs 1, 3 and 4.

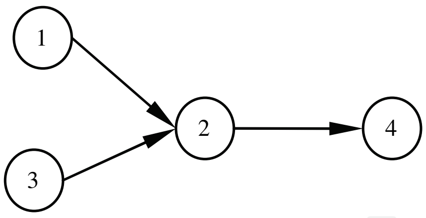


Fig. 3. Precedence relations between FGs for doing work 2

$$\begin{array}{c|cccc|}
\text{FG} & 1 & 2 & 3 & 4 \\
\hline
1 & 0 & 1 & 0 & 0 \\
2 & 0 & 0 & 0 & 1 \\
3 & 0 & 1 & 0 & 0 \\
4 & 0 & 0 & 0 & 0 \\
\hline
\end{array}$$

Fig. 4. Matrix form presentation for the precedence relations of Work 2

2.2 Scheduled and unscheduled shutdowns

As pointed out earlier, each work i is submitted to CS at time r_i and must be finished before due date d_i. r_i is typically called submission time or earliest start (ES) time, and d_i is typically called the latest request (LR) or latest finish (LF) date. After submission, the process of the work will start in s_i where $ES \leq s_i$ and completed in c_i, where $c_i \leq LF$. s_i is called the starting time, or "Time in", and c_i is called the completion time, or "Time out" of work i. Thus, the duration or processing time of work i is determined as s_i - c_i (see Figure 5). This duration is also known as scheduled shutdown in which the asset or equipment will not be available in interval [s_i, c_i]. However, sometimes an unscheduled shutdown is also considered for the work request which depends on the starting time of the work. Unscheduled shutdown is an approximated time interval that is estimated in terms of the magnitude of s_i. That is, by increasing s_i, the processing time of the work request (or equivalently the unavailability of the asset) will increase progressively because of the nature/mode of the failure. The unscheduled shutdown can be used to determine the importance degree (or weight) of the work request. The local objective of each BU is to minimize the flow time of corresponding work requests, i.e., to minimize $f_i = c_i – LE$. However, solely meeting this objective increases the workforce costs.

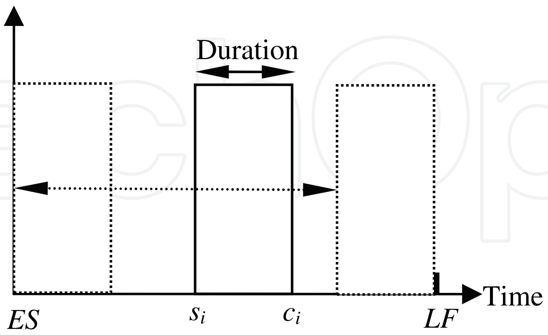


Fig. 5. Scheduled and Unscheduled Shutdowns

According to the above explanation, the MWSP considered in this study deals with two conflicting objectives:

1. Minimization of the total weighted flow time (TWFT) of works (BUs ultimate objective).
2. Minimization of the workforce costs (WfCs) consisting of fixed, overtime and contracting costs (one of the CS objectives)

2.3 Man-hour unit to measure the labour requirements

The processing time of a work request is often a function of the number of assigned workers. That is, by increasing the number of assigned workers, the processing time of the work decreases with a decreasing slope. In such cases, we need a proper unit to measure work done. Man-hour is a common time unit used in industry for measuring work. For example, if the size of a FG is 5 and the number of working hours per day is 8, then 5×8=40 man-hours are available per day for that FG. Thus, if a work request needs 10 man-hours, it can be done by one worker in 10 hours, or 2 workers in 5 hours, etc. Man-hour integrates the time and size of the labour requirements together. A number of studies can be found in which the labour requirements are estimated in terms of man-hour unit. For instance, in (Yanga et al., 2003), the maintenance department estimates that the short-term layover maintenance manpower demand in terms of man-hours, based on the available ground holding time slots, the different aircraft types, and the tasks required.

2.4 Assumptions

The assumptions of the problem can be summarized as follows:

1. The length of the planning horizon is fixed and the work requests submitted in the current planning horizon will be scheduled for succeeding planning horizon.
2. All work requests are submitted to CS during the current planning horizon with a known submission time.
3. The labour requirement and the processing time (duration) of each work request by each FG are known in advance.
4. Each work request has a known due date.
5. Each FG has a certain proficiency which is provided by the internal resources as regular and overtime, or the external resources as contract.
6. The number of members (size) of each FG in regular time, overtime and contacting is known in advance.
7. The labour requirement for work requests is measured in terms of the "man-hour" unit.
8. Workforce availability: The available man-hours for each FG as regular time, overtime and contract are known in advance.
9. Workforce costs consist of fixed cost, overtime cost and contracting cost per man-hour. Obviously, the unit cost of contracting is greater than one of overtime.
10. A fixed cost per man-hour is considered irrespective of the type of the workforce (i.e., internal or external). This cost can be interpreted as to include the transportation, tools, lunch and idle costs.
11. Each FG can operate only one work request at a time.
12. The scheduled shutdown of each work request is represented by its flow time. Flow time is defined as the difference between the completion time (time out) and submission time of the work request.
13. A weight is also associated with each work request which measures the importance degree of the work request. This weight is determined in terms of the unscheduled shutdowns of the work request.

After detailed explanation of the problem, it is worthwhile to briefly highlight how this study differs from previous works:

1. We consider the total weighted flow time instead of Makespan.
2. We consider the precedence relations between FGs to do a given work.

3. We consider workforces with different proficiencies, and overtime and subcontracted workers simultaneously.

2.5 Typical data set

For illustration, a typical data set with 10 work requests and 4 Field groups (mechanical, electrical, pipefitting and lubrication proficiencies) inspired by the real data is presented in this section. Consider a one-week planning horizon with 5 workdays and 2 holidays (weekend). Each workday consists of 8 hours regular time and 4 hours overtime and each holiday includes 4 hours overtime for each internal worker. Moreover, 4 hours in each workday is available for each subcontracted worker as an external labour. Subcontracted workers don’t work on weekends. Other information related to work requests and FGs are presented in Tables 1 to 3. The expected duration of each work request by each FG (in terms of man-hour), submission time and unscheduled shutdown (in terms of hours), and also the weight of work requests are shown in Table 1. Table 2 shows the workforce availability in regular time, overtime and contracting. In Table 3, the precedence relations between FGs associated to each work are shown (in all tables FG stands for field group).

Man-hour	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10
FG1	0	0	19	8	6	0	0	0	8	0
FG2	18	13	15	0	11	4	11	13	10	4
FG3	15	2	17	11	18	15	3	0	12	0
FG4	6	5	18	0	0	3	0	12	0	0
Submission time	57.6	43.3	16.21	49.82	4.86	31.49	8.73	39.17	1.94	45.32
Due date	170.86	109.78	214.26	115.12	117.11	102.81	99.94	119.49	112.41	113.51
Shutdown	0.21	0.36	0.12	0.37	0.21	0.34	0.26	0.3	0.22	0.35
Weight	0.58	0.98	0.33	1	0.58	0.92	0.72	0.81	0.59	0.96

Table 1. Work request Information

	Size			Cost per hour per man ($)			Availability per day per man (hours)		
	Regular time	Overtime	Contracting	Fixed Cost	Overtime	Contracting	Regular time	Overtime	Contracting
FG1	9	8	3	2	22	29	8	4	4
FG2	9	5	3	4	24	27	8	4	4
FG3	10	7	3	4	20	28	8	4	4
FG4	8	6	2	4	24	28	8	4	4

Table 2. Field group Information

FG	W1				W2				W3				W4				W5			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0
2	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0
3	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
4	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

FG	W6				W7				W8				W9				W10			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0
3	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 3. Precedence relations

3. Multi-objective simulated annealing

As depicted earlier, MWSP is an extended version of the job shop scheduling problem and obviously it is NP-hard and cannot be solved in a reasonable time using an exact approach for the real-sized problems. This reasoning was a motivation to develop a MOSA approach to solve the MWSP. In this section, the preliminary definitions and concepts of the multi-objective optimization are presented to illustrate the performance of the MOSA. Also, the MOSA's literature is completely reviewed.

3.1 Multi-objective optimization

In multi-objective optimization problems, we attempt to simultaneously optimize a number of conflicting objective functions in which the objectives are non-commensurable and the decision-maker has no clear preference for the objectives relative to each other. Without loss of generality, we will assume that all objectives are of the minimization type. A minimization multi-objective decision problem with K objectives is defined as follows: Given an n-dimensional solution space S of decision variables vectors X={x_1,…,x_n}, find a vector X^* that satisfies a given set of criteria depending on K objective functions Z(X)={Z_1(X),…Z_K(X)}. We wish to find an "ideal" vector X^* that minimizes all objective functions simultaneously which is usually not possible. The solution space S is generally restricted by a series of constraints, such as $g_j(X^*) = b_j$ for j= 1,…,m, and bounds on the decision variables. In many real-life problems, objectives under consideration conflict with each other. Hence, optimizing vector X with respect to a single objective often results in unacceptable results with respect to the other objectives. Therefore, a perfect multi-objective solution that simultaneously optimizes each objective function is almost impossible. A reasonable solution to a multi-objective problem is to investigate a set of solutions, each of which satisfies the objectives at an acceptable level, and without being dominated by any other solution. We summarize the multi-objective optimization area within the following definitions (Zitzler & Thiele., 1998):

- ***Dominant solution***: If all objective functions are used for minimization, a feasible solution *X* is said to dominate another feasible solution *Y* ($X \succ Y$), if $Z_i(X) \leq Z_i(Y)$ for $i=1,\ldots,K$ and $Z_i(X) < Z_i(Y)$ for at least one objective function *j*.
- ***Pareto optimal (Efficient) solution***: A solution is said to be Pareto optimal if it is not dominated by any other solution in the solution space. A Pareto optimal solution cannot be improved with respect to any objective without worsening at least one of other objective.
- ***Pareto optimal set***: The set of all feasible non-dominated solutions in *S* is referred to as the Pareto optimal set. For many problems, the number of Pareto optimal solutions is enormous (perhaps infinite).
- ***Pareto front***: For a given Pareto optimal set, the corresponding objective function vector values in the objective space are called the Pareto front.

The ultimate goal of a multi-objective optimization algorithm is to identify solutions in the Pareto optimal set.

3.2 Literature on MOSA

Numerous approaches have been developed in the literature with the aim of determining the Pareto optimal set using SA. A comprehensive review of SA based optimization algorithms to tackle multi-objective problems can be found in Suman & Kumar (2006). The first MOSA method has been proposed by Serafini (1992). The algorithm of the method is almost the same as the algorithm of single objective SA. The method uses a modification of the acceptance criteria of solutions in the original algorithm. Various alternative criteria have been investigated in order to increase the probability of accepting non-dominated solutions. A special rule given by the combination of several criteria has been proposed in order to concentrate the search almost exclusively on the non-dominated solutions.

Suppapitnarm & Parks (1999) proposed a multi objective SA method, namely Suppapitnarm-MOSA, in which only one solution is used and the annealing process adjusts each temperature independently according to the performance of the solution in each criterion during the search. The concept of archiving the Pareto optimal solutions with SA has been initially used by Suppapitnarm et al., (2000). In their study, an archive set stores all the non-dominated solutions between each of the multiple objectives. A new acceptance probability formulation based on an annealing schedule with multiple temperatures (one for each objective) has also been used. The acceptance probability of a new solution depends on whether or not it is added to the set of potentially Pareto-optimal solutions. If it is added to this set, it is accepted to be the current solution with probability equal to one. Otherwise, a multi-objective acceptance rule is used.

Ulungo et al., (1999) proposed another MOSA method in which for a multi-objective problem, a move from the present position to a new position can result in three different possibilities:

a) Improving moves with respect to all objectives is always accepted with probability one.
b) Simultaneous improvement and deterioration with respect to different objectives. In this case neither the new move nor the current solution dominate. Therefore, the strategy devised must be sound enough to discriminate between the new and the current solutions.
c) Deterioration with respect to all objectives is accepted with a given probability.

Their method uses a strategy called the *criterion scalarizing strategy* since probability to accept the new solution must take into account the distance between the old and the new move. This strategy maps the multi-dimensional criteria space into a one-dimensional space. Thus, this strategy works with a predefined diversified weight vector. This set of weights is uniformly generated. Two scalarizing functions have also been used: the weighted sum of objectives and the Chebyshev norm (Teghem et al., 2000).

Czyzak, & Jaszkiewicz (2000) proposed a MOSA approach by combining SA with a genetic algorithm (GA). This method uses the concept of neighborhood, acceptance of new solutions with some probability and annealing schedule from SA and the concept of using a sample population of interacting solutions from GA. Their method uses scalarizing functions based on probabilities for accepting new solutions. In each iteration of the procedure, a set of solutions called generating samples controls the objective weights used in the acceptance probability. This assures that the generating solutions cover the whole set of efficient solutions. One can increase or decrease the probability of improving values of a particular objective by controlling the weights. The higher the weight associated with a given objective, the lower the probability of accepting moves that decrease the value of this objective and the greater the probability of improving the value of this objective.

Suman (2002) proposed a MOSA approach to tackle the constraint violations. The proposed MOSA attempts to handle constraints within its main algorithm by using a weight vector in the acceptance criterion by directing the move towards the feasible solutions. It does not use any extra techniques such as the penalty function approach to handle constraints. It has been shown that the substantial reduction in computational time can be achieved without worsening the quality of solution with this method. The weight vector depends on the number of constraints violated by the given solution and the objective function. Suman (2005) proposed a MOSA approach using Pareto-domination-based acceptance criterion. He uses an idea that a strategy of Pareto-domination based fitness can easily be adapted to simulate annealing in the acceptance criterion. Here, fitness of a solution is defined as one plus the number of dominating solutions in Pareto-optimal set (containing both feasible as well as infeasible solutions). The larger the value of fitness, the worse the solution. Initially, the fitness difference between the current and the generated solution is small and the temperature is high so any move is accepted. This gives us a way to explore the full solution space. As the number of iterations increases, temperature decreases and fitness difference between the current and generated solutions may increase. Both make the acceptance move more selective and it results in a well-diversified solution in true Pareto-optimal solutions.

Most of the proposed MOSA approaches, except Suppapitnarm-MOSA, use a kind of scalarizing function for combining the objectives into a weighted summation term as fitness/energy function to evaluate the solutions. However, it is unclear how to choose the weights in advance. Indeed, one of the principal advantages of multi-objective optimization is that the relative importance of the objectives can be decided with the Pareto front on hand. To overcome this disadvantage, Smit et al., (2004) proposed a dominance based energy function. According to this function, the energy value of solution x is equal to the cardinality of set $F_x \subset F$ where F is the best Pareto front obtained so far (archive of the estimated Pareto front) and subset F_x contains all solutions belong to F that dominate x. This function ensures that the new solutions that move the estimated front towards the true (ultimate) Pareto front are always accepted. As the authors claim, a benefit of this energy function is that it encourages exploration of sparsely populated regions of the front.

However, the performance of this function highly depends on the cardinality of set *F*. That is, when *F* is small the resolution in the energies can be very coarse, leading to a low resolution in acceptance probabilities. To overcome this disadvantage, they artificially increased the size of *F* using three methods: conditional removal of dominated points, linear interpolation and attainment surface sampling.

4. MOSA to solve MWSP

The consideration of precedence relations in addition to interference relations causes the size of the feasible space to decrease; however, it doesn't mean the Pareto optimal set will be achieved simply. Contrariwise, the ultimate Pareto optimal set will be difficult to access, especially when the size of the problem increases. In this case, the population-based algorithms such as Genetic Algorithms lead to infeasible solutions most of the time. This reasoning became a motivation to select a single solution-based meta-heuristics such as SA to solve the considered problem.

In our opinion, the method proposed by Suppapitnarm et al., (2000) is one of the best in the context of the MOSA. In this method, we don't need to determine a weight for each objective function while all objectives affect the acceptance probability of the non-improver solutions. Moreover, a new solution is accepted if it can be added to the best Pareto archive set obtained so far. This strategy guarantees the continuous improvement of the current Pareto front toward the ultimate one. Thus, we use Suppapitnarm-MOSA to solve the MWSP. The specialization of the Suppapitnarm-MOSA to solve the MWSP is presented in the following subsections, using the nomenclature presented in the Appendix.

4.1 Initial Temperature

According to the fundamental concepts of SA, non-improver solutions are accepted in the primary iterations with high probability. Thus, we set the initial temperature (for each objective) in such a way that the non-improver solutions are accepted with a probability of about 95 percent in the primary iterations. The related pseudo code is shown in Figure 9 (Safaei et al., 2008). Parameter *Q* represents the number of samples.

```
Sub Initial_Temperature( )
 For $k$=1 to $K$
   For $q$=1 to $Q$
     Do
       Generate two solutions $X_1$ and $X_2$ at random
       LOOP UNTIL $(Z(X_1) \neq Z(X_2))$
       Set $T_q^0 = \frac{|Z_k(X_1) - Z_k(X_2)|}{-\ln(0.95)}$
     Next $q$
   Set $T_0^k = (1/Q)\sum_{q=1}^{Q} T_q^0$
 Next $k$
End Sub
```

Fig. 9. Pseudo code of the initial temperature generation subroutine

4.2. Solution representation

The main objective of the MWSP in this study is to determine the sequence of work requests (works, for short) that must be done by each FG in such a way that some objectives are optimized. Thus, the solution representation must determine the sequence of operating works for each FG. To this propose, we consider a matrix consisting of *K* rows (number of FGs) and *M* columns (number of works) to represent the solution to the MWSP. The solution representation is shown in Figure 10 for typical solution $X=[x_{i[j]}]_{K\times M}$ where $x_{i[j]}=w$ means that work *w* must be scheduled on j^{th} position (i.e., [*j*]) in the sequence of works associated with FG *i*. It should be noted that some of the entries in the solution representation are inherently zero/null because all works need not be done by all FGs. For more clarity, an example solution related to the data set presented in section 2.5 is shown in Figure 11.

$$\begin{array}{r|cccccc|}
 & W[1] & W[2] & \dots & W[j] & \dots & W[M] \\
\hline
FG1 & x_{1[1]} & x_{1[2]} & & x_{1[j]} & & x_{1[M]} \\
FG2 & x_{2[1]} & x_{2[2]} & & x_{2[j]} & & x_{2[M]} \\
\vdots & & & & & & \\
FGi & x_{i[1]} & x_{i[2]} & & x_{i[j]} & & x_{i[M]} \\
\vdots & & & & & & \\
FGK & x_{K[1]} & x_{K[2]} & & x_{K[j]} & & x_{K[M]} \\
\hline
\end{array}$$

Fig. 10. Solution Representation

	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10
FG1	4	5	3	9	-	-	-	-	-	-
FG2	5	8	2	10	6	7	3	9	1	-
FG3	4	5	7	3	6	2	9	1	-	-
FG4	3	8	1	6	2	-	-	-	-	-

Fig 11. An example solution for data set presented in section 2.5

4.3 Initial solution generation

In general, for better exploration of the feasible space, the initial solution is generated at random. However, as discussed in Section 2, MWSP is actually a generalized job shop scheduling problem with precedence constraints in addition to interference constraints inherently embedded in the scheduling problems. Thus, the generating of a random solution which simultaneously satisfies both precedence and interference constraints is one of the most important portions of this research that makes it different from workforce scheduling problems described in the literature. In this case, the applied approach for generating the initial solution must maintain the CPU time on an acceptable level and use advantages of the random generation.

To overcome this drawback, we introduce a recursive-sequential approach in which at each iteration *i*, the sequence of works corresponding to FG*i* is randomly generated considering the history of assignments in previous FGs 1,…, *i*-1 as well as the precedence relations. The recursive procedure verifies the feasibility of the current assignment. This procedure uses the information given in the matrix $S=[s_{il}]_{K\times K}$ where $S=R\oplus R^2\oplus\ldots\oplus R^{K-1}$, in which $s_{il}\in\{0,1\}$, $R=[r_{il}]_{K\times K;}\ r_{il}\in\{0,1\}$ is the precedence relation matrix for a given work (see Figure 3) and ⊕ represents the Boolean summation operator (Seyed-Hosseini et al., 2006). Matrix *S* consists of all direct and indirect precedence relations between FGs to do a given work. In other words, s_{il} =1 means FG*i* is prior to FG*l* directly (*i*→*l*) or indirectly (*i*→…→*l*). This recursive procedure prevents the creation of the infinite loop during the sequential assignment process. An infinite loop is a sequence of the precedence and interference relations that loops endlessly. A typical example of the infinite loop is shown in Figure 12. In this figure, both works A and B must be done by both FGs *i* and *l*. However, FG*l* is directly prior to FG*i*, i.e., $l \to i$, for work A and contrariwise FG*i* is indirectly prior to FG*l* for work B, i.e., $i \to k \to l$, where $i<l<k$. Assume that the sequence of works for FG*i* is already created according to the sequential phase of the approach. Moreover, the sequence of works for FG*l* is being preceded and for FG*k* has not created yet. Currently, work A is randomly selected and would be scheduled immediately after work C on FG*l*. We want to check the feasibility of this assignment. The completion time of work A on FG*l* is obtained as $ct_{lA} = pt_{lA} + ct_{lC}$. Without loss of generality, we define the term $a \subset b$ that means for obtaining parameter *a*, parameter *b* must already be determined. Thus, we have $ct_{lA} \subset ct_{lC}$. Using the backward recursive algorithm, the following infinite loop is obtained: $ct_{lC} \subset ct_{lB} \subset ct_{iB} \subset ct_{iA} \subset ct_{lA} \subset ct_{lC}$. Thus, the assignment presented in Figure 11 is infeasible. Consequently, the proposed approach doesn't allow that work A is scheduled after work B on FG*l* and so it must be scheduled before work B.

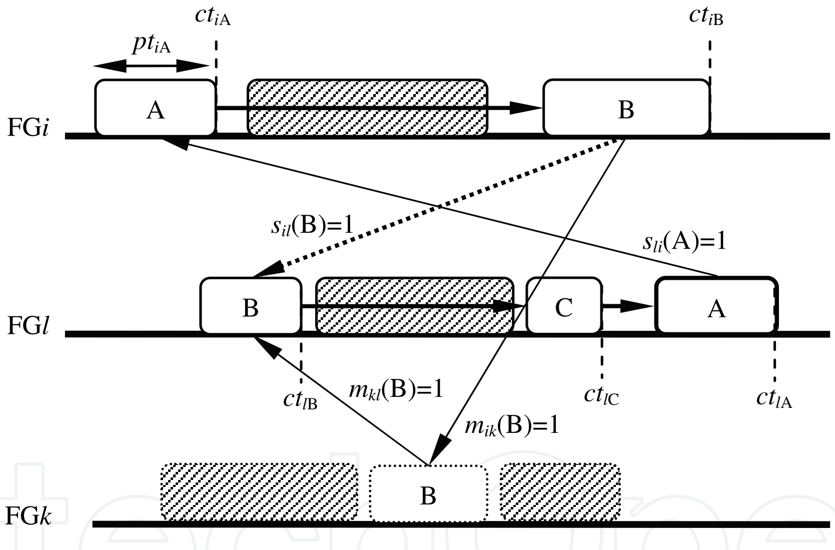


Fig 12. Typical infinite loop

As an example, according to the precedence relations given in Table 3, for work 3, we have (1→3) and for work 4, we have (3→1). Assume that works 3 and 4 are swapped together on FG 3 in the solution presented in Figure 11. Thus, we encounter an infinite loop as: $ct_{34} \subset ct_{37} \subset ct_{35} \subset ct_{33} \subset ct_{13} \subset ct_{15} \subset ct_{14} \subset ct_{34}$. Thus, work 4 cannot be scheduled anywhere after work 3 on FG 3, if the sequence of works for FG 1 has already been fixed.

4.4 Neighbourhood solution generation

The swapping adjacent pair method is used to generate the neighbourhood solutions. At first, two adjacent works on a FG are randomly selected and then are swapped together. The feasibility of this change is checked by the recursive procedure explained in the previous section.

A stochastic sampling scheme of size 1000 within the objective function space is used to verify the efficiency of the applied strategy. The scatter diagram corresponding to this sampling is shown in Figure 13. Point A represents the initial solution and other points are generated by the swapping adjacent pair method. Point B is associated with the best obtained solution. As it can be seen in this figure, this method can correctly navigate the solution space. Our reason for it is that the generated solutions have an improvement trend in terms of the objective function values as the best obtained solution (Point B) improves each objective function by about 50% compared with the initial solution (Point A). It should be noted that this sampling is completely random, without using an improvement criterion. In other words, the results indicate that the probability of the improver movements is significantly greater than non-improver ones and hence the strategy used is a proper one to explore the solution space.

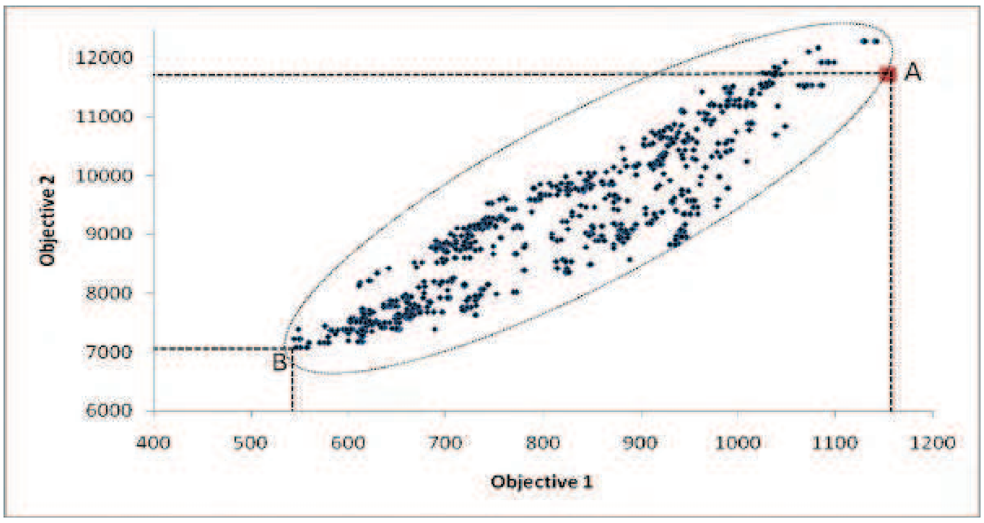


Fig 13. Scatter diagram related to the stochastic sampling of the neighbourhood solution generation method

4.5. Cooling schedule

The classical cooling schedule of SA is used for each temperature (one for each objective) as $T_t^k = \alpha T_{t-1}^k$ where α is the cooling rate or decrement factor and k=1,2.

4.6 Fitness function

As mentioned in Section 3, the MWSP involves two conflicting objectives: minimizing the TWFT versus minimizing the WfC. Even though, the generated solutions satisfy the precedence and interference constraints, there are still two restrictions which must be considered by the generated solutions. These two restrictions are the due date of the works and the workforce resource limitations in regular time, overtime and contracting. To this end, we consider two penalty functions, one for each objective. The first penalty function (PF_1) that is added to the first objective as Z_1=TWFT+λPF_1 penalizes the solutions violating the due date of some works. Parameter λ represents the penalty coefficient which is a large positive number. Likewise, the second penalty function (PF_2) that is added to the second objective as Z_2=WfC+λPF_2 penalizes the solutions violating the workforce limitations. These penalty functions lead the infeasible solutions toward feasible space. The mathematical expressions of the obtained fitness functions are given in Eq. (1) and (2):

$$Z_1 = \min \sum_{m=1}^{M} w_m \left(rw_m - r_m\right) + \lambda \sum_{m=1}^{M} \max\left\{rw_m - d_m, 0\right\}, \tag{1}$$

$$\begin{aligned} Z_2 = \min \sum_{k=1}^{K} \Big(& c_k rf_k + c'_k \max\left\{\min\left(rf_k, s_k h_k + s'_k h'_k\right) - s_k h_k, 0\right\} \\ & + c''_k \max\left\{\min\left(rf_k, s_k h_k + s'_k h' + s''_k h''_k\right) - s_k h_k - s'_k h'_k, 0\right\}\Big) \\ & + \lambda \sum_{k=1}^{K} \max\left\{rf_k - \left(s_k h_k + s'_k h' + s''_k h''\right), 0\right\}, \end{aligned} \tag{2}$$

The release time of work *m* is recursively computed as follows:

$$rw_m = \max_{1 \le k \le K}\left\{ct_{km}\right\};\ ct_{km} = pt_{km} + \max\left\{\max_{l;\, p_{mlk}=1}\left\{ct_{lm}\right\}, ct_{kn}, r_m\right\},\ m = 1, 2, \dots, M, \tag{3}$$

where *n* represents the work that must be scheduled immediately before work *m* for FG *k*. Initial values are $ct_{k1} = pt_{k1}$ for each *k*.

4.7 Acceptance strategy

Similar to the SMOSA, an archive set stores all the non-dominated/Pareto solutions between each of the multiple objectives. The acceptance probability of a new solution depends on whether or not it is added to the set of potentially Pareto-optimal solutions. If it is added to this set, it is accepted to be the current solution with probability equal to one. Otherwise, it is accepted with the following probability.

$$p = \min\left\{1, \exp\left(\frac{-\Delta Z_1}{T_1}\right) \times \exp\left(\frac{-\Delta Z_2}{T_2}\right)\right\}. \tag{4}$$

In Eq. (4), $\Delta Z_k = Z_k(Y) - Z_k(X)$ in which X is the current solution and Y is a neighbourhood solution resulting from X using the neighbourhood solution generation method.

4.8 Stoppage criteria

The MOSA algorithm is stopped when one of the following criteria is satisfied:

1. Maximum number of consecutive temperature trails (*R*).
2. Minimum allowable value of temperatures (final temperature) (T_f).
3. Maximum elapsed time after the last updating of Pareto archive set ($t_{\max}$).

4.9 Lower bounds for objective functions

As mentioned before, a large amount of time is needed to obtain the Pareto optimal set for MWSP. Hence, due to unavailability of Pareto optimal set for comparison and having an idea about the quality of the obtained Pareto solutions, a traditional methodology is to compare the lower bound of the objective functions with the obtained Pareto front. In this study, we use two different methods to obtain the lower bound of TWFT and WfC. The lower bound of the first objective function, *TWFT*$_{\text{LB}}$, is computed assuming the processing of each work by each FG is started immediately after submission or equivalently after completion by the preceding FGs (no-wait strategy). The mathematical expression of *TWFT*$_{\text{LB}}$ is given in Eq. (5):

$$TWFT_{LB} = \sum_{m=1}^{M} w_m \left(rw_m^{LB} - r_m \right),$$

$$rw_m^{\text{LB}} = \max_{1 \le k \le K} \left\{ ct_{km}^{\text{LB}} \right\};\ ct_{km}^{\text{LB}} = pt_{km} + \max \left\{ \max_{l:\, p_{mkl}=1} \left\{ ct_{lm}^{\text{LB}} \right\}, r_m \right\}, \tag{5}$$

where ct_{km}^{LB} represents the possible earliest time of completing processing of work *m* by FG*k*. The possible earliest time occurs when the processing of work *m* is started immediately after submission by BU or completion by the preceding FGs. Initial values are $ct_{k1}^{LB} = pt_{k1}$ for each *k.* The lower bound for the second objective function, *WfC*$_{\text{LB}}$, is computed according to the FIFO strategy in which the works are scheduled for each FG in increasing order of their arriving times. In this case, FGs are scheduled independently as a single-machine scheduling problem.

Although, the solutions under the obtained lower bounds are not necessarily feasible, the obtained lower bounds can be considered as a criterion to measure the goodness of the obtained Pareto front. In this case, we say the performance of the solution method is acceptable, if under the same conditions, the relative gap (distance) between lower bounds and obtained Pareto front is relatively small or at least does not increase significantly, while the size of the problem increases. It is worth noting that the difference between *TWFT*$_{\text{LB}}$ and its optimal value will increase while the size of the problem increases. It is because the precedence relations cause the waiting time of the in-process works to increase significantly.

5. Computational results

In this section, we verify the performance of the developed MOSA to solve the MWSP using a number of numerical examples. Numerical examples are inspired by the real data and generated randomly in pre-defined intervals. Ten numerical examples with 10, …, 100, works and 4 FGs are generated and solved by the developed MOSA. The details of these examples are not given here. The number of FGs is constant for all problems, as in the real case. MOSA is developed by Visual Basic 2008 on an x64-based multi-processor personal computer with 8 Intel Xeon processors and 2 GB memory. Each numerical example is solved 10 times and the best Pareto solutions obtained are reported and then the corresponding Pareto front is compared with the lower bound of the objectives. The parameter setting of the developed MOSA is shown in Table 4. For tuning the MOSA's parameters, some examples with different sets of parameters were solved. In the end, we found that the following parameter setting was effective to solve the MWSP. As it is evident from Table 4, parameters *N* and *R* are considered as linear functions in terms of the problem size.

Parameter	Q	α	N	R	T_f	$t_{\max}$	λ
Value	$M/2$	0.95	$10M$	$10M$	0.01	120 Sec.	10

Table 4. MOSA parameter setting

In the first step, the numerical example presented in Section 2.5 is solved and the best Pareto solutions are reported in Table 5. The average and standard deviation (SD) of *TWFT* and *WfC* values associated with the obtained Pareto solutions are also presented in this table. As it is evident from Table 5, the small values of SD imply that the algorithm converges to a small region of the objective space. That means that the distance between the obtained Pareto solutions is insignificant and the solutions have a relatively identical importance degree from the decision making point of view. The small values of SD can be the necessary condition for efficiency of the proposed method. However, the sufficient condition for efficiency is that the ultimate/optimal Pareto front is also in this small region. This issue will be discussed in below this section. For more clarity, the Pareto front associated with the Pareto set indicated in Table 5 is shown in Figure 14.

Pareto No.	*TWFT*	*WfC*
1	244.17	1531.48
2	259.76	1441.06
3	268.88	1433.06
4	247.18	1491.48
5	251.96	1453.36
6	265.99	1435.16
7	248.95	1469.92
8	254.39	1443.16
Average	255.16	1462.33
S.D	8.94	34.19

Table 5. Best Pareto solutions associated with the data set from Section 2.5

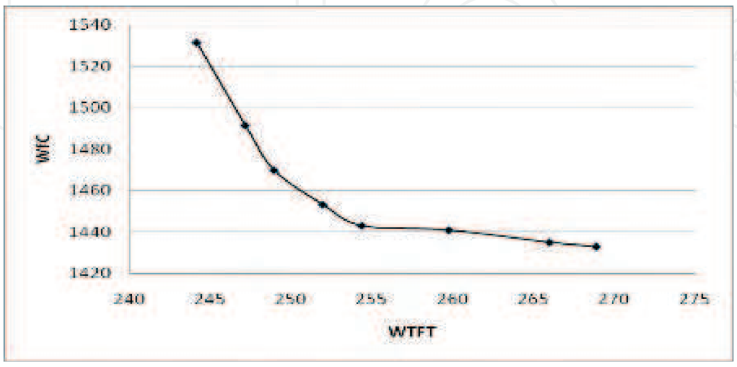


Fig 14. Pareto front associated with the Pareto set indicated in Table 5

Likewise, the information related to the obtained Pareto solutions and Pareto front for the numerical example with 20 works, i.e., 20×4, is provided in Table 6 and Figure 15. The same reasoning applicable to the first test problem is also applicable to the second one.

Pareto No.	*TWFT*	*WfC*
1	147.77	1173.10
2	148.31	1171.42
3	149.30	1171.42
4	149.90	1157.10
5	151.10	1139.42
6	152.63	1129.74
7	155.84	1127.42
8	156.13	1123.58
9	157.86	1121.26
10	158.09	1117.26
11	158.66	1115.34
Average	153.24	1140.64
S.D	4.18	23.16

Table 6. Best Pareto solutions associated to problem 20×4

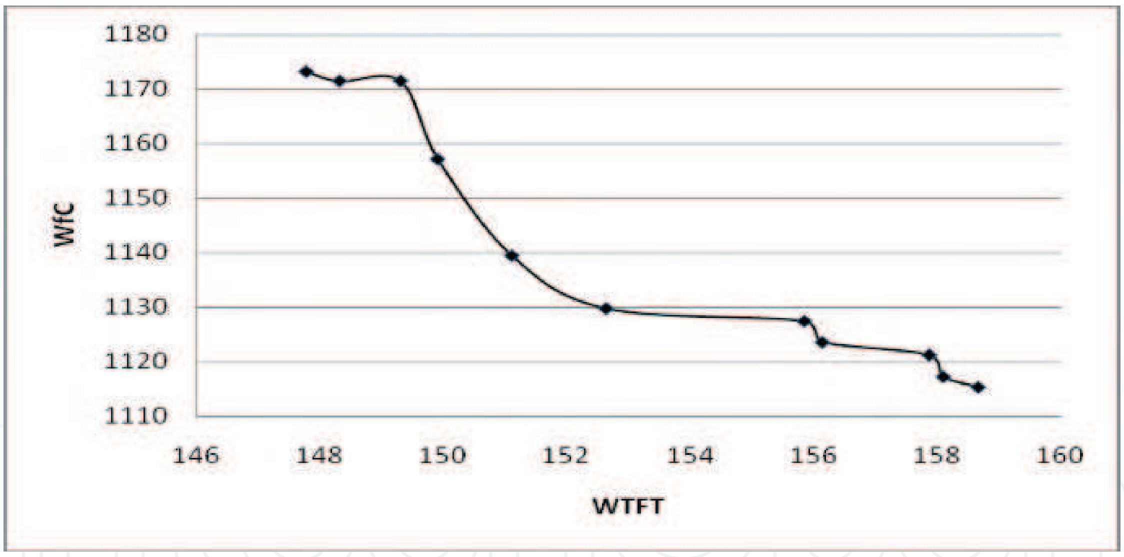


Fig 15. Pareto front associated with the Pareto set indicated in Table 6

The obtained results associated with the different real-sized problems are summarized in Table 7 in terms of the mean of objective values, i.e., $TWFT_M$ and WfC_M, corresponding to Pareto solutions, lower bounds, CPU time, and relative gaps. The relative gap between $TWFT_M$ and $TWFT_{LB}$ is computed as their ratio. The relative gap between WfC_M and WfC_{LB} is computed as the relative difference between WfC_M and WfC_{LB}, that is $[(WfC_M-WfC_{LB})/WfC_{LB}]\times100$. As shown in Table 7, by increasing the size of the problems, TWFT_Gap doesn't necessarily increase. Moreover, WfC_Gap is significantly small, which means that the obtained WfC_M values are very close to the optimal ones. Thus, according to the discussion presented in Section 4.9 and earlier in this section, we can conclude the developed MOSA is a proper and robust approach to solve the considered MWSP. The trend of the CPU time shown in Figure 16 can be estimated by the formula $CPUtime$=51.21M^2-216M+400, with R^2= 0.97, which means the developed MOSA algorithm is of a polynomial order, with a complexity degree $O(M^2)$.

Test Problem		Objective mean		Lower bound		CPU time (Sec.)	Gap	
Size	Planning cycle (week)	$TWFT_M$	WfC_M	$TWFT_{LB}$	WfC_{LB}		$\frac{TWFT_M}{TWFT_{LB}}$	WfC (%)
10×4	1	255.16	1462.33	161.39	1321	34	1.58	10.69
20×4	1	153.24	1140.64	64.56	1080.70	252	2.37	5.54
30×4	1	637.53	1065.14	95.5	1005.84	329	6.67	5.89
40×4	2	173.72	2064.66	66.9	2062.5	528	2.59	0.10
50×4	2	186.97	1699.55	67.48	1680.34	648	2.77	1.14
60×4	2	179.86	1820.8	59.17	1800.88	978	3.03	1.10
70×4	2	572.18	2193.2	92.31	2117.34	1136	6.19	3.58
80×4	2	248.07	2047.41	34.66	1991	1866	7.15	2.83
90×4	2	280.09	2618.21	34.69	2568.7	2416	8.07	1.92
100×4	2	386.45	1614.55	59.89	1539	3612	6.45	4.90

Table 7. Comparison between Pareto fronts and lower bound values

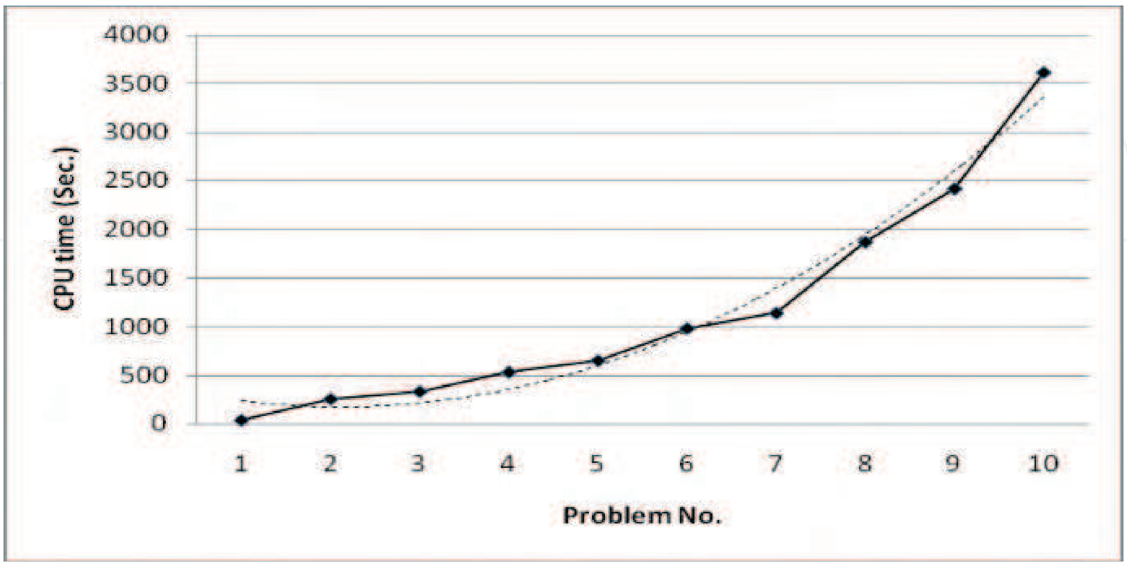


Fig 16. Trend of CPU times according to the information provided in Table 7

6. Conclusion

In this chapter, we proposed a multi-objective simulated annealing (MOSA) algorithm to solve a real maintenance workforce scheduling problem (MWSP) with the aim of simultaneously minimizing the workforce cost and the flow time of the work requests. The latter objective is equivalent to the maximization of the equipment availability because by increasing the flow time of a work request the unscheduled shutdown of the corresponding asset will increase too. Workforces have different proficiencies and are grouped into a number of teams called "Field Groups" (or FG for short). Labour requirements are provided from internal and external resources as regular time, overtime and contract.

We use a MOSA algorithm introduced in the literature namely Suppapitnarm-MOSA to solve the MWSP. In this method, an archive set stores all the non-dominated/Pareto solutions between each of the multiple objectives. The acceptance probability of a new solution depends on whether or not it is added to the set of potentially Pareto optimal set. However, all objectives affect the acceptance probability of a non-improver solution. The developed MOSA uses the swapping adjacent pair strategy to explore the feasible solution.

One of the main differences between the current study and previous ones is that we consider the precedence relations between FGs to do a given work request, in addition to the traditional interference relations between work requests that must be scheduled for a given FG. This extra assumption is a big obstacle to generating the feasible or neighbourhood solutions. Hence, the single solution-based meta-heuristics such as SA or Tabu search seem to be the unique alternatives to solve this problem. This is because population-based operators, such as crossover in Genetic Algorithm, lead to infeasible solutions most of the time.

To overcome this drawback, we introduce a recursive-sequential approach to construct the sequence of works for each FG with the aim of identifying the infinite loops resulting from consecutive interference and precedence relations.

Because the Pareto optimal set cannot be obtained in real-sized problems, a lower bound was developed separately for each objective function and the obtained Pareto front is compared with these lower bounds.

The obtained results show that the developed MOSA is a robust method to solve the MWSP. Our reasoning is that the developed MOSA always converges to a small region of the feasible space, very close to the lower bound of one of the objective functions while the relative difference between the obtained results and the lower bound of another objective function doesn't increase significantly when the size of the problem increases.

7. References

Ahire, S.; Greenwood, G.; Gupta, A. & Terwilliger, M. (2000). Workforce-constrained preventive maintenance scheduling using evolution strategies, *Decision Science Journal*, vol. 31 (4), pp. 833–859.

Czyzak, P. & Jaszkiewicz, A. (1998). Pareto simulated annealing - a metaheuristic technique for multiple-objective combinatorial optimization, *Journal of Multi-Criteria Decision Analysis*, vol. 7, pp. 34-47.

Quan, G.; Greenwood, G.W.; Liu, D. & Hu, S. (2007). Searching for multiobjective preventive maintenance schedules: Combining preferences with evolutionary algorithms, European Journal of Operational Research, vol. 177, pp. 1969–1984.

Safaei, N.; Saidi-Mehrabad, M. & Jabal-Ameli, M.S. (2008). A hybrid simulated annealing for solving an extended model of dynamic cellular manufacturing system, *European Journal of Operational Research*, vol. 185 (2), pp 563-592.

Serafini, P. (1992). Simulated annealing for multiple objective optimization problems, *Proceedings of the Tenth International Conference on Multiple Criteria Decision Making*, pp 87–96, Taiwan, 19–24 July 1, 1992, Taipei.

Seyed-Hosseini, S.M.; Safaei, N. & Asgharpour, M.J. (2006). Reprioritization of failures in a system failure mode and effeCS analysis by decision making trial and evaluation laboratory technique, *Reliability Engineering and System Safety*, Vol. 91(8), pp. 872-881.

Smith, K.I.; Everson, R.M. & Fieldsend, J.E. (2004). Dominance measures for multi-objective simulated annealing, *Congress on Evolutionary Computation (CEC2004)*, vol. 1, pp.23 – 30.

Suman, B. (2002). Multiobjective simulated annealing—a metaheuristic technique for multiobjective optimization of a constrained problem, *Foundations of Comput Decision Science*, vol. 27, pp. 171–191.

Suman, B. (2005). Self-stopping PDMOSA and performance measure in simulated annealing based multiobjective optimization algorithms, *Computs and Chemical Engineering*, vol. 29, pp. 1131–1147.

Suman, B. & Kumar, P. (2006). A survey of simulated annealing as a tool for single and multiobjective optimization, *Journal of the Operational Research Society*, vol. 57, pp, 1143–1160.

Suppapitnarm, A. & Parks, T. (1999). Simulated annealing: an alternative approach to true multiobjective optimization. *Proceeding of Genetic and Evolutionary Computation Conference. Conference Workshop Program*, pp 406–407, Florida, Orlando.

Suppapitnarm, A.; Seffen, K.A.; Parks, G.T. & Clarkson., P.J. (2000). A simulated annealing algorithm for multiobjective optimization, *Engineering Optimization*, vol. 33. pp. 59-85, 2000.

Teghem, J.; Tuyttens, D. & Ulungu, E.L. (2000). An interactive heuristic method for multiobjective combinatorial optimization, *Journal of Comput and Operations Research*, vol. 27, pp. 621–634.

Ulungu, E.L.; Teghaem, l.; Fottemps, Ph. & Tuyttens, D. (1999). MOSA method : a tool for solving multiobjective combinatorial decision problems, *Journal of multi-criteria decision analysis*, vol. 8. pp. 221-236.

Zitzler, E. & Thiele, L. (1998). Multiobjective optimization using evolutionary algorithms: a comperative case study. In: Eiben., AE., Back, T., Schoenauer, M. & Schwefel, H.P. (eds). Parallel Problem Solving from Nature V. Springer, Berlin, Germany, pp 292–301.

Yanga, T-H.; Yanb, S. & Chen H-H. (2003). An airline maintenance manpower planning model with flexible strategies, *Journal of Air Transport Management*, vol. 9, pp. 233–239.

Appendix: Nomenclature

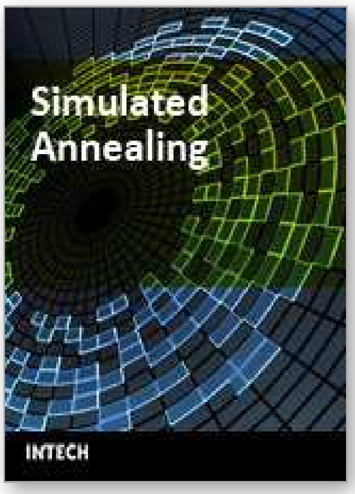
MWSP:

M	number of work requests (m=1,2,…,M)
K	number of FGs (k = 1,2,…,K)
r_m	submission (ready) time of work m
pt_{mk}	man-hours required for FGk to process work m. This parameter is interpreted as the duration or processing time of work m by FGk
a_{mk} =1	if work m must be operated by FGk; and =*0* otherwise
p_{mkl} =1	if FGl must operate immediately before FGk on work m; and =0 otherwise (precedence relations)
h_k	hours available for FGk in regular time during the planning horizon
h'_k	hours available for FGk in overtime during the planning horizon
h''_k	hours available for FGk in contracting time during the planning horizon
s_k	size of FGk in regular time during the planning horizon
s'_k	size of FGk in overtime during the planning horizon
s''_k	size of FGk as contract during the planning horizon
c_k	fixed cost of FGk per hour
c'_k	unit cost of FGk per hour in overtime
c''_k	unit cost of FGk per hour in contracting time
w_m	weight (or importance degree) of work request w. We assume that $w_m = \tau_m / \max_m\{\tau_m\}$, where τ_m represents the unscheduled shutdown of work m
ct_{mk}	completion time of work m by FGk
rw_m	release time of work m. The difference between rw_m and r_m is interpreted as shutdown of work m
rf_m	release time of FGk

MOSA:

α	rate of cooling (decrement factor)
T_0^k	initial temperature for objective k
T_t^k	system temperature in iteration t associated with objective k
T_f	final temperature

$Z_k(X)$	value of objective function *k* (or fitness function) for solution *X*. Here, *k*=1,2
N	number of accepted solutions in each temperature (Epoch Length)
R	maximum number of consecutive temperature trails



Simulated Annealing

Edited by Cher Ming Tan

ISBN 978-953-7619-07-7

Hard cover, 420 pages

Publisher InTech

Published online 01, September, 2008

Published in print edition September, 2008

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821