A Multi-objective Simulated Annealing Algorithm for Solving the Flexible no-wait Flowshop Scheduling Problem with Transportation Times

Bahman Naderi[a,*], Hassan Sadeghi[b]

[a]*Assistant Professor, Young Researchers Club, Qazvin Branch, Islamic Azad University, Qazvin, Iran*
[b]*Bsc, Young Researchers Club, Qazvin Branch, Islamic Azad University, Qazvin, Iran*
Received 17 May, 2011; Revised 19 August, 2011; Accepted 13 November, 2011

Abstract

This paper deals with a bi-objective hybrid no-wait flowshop scheduling problem minimizing the make span and total weighted tardiness, in which we consider transportation times between stages. Obtaining an optimal solution for this type of complex, large-sized problem in a reasonable computational time by using traditional approaches and optimization tools is extremely difficult. This paper, therefore, presents a new multi-objective simulated annealing algorithm (MOSA). A set of experimental instances are carried out to evaluate the algorithm by advanced multi-objective performance measures. In fact, the performance of the algorithm is carefully evaluated in terms of the available algorithms by means of multi-objective performance measures and statistical tools. The results show that compared with other algorithms, a variant of our proposed MOSA produces a good performance.
Keywords: No-wait hybrid flowshop scheduling; Multi-objective simulated annealing algorithm; Makespan; Total weighted tardiness.

1. Introduction

A flowshop problem has a set of n jobs $\{J_1, J_2,\ldots,J_n\}$ and a set of m machines. All jobs visit machines in the same processing route, starting with machine 1 and finishing with machine m. The hybrid no-wait flowshop scheduling (HNFS) can be defined by a set of n jobs that need to be processed by a set of $m$stages. The jobs visit stages by the same order starting with stage 1, then stage 2 until stage m (Pinedo, 2008). It is known that the solution of HNFS is necessarily permutation flowshop-like too(these quences of jobs in all stages are the same).

Due to some characteristics of the circumstances and processing technology, the operations of a job must be performed without any waiting between stages, which is known as no-wait restriction. The operations of a job can be performed in any order, but they must be performed without any interruption in the stages and without any waiting between the stages. That is, each job must be processed continuously from its start to its completion (i.e. no-wait constraint). Hence, if necessary, the start of a job in one stage must be postponed so that the completion of this operation coincides with the beginning of the operation in another stage.

The following assumptions usually characterize the HNFS. Setup times are negligible. All the jobs are Independent and available for the process at time zero. All machines in each stage are continuously available. Each machine in each stage can process at most one job at a time. Each job can be processed by at most one machine at a time in each stage. The process of a job on a machine cannot be interrupted. Moreover, transportation times are considered in this problem. The transportation time from machine i-1 to machine i is denoted by $T_f$and the transportation time from machine$i$ to machine i-1 is denoted by T_b. The time to load and unload the transporter is included in the transportation time. When the transporter leaves the first machine, it always returns in time $T_f + T_b$to take the next job. This transportation time can be either job-dependent or job-independent. We assume that all transportations are job-independent, and transportations between two machines have to be done by one transporter.

The aim of production scheduling is to discover the job sequence to optimize one or some objectives. One of the most commonly used objectives is the minimization of makespan (C_{max}). Even for the best solution regarding the makespan, it is likely that a large number of jobs are completed after their due dates. In this case, the simultaneous consideration of the total tardiness (or TT)

*Corresponding author E-mail addresses: bahman.naderi@aut.ac.ir

minimization could result in a higher productivity. Therefore, we investigate the case of minimizing these two objectives.In fact, in this research we investigate the case of minimizing *TT* and C_{max}.

Almost all of the studies done are single-objective no-wait flowshop scheduling problems. The earliest research on the single-objective no-wait flowshop scheduling wasconducted by Wismer(1972). Later, different genetic algorithms were applied by Chen and Neppalli(1996) as well as Aldowaisan and Allahverdi(2004). Among the other metaheuristics, one could refer the reader to theparticle swarm optimization by Pan et al.(2008), the ant colony optimization by Shyu et al.(2004) and the tabu search by Grabowski and Pempera(2005).There are also some studies on multi-objective HNFS problems.For instance, Tavakkoli-Moghaddam et al. (2007)proposeda multi-objective immune algorithm for the NFS to minimize the weighted mean completion time and the weighted mean tardiness.In another study, Khalili and Tavakkoli-Moghaddam(2012)developed a new multi-objective electromagnetism algorithm for a bi-objective flowshop scheduling problem. To evaluate the performance of the proposed algorithm, we use a set of instances taken from the literature. Using multi-objective performance measures and statistical tools, the performance of the proposed algorithms are compared withthat of the available multi-objective immune algorithm (MOIA) proposed by Tavakoli-Moghaddam et.al(2007).Khalili(2012)also proposed an iterated local search algorithm for flexible flow lines with sequence dependent setup times to minimize the total weighted completion. In the same vein, Khalili(2012) studied multi-objective no-wait hybrid flowshop scheduling problems to minimize both makespan and total tardiness.

Similarly, in this paper we presentthe multi-objective no-wait flowshop scheduling problemwith minimizingboth makespan and total tardiness. Specifically, we proposea high performing novelmulti-objective solution method based onthe simulated annealingalgorithm.

The rest of the paper is organized as follows. Section 2describes the mathematical mode of the problem, and Section 3 presents the multi-objective simulated annealing (MOSA). Then Section 4 describes the experimental design to evaluate the proposed algorithms. Finally, Section 5 gives some interesting conclusions and suggestions for future studies.

2. The Mathematical Model for Hybrid No-Wait Flowshop Scheduling Problems

This paper presents a mixed integer linear program model for hybrid no-wait flowshop scheduling problems. In the mathematical model which is based on finding job in the sequence, the following parameters and indices are used.

n	Number of jobs
m	Number of machines
j, k	Indices for jobs where $j, k = \{1, 2, \dots, n\}$
i	Indices for machines where $i = \{1, 2, \dots, m\}$
m_i	Number of machines in stage i
l	Indices for machines at stage i where $\{1, 2, \dots, m_i\}$
$p_{j,i}$	Processing time of job j on machine i
r_{ji}	Transportation time of job j from machine i to machine i+1
d_j	Due date of job j
M	A large positive number

In the following part, the mathematical model is explained.

$X_{j,i,k}$	Binary variable taking value 1 if job j is processed after job k at stage i, and 0 otherwise. $k>j$
$Y_{j,i,l}$	Binary variable taking value 1 if job j is processed at stage i on machine l, and 0 otherwise.
$C_{j,i}$	Continuous variable for the completion time of job j at stage i
C_{max}	Continuous variable for makespan.
T_k	Continuous variable for the tardiness of job k

The model formulates the problem as follows:

Minimize $Z_1 = C_{max}$ (1)

$$Z_2 = \sum_{k=1}^{n} T_k \quad (2)$$

subject to:

$\sum_{l=1}^{m_i} Y_{j,i,l} = 1 \quad \forall_{j,i}$ (3)

$C_{j,1} \geq p_{j,1}$ (4)

$C_{j,i} = C_{j,i-1} + r_{ji-1} + p_{j,i} \forall_{j,i>1}$ (5)

$C_{j,i} \geq C_{k,i} + p_{j,i} - M \cdot (3 - X_{j,i,k} - Y_{j,i,l} - Y_{k,i,l})$

$\forall_{j<n,k>j,i,l}$ (6)

$C_{k,i} \geq C_{j,i} + p_{k,i} - M \cdot X_{j,i,k} - M \cdot (2 - Y_{j,i,l} - Y_{k,i,l})$

$\forall_{j<n,k>j,i,l}$ (7)

$C_{max} \geq C_{j,m} \forall_j$ (8)

$T_j \geq C_{j,m} - d_j \quad \forall_j$ (9)

$C_{j,i} \geq 0 \forall_{j,i}$ (10)

$X_{j,i,k} \in \{0,1\} \forall_{j,i,k>j}$ (11)

$Y_{j,i,l} \in \{0,1\} \forall_{j,i,l}$ (12)

In Equations (1) and (2), the makespan and the total tardiness are calculated, respectively. The first two constraint sets together ensure the construction of a feasible sequence. The subsequent constraint sets are to schedule the sequence obtained in the first two sets and compute the completion times, tardiness and makespan of the relevant schedule. More specifically, Constraint set (3) specifies which machine in each stage is assigned to each job. Constraint set (4) ensures that the completion time of every job on machine 1 is larger than its processing time on the machine. Constraint set (5) says

that once the process of a job on a machine is completed, the process of the job on the next machine must begin without any interruption. Constraint sets (6) and (7) are the dichotomous pairs of constraints relating to each possible job pair. They ensure that one machine processes at most one job at a time. Finally, Constraint sets (8) and (9) are used for obtaining the makespan and job tardiness, respectively, and Constraint sets (10), (11) and (12) define the decision variables.

3. The Multi-Objective Simulated Annealing Algorithm

Many real-world problems involve simultaneous optimization of several objectives.As mentioned before, finding an optimal solution for large-sized problems in a reasonable computational time by using traditional approaches and optimization tools (like solving mathematical models) is very difficult.Thus,a mathematical model for this problem in big size of job and machine is not suitable.

The objectives often compete and conflict with themselves. A multi-objective optimization problem containing the simultaneous minimization of p uncorrelated objectives can be defined as follows:

$$\min \mathrm{Z} = \left(f_1(x), f_2(x), \dots, f_p(x)\right) \quad \text{s.t. } x \in X \tag{13}$$

where$x$ is the decision vector (or a feasible solution), X is the set of feasible solution space, $f_l(x)$ is the$l$th objective function value of solution x.

In the multi-objective optimization problems, the simplest method is “a priori” one where objectives are first weighted and then combined into a single value. For instance, given two objectives f_1 and f_2 , a linear combination, such as $Z = wf_1 + (1-w)f_1$ where $0 \le w \le 1$, converts the master multi-objective optimization problem into a single-objective problem. However, giving a priori to $w$is a shortcoming that this method suffers from. Moreover, in the case that f_1 and f_2 are measured in different units, this method is likely to be misleading. Therefore, “a posteriori” method can be more appealing. In this method, the traditional concept of “optimum” solution does not apply. In fact, we may obtain a set of solutions which all are equally good due to the fact that it cannot be exactly clarified which one is better or worse. That is, all solutions in the set are the “best” solutions for the problem in a multi-objective scenario, rather than one optimum solution in a single-objective concept.

For example, consider a given bi-objective problem with two minimization objectives $f_1$andf_2. Let x_1 and x_2 be two solutions of this problem. If solution x_1 has a better f_1 value than solutionx_2 and yet a worse f_2 value, it is evident that neither solution is better than the other in a multi-objective sense. Now, consider the third solutionx_3 where $f_1(x_1) < f_1(x_3)$and $f_2(x_1) < f_2(x_3)$. In this case, it can be said that x_3 is worse than x_1. To properly compare solutions in the multi-objective optimization problems, some definitions are needed. The following definitionsare presented for the minimization cases.

Definition 1: strong (or strict) domination. The solution x_1 is said to strongly dominate the solution x_2 ($x_1 \prec\prec x_2$) if $f_j(x_1) < f_j(x_2) \forall_{j=1,2,\dots,p}$; solution x_1 is better than x_2 for all the objectives.

Definition 2: domination. The solution x_1 is said to dominate the solution x_2 ($x_1 \prec x_2$) if
1) $f_j(x_1) \not\succ f_j(x_2) \forall_{j=1,2,\dots,p}$; solution x_1 is not worse than x_2 for all the objectives.
2) $f_j(x_1) < f_j(x_2) \exists_{j=1,2,\dots,p}$; solution x_1 is better than x_2 for at least one objective.

Definition 3: weak domination. The solution x_1 is said to weakly dominate the solution x_2 ($x_1 \preccurlyeq x_2$) if $f_j(x_1) \not\succ f_j(x_2) \forall_{j=1,2,\dots,p}$; solution x_1 is not worse than x_2 for all the objectives.

Definition 4: incomparable domination. Solutions x_1 and x_2 are said to be incomparable ($x_1 || x_2$ or $x_2 || x_1$) if $f_j(x_1) \not\leq f_j(x_2)$ nor $f_j(x_1) \not\geq f_j(x_2) \forall_{j=1,2,\dots,p}$.

Note that all the above definitions are extendable to sets of solutions. For example, suppose A and $B$are two sets of solutions for a multi-objective optimization problem. The set A is set to (strongly) dominate B if for every solution$x_i \epsilon B$, there is at least the solution $x_j \epsilon A$ (strongly) dominatingx_i.

Definition 5: Pareto optimal set. Among the set of solutions A, the subset A' is said to be the Pareto optimal set if and only if it includes only and all solutions $x_i \epsilon A$ not dominated by any other solutions in $x_j \epsilon A$.

An approximation of the Pareto optimal set is said to be good if it is close to this set. Furthermore, a good spread of solutions is also desirable, i.e., an approximation set is good if the whole Pareto optimal set is adequately covered.

3.1. Introduction of the traditional simulated annealing

Simulated annealing (SA) belongs to the class of stochastic search algorithms known as meta-heuristics. It is a fast local search-based algorithm designed to provide good optimal or near-optimal solutions within a reasonable computation time(Kirkpatrick et al., 1983). Ever since its introduction, the SA has shown a high performance in large combinatorial optimization

problems, particularly in scheduling problems (Kubotani and Yoshimura, 2003;Minellaetal., 2008). The rationale for this algorithm comes from an analogy between the physical annealing of solid materials and optimization problems. It can be regarded as an enhanced version of local search or iterative improvement. Figure 1 shows the general outline of the traditional SA.

Procedure*of the traditional simulated annealing*
$t = T_0$
$x = initialization$
%Initial solution by another algorithm
$x_{best} = x$
while*stopping criterion is not met***do**
for *iter* = 1 **to***max***do**
$s =$ move $x$by an operator %generating a neighbor solution from x
if$f(s) < f(x)$**then** % Acceptance criterion
$x = s$
if$f(s) < f(x_{best})$**then** % check with the best solution
$x_{best} = s$
endif
else
if$random < exp\{-(f(s) - f(x))/t\}$**then**
$x = s$
endif
endif
endfor
$t = \alpha \cdot t$ % temperature decrease
Endwhile

Fig. 1.The general outline of the traditional simulated annealing

3.2. Multi-objective parametric simulated annealing

A typical SA starts with an initial solution and examines the possibility of improving it by repeatedly making small local changes in the incumbent solution until a stopping criterion is fulfilled. The SA performs this procedure in a way that occasionally permits changes that deteriorate the incumbent solution to increase the chance of leaving a local optimum. The probability of accepting worse solutions depends on the change in the goodness and the annealing temperature. The SA is commonly said to be the oldest among the meta-heuristics that has an explicit strategy to avoid local optima. Itstarts at a high temperature (T_0), so most of the moves are accepted at first steps of the procedure. The probability of doing such a move is decreased during the search.

3.2.1. Encoding scheme and initialization

The most frequently used encoding scheme for the flowshop is a simple permutation of jobs. The relative order of jobs in the permutation indicates the processing order of jobs on the first machine in the shop. To qualify our encoding scheme using the SA, the permutation of jobs is shown through random keys (RK). Each job is assigned a real number whose integer part is the machine number to which the job is assigned and whose fractional part is used to sort the jobs assigned to each machine.

The initial solution is produced by the NEH(Khalili and Tavakoli-Moghadam, 2012). This algorithm checks 100 neighbors at each temperaturet_i. The SA starts at a high temperature (t_0) and the temperature is slightly lowered under a certain mechanism which is called cooling schedule when the procedure proceeds. In this paper, we use the exponential cooling schedule,$t_i = \alpha \cdot t_{i-1}$,where $\alpha \in (0, 1)$ is the temperature decreasing rate.

The fundamental idea is to generate a new job sequence v a random rule from the neighborhood of incumbent sequence$x$. The new solution v is assessed by a mechanism called acceptance criterion to decide whether accept the$v$or reject it. Clearly, if the v improves the$x$, it is accepted. Moreover, the worse solution might be accepted by a probability function depending on the difference between the goodness of the two solutions and current temperaturet_i. Therefore, there is a higher chance of accepting the worse solution in higher temperatures.

3.2.2. Moving operator

Last but not least, t_i an operator is employed to generate a neighbor solution v from the current candidate solution x by making a slight change in it. This operator performs so as to avoid producing infeasible solutions. In this paper, we take into consideration two different move operators:

1) Swap: The positions of two randomly selected jobs are swapped. For example, consider a problem with $n = 5$ and some permutation {3, 5, 2, 4, 1}. Suppose the two randomly selected jobs are jobs 5 and 4. The corresponding positions are exchanged; therefore, we have {3, 4, 2, 5, 1}.
2) Single point: One randomly selected job is randomly relocated. Consider the previous permutation {3, 5, 2, 4, 1}. Suppose the selected job is job 4 and the new randomly chosen position becomes 2. Therefore, the new solution becomes {3, 4, 5, 2, 1}.

3.2.3. Acceptance criterion

As mentioned before, inthe single-objective SA, the variation is calculated$\Delta C = f(v) - f(x)$. If $\Delta C \leq 0$, solution$v$ is accepted. Otherwise, solution$v$ is accepted with a probability equal to$P_r = \exp(-\Delta C / t_i)$. In the multi-objective SA, there are several objectives and thus an effective criterion should consider all of them. Consequently, it cannot be straightforward. There are, among the others, four different frequently used criteria in the literature (Kubotani and Yoshimura, 2003):

1) Rule SL $$P_r = \min\left\{1, \exp\left(\frac{\sum_{j=1}^{p} w_j \cdot \Delta f_j}{t_i}\right)\right\} \quad (14)$$

2) Rule C $$P_r = \min\left\{1, \min_j\left(\exp\left\{\frac{w_j \cdot \Delta f_j}{t_i}\right\}\right)\right\} \quad (15)$$

3) Rule W $$P_r = \min\left\{1, \max_j\left(\exp\left\{\frac{w_j \cdot \Delta f_j}{t_i}\right\}\right)\right\} \quad (16)$$

where $\Delta f_j = f_j(v) - f_j(x)$, $\sum_{j=1}^{p} w_j = 1$ and $w_j$is the weight of the$j$th objective. $t_i$also donates the current temperature.

The fourth criterion is a more complicated criterion, named parameterized acceptance criterion (PAC). Let $e^{(l)} \epsilon R^p (l = 1, \dots, p)$ be a p-dimensional unit vector such that only the lth element is one and the others are zero, and let u be a unit vector definedas $u = \left(\frac{1}{\sqrt{p}}, \frac{1}{\sqrt{p}}, \dots, \frac{1}{\sqrt{p}}\right)$. The vector $a^{(l)}$ is defined by:

$$a^{(l)} = \frac{|\lambda| e^{(l)} + (1 - |\lambda|)u}{\| |\lambda| e^{(l)} + (1 - |\lambda|)u \|} \quad (17)$$

whereλ is a parameter such that $-1 \le \lambda \le 1$. Finally, the PAC is defined by:

4) $$P_r = \begin{cases} \min\left\{1, \min_l\left\{\exp\left[\frac{1}{t_i}\sum_{j=1}^{p} a_j^{(l)} \cdot w_j \cdot \Delta f_j\right]\right\}\right\}, & -1 \le \lambda < 0 \\ \min\left\{1, \min_l\left\{\exp\left[\frac{1}{t_i}\sum_{j=1}^{p} a_j^{(l)} \cdot w_j \cdot \Delta f_j\right]\right\}\right\}, & 0 \le \lambda \le 1 \end{cases} \quad (18)$$

where$a_j^{(l)}$ represent the jth element of the vector $a^{(l)}$. In our case, we assume that the weights are equal; therefore, they can be omitted from the above formulas. Figure 2 shows the behavior of the PAC regarding different values of λ in the bi-objective case (p = 2). In this figure, the gray regions represent areas where the candidate solutions are certainly accepted and the broken lines indicate the contour lines of the acceptance probability. In the PAC, when λ = −1, 0 and 1 are equivalent to Rules SL, C and W, respectively. An SA algorithm for each of these four acceptance criteria is developed, namely SLSA, CSA, WSA and PACSA.

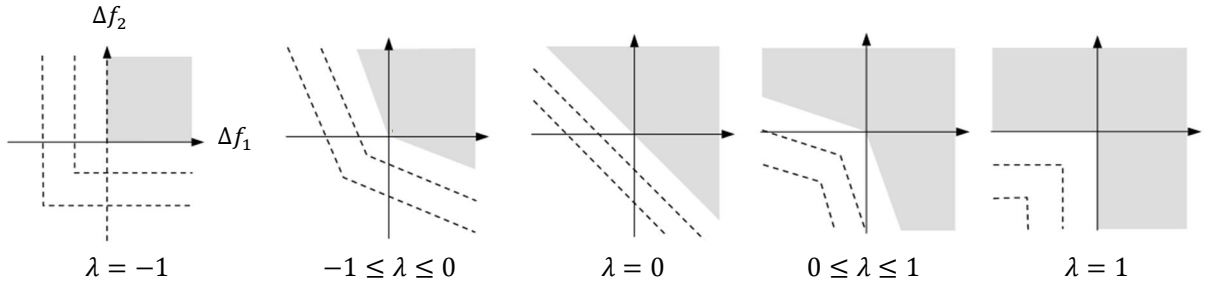


Fig. 2. Parameterized acceptance criterion

3.2.4. Updating the Pareto archive set

A Pareto archive set is usually designed to hold a limited number of non-dominated solutions. During the search, when a new non-dominated solution is obtained, it is put to the archive set if the archive set is not full. If a new solution enters the archive set, any solution in the archive dominated by this solution is removed from the set. Once the archive becomes full, a new non-dominated solution enters the archive if its distance to all the non-dominated solutions is greater than that to a pre-determined threshold. The distance between the new non-dominated solution and a given non-dominated solution in the archive is measured based on the Euclidean distance. The motivation is to keep diverse non-dominated solutions without losing any existing non-dominated solutions in the archive. The general procedure of our proposed SA is as follows.

Procedure*Simulated annealing*

Step 1: Initialize (initial solution, initial temperature, $r = 0$)
Step 2: Generate a new solution from the incumbent solution by an operator.
Step 3: Accept this new solution if it satisfies the acceptance criterion.
Step 4: Increase r by one unit. If $r > 100$, go to Step 2; otherwise, go to Step 5.
Step 5: Decrease the temperature, and put $r = 0$.
Step 6: If the stopping criterion is not met, go to Step 2; otherwise, Stop.

4. Experimental Evaluation

In this section, the performance of our proposed algorithms (i.e. SLSA, CSA, WSA and PASCA) is evaluated in terms of a benchmark. These algorithms are implemented in Borland C++ and run on a PC with 2.0 GHz Intel Core 2 Duo and 2 GB of RAM memory.The tested algorithms are SLSA, CSA, WSA, PACSA and MOIA proposed by (Tavakkoli-Moghaddam et al., 2007). These algorithms are stopped after a running time of 5×n×m milliseconds. Recently, the computational time is considered as a stopping criterion(Montgomery, 2000).

In multi-objective cases, performance quality measures (PQM)are more challenging. Many of the PQMs used in the literature have a serious drawback known as being non-Pareto-compliant. It means they can assign a better goodness to a given approximation Pareto set A and a worse one to another set B even when B dominates$A$. Unfortunately, it is even shown by (Knowles et al., 2006) that these PQMs not only are non-Pareto compliant, but also provide wrong and misleading results more often than not. As examples for these misleading PQMs, one may state, among other PQMs, the generational distance or maximum deviation from the best Pareto set (Geiger, 2007; Rahimi-Vahed and Mirghorbani, 2007). After all, three PQMs are known to give reliable analyses(Knowles et al., 2006)and overcome the mentioned shortcoming. These three PQMs sorted from easy to complicated are as follows:

1) Dominance ranking: To compare two approximation Pareto sets A and B, one can rank a given algorithm over another one by counting the number of points in A dominated by or equal to points in B.
2) Quality indicator: This is a function that assigns a real number to a full Pareto approximation set. The two major types of this PQM are hyper volume (I_H) and unary epsilon indicator (I_ε^1), respectively, proposed by Zitzler and Thiele (1999) and Zitzler et al. (2003).
3) Empirical attainment function: The relative frequency that each region is attained by the approximation set is calculated(Knowles et al., 2006).

Since both dominance ranking and empirical attainment function compare algorithms in pairs, they are impractical PQMs for cases in which several algorithms are compared.Moreover, in the case of empirical attainment function, the results of each pair comparison should be graphically analyzed. All in all, the best choice for our case is quality indicators. We employ both hyper volume (I_H) and unary epsilon (I_ε^1) indicators, because compared with a single indicator, the combination of quality indicators could provide more precise conclusions. If the results of the two quality indicators are in conflict with one another on preference ranking of two approximation sets, the two sets are incomparable. Each of the two quality indicators could be described as follows:

The hyper volume indicator (I_H): It calculates the hypervolume (or the area in bi-objective cases) covered by the approximation Pareto set given by one algorithm. In order to measure this quantity, the area must be bounded by a reference point (usually a point that is dominated by all the points). The higher values of I_H correspond to higher quality. Comparing the two sets A and$B$, A is preferable if $I_H(A) > I_H(B)$.

The unary epsilon indicator (I_ε^1): For the two approximation sets $A$and B, $I_\varepsilon^1(A,B)$ equals to

$$I_\varepsilon^1(A,B) = \max_{x\epsilon A} \min_{y\epsilon B} \max_{1\le l\le p} \frac{f_l(x)}{f_l(y)} \tag{19}$$

Although this binary indicator seemingly needs to compare all algorithms in pairs, Knowles et al. (2006) proposed a version in which the approximation set B is any reference set, usually the best-known Pareto set. Another advantage of this replacement is to measure how much worse an approximation set is with respect to the best-known Pareto set in the best case.

To obtainI_ε^1, we first normalize the objective function values using Eq. (19). In this case, the normalized I_ε^1 indicator ranges between one and two; and $I_\varepsilon^1 = 1$ for a given algorithm implies that its approximation set is not dominated by the best-known one. For a given instance, the reference set is the best Pareto set that is the combination of all the approximation Pareto sets obtained by any of the algorithms.

4.1. Parameter tuning

The proposed SA has two parameters {Initial temperature (T_0), cooling rate (α)} and move operators. In the case of PACSA, we have another parameter (λ). We also consider two levels of-0.5 and 0.5. Finally, the value of 0.5shows the better performance.

The considered levels of {Initial temperature (T_0), cooling rate (α)} and the move operators are as follows:

a) 3 levels ({100, 0.98}, {150, 0.97}, {250, 0.96})
b) Move operator (*MO*): 2 levels (SO, SPO).

Therefore, 3×2=6 different SAs are obtained by these levels and all the 75 instances are solved by them. The results are first transformed into the hyper volume indicator and the unary epsilon indicator and then analyzed by the means of the analysis of variance (ANOVA) and the least significant differences (LSD) test at the 95% confidence level. As shown in Figure 3,*popsize = 6* provides statistically better results than the other values of $popsize$ 2, 4 and 8. There is a statistically significant difference between the two move operators and in fact the SPO performs better.

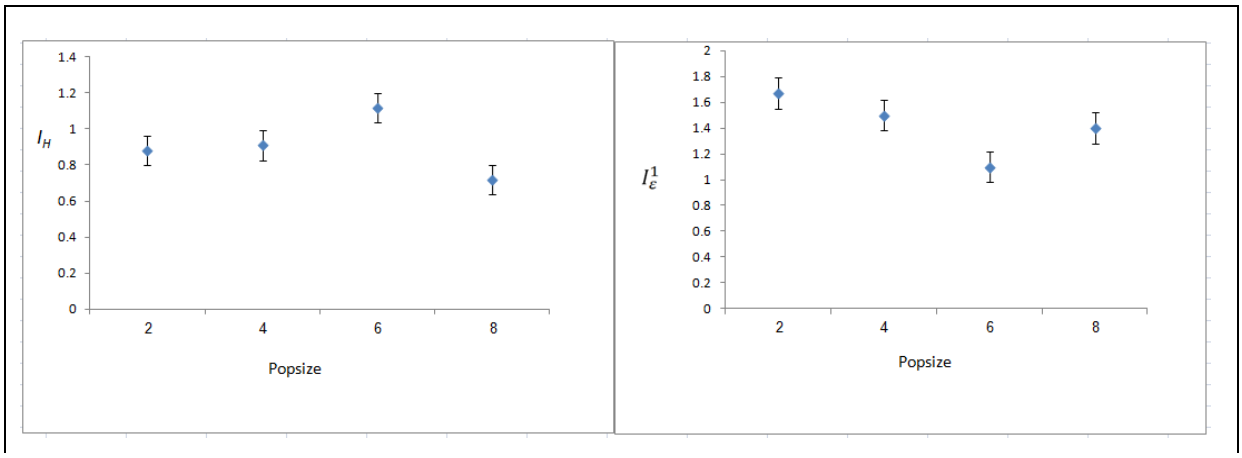


Fig. 3. Results of the parameter tuning

4.2. Experimental results

In this subsection, the performances of four tested algorithms (i.e., SLSA, CSA, WSA, PACSA) are evaluated on a set of instances. Data required for an instance consist of a number of jobs (n), a number of machines (m), a range of processing times (p_{ji}) and transportation times (T_f andT_b). We generate our instances based on the Taillard's benchmark values (Taillard, 1993). We have n = {20, 50, 100, 200} and m = {5, 10, 20} resulting in 15 combinations of$n \times m$. The processing time in the Taillard's instances are generated from a uniform distribution over the range (1, 99).The transportation times ($T_f$andT_b) come from a uniform distribution in the range (1, 30) that is 30% of processing time. Different levels of the factors result in 30 different scenarios. Then like the Taillard's benchmark, we produce 10 instances for each scenario. Therefore, we have 300 instances. To generate due dates of all n jobs, we use an approach similar to that of Khalili and Tavakoli-Moghaddam(2011) [6]. More specifically, for each job j, first we compute$S_j = \sum_{i=1}^{m}(p_{ji} + t_{ji})$. Then, the due date of each job is obtained as follows:$d_j = S_i(1 + 3\beta)$ where β is a random number between (0, 1).

Table 1 summarizes the average hyper volume (I_H) and unary epsilon indicator (I_ε^1) values for all the algorithms obtained in different problem sizes (n). Among the developed SAs, the PACSA obtains better results.

Table 1
$I_H$and$I_\varepsilon^1$for the algorithms grouped by$n$

n	Algorithms									
	CSA		PACSA		WSA		SLSA		MOIA	
	I_H	I_ε^1	I_H	I_ε^1	I_H	I_ε^1	I_H	I_ε^1	I_H	I_ε^1
20	1.112	1.181	1.129	1.111	1.018	1.212	0.978	1.212	0.982	1.29
50	1.118	1.188	1.119	1.123	1.111	1.192	0.922	1.311	0.925	1.392
100	1.1	1.191	1.189	1.134	1.121	1.199	0.712	1.412	0.721	1.412
200	1.111	1.192	1.141	1.139	1.108	1.29	0.645	1.455	0.799	1.623
Ave.	1.110	1.188	1.145	1.127	1.090	1.223	0.814	1.348	0.857	1.429

For the further analysis, we carry out the ANOVA method. The results show that there is a statistically significant difference between the performances of the algorithms. The means plot for the different algorithms with the least significant difference (LSD) intervals are shown in Figure 4. As it illustrates, our proposed PASCA provides statistically better results than other methods.
Figure 5 depicts the performances of the algorithms versus the number of jobs for both quality measures. As it shows, the proposed parametric SA algorithms keep robust performances in different problem sizes.

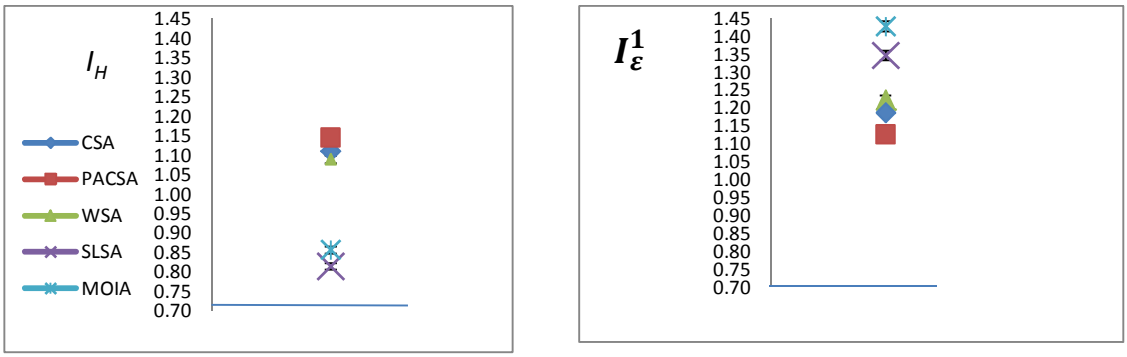
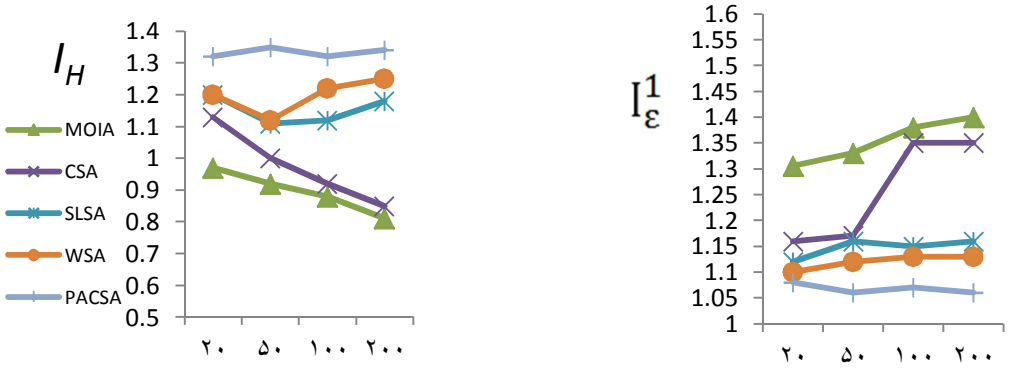


Fig. 4. Means plot and LSD intervals for algorithms in both I_H and I_ε^1



Number of jobs

Fig. 5. Means plot for the proposed SA versus the number of jobs in both I_H and I_ε^1

5. Conclusion and Future Work

This studyinvestigated the multi-objective hybrid no-wait flowshop scheduling problem. The aimwasto minimizeboth makespan and total tardiness. To accomplish this purpose, a novel multi-objective simulated algorithm was proposed. After tuning the algorithm, an experiment was designed to carefully evaluate its performance against some available algorithms in the literature. The results were analyzed through multi-objective performance measuresand statistical tools (ANOVA and LSD). The findings showed that the moderate variant of the proposed solution method outperforms the others.For furtherstudies, researchersmay work on extending the applications of multi-objective SA to other optimization problems. It is also interesting to study the performance of other novel solution methods for the same problem. Furthermore, the problem under consideration could be studied by various objectives such as total tardiness as well as early and tardy penalties.

6. References

[1] Aldowaisan, T.and Allahverdi, A. (2004).New heuristics for m-machine no-wait flowshop to minimize total completion time. Omega, 32(5), 345–352.

[2] Altiparmak, F.,Gen, M.,Lin, L. andPaksoy, T. (2006).A genetic algorithm approach for multi-objective optimization of supply chain networks. Computers & Industrial Engineering, 51, 196–215.

[3] Chen, R. andNeppalli, C. (1996).Genetic algorithms applied to the continuous flow shop problem. Computers and Industrial Engineering, 30(4), 919–929.

[4] Geiger M. (2007).On operators and search space topology in multi-objective flowshop scheduling. European Journal of Operational Research, 181(1), 195-206.

[5] Grabowski, J. and Pempera, J. (2005)Some local search algorithms for no-wait flow-shop problem withmakespan criterion. Computers and Operations Research, 32, 2197–2212.

[6] Khalili, M. and Tavakoli-Moghadam, R. (2012). A multi-objective electromagnetism algorithm for a bi-objective flowshop scheduling problem. Journal of Manufacturing Systems, Article in press, doi:10.1016/j.jmsy.2011.08.002.

[7] Khalili, M. (2012).An iterated local search algorithm for flexible flow lines with sequence dependent setup times to minimize total weighted completion. International Journal of Management Science and Engineering Management, 7(1), 63-66.

[8] Khalili, M. (2012). Multi-objective no-wait hybrid flowshop scheduling problem with transportation times. International Journal of Computational Science and Engineering, Article in press.
[9] Kirkpatrick, S., Gelatt, J. and Vecchi M. (1983).Optimization by simulated annealing. Science 220, 671-680.
[10] Knowles, J., Thiele, L. and Zitzler, E. (2006).A tutorial on the performance assessment of stochastic multi-objective optimizers. Technical Report 214. Computer Engineering and Networks Laboratory (TIK). ETH Zurich.
[11] Kubotani, H. andYoshimura, K. (2003).Performance evaluation of acceptance probability functions for multi-objective SA. Computers & Operations Research, 30, 427-442.
[12] Minella, G., Ruiz, R. and Ciavotta, M. (2008).A review and evaluation of multi-objective algorithms for the flowshop scheduling problem. INFORMS Journal on Computing, 20, 451-471.
[13] Montgomery, D. (2000).Design and analysis of experiments. Fifth edition, John Wiley & Sons.
[14] Pan, Q., Tasgetiren, M. and Liang, Y. (2008). A discrete particle swarm optimization algorithm for the no-wait flowshop scheduling problem. Computers and Operations Research, 35, 2807–2839.
[15] Pinedo, M. (2008).Scheduling: Theory, Algorithms, and Systems. 3th edition, Springer Science+Business Media, New York.
[16] Rahimi-Vahed, A. and Mirghorbani, S. (2007).A multi-objective particle swarm for a flowshop scheduling problem. Journal of Combinatorial Optimization, 13, 79-102.
[17] Ruiz, R. and Stützle, T. (2008).An iterated greedy heuristic for the sequence dependent setup times flowshop problem with makespan and weighted tardiness objectives. European Journal of Operational Research, 187(3), 1143-1159.
[18] Shyu, S., Lin, B. and Yin P. (2004).Application of ant colony optimization for no-wait flowshop scheduling problem to minimize the total completion time. Computers and Industrial Engineering, 47, 181–193.
[19] Taillard, E. (1993). Benchmarks for basic scheduling problems. European Journal of Operational Research, 64(2), 278-285.
[20] Tavakkoli-Moghaddam, R., Vahed, A. andHosseinMirzaei, A. (2007).A hybrid multi-objective immune algorithm for a flow shop scheduling problem with bi-objectives: Weighted mean completion time and weighted mean tardiness. Information Sciences, 177, 5072–5090.
[21] Wismer, D. (1972). Solution of flowshop-scheduling problem with no intermediate queues. Operations Research, 20, 689–697.
[22] Zitzler, E. and Thiele, L. (1999). Multi-objective evolutionary algorithms: A comparative case study and the strength Pareto approach. IEEE Transactions evolutionary Computation, 3, 257-71.
[23] Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. and Fonseca, V. (2003). Performance assessment of multi-objective optimizers: An analysis and review. IEEE Transactions evolutionary Computation, 7, 117-132.