International Business Research; Vol. 10, No. 12; 2017
ISSN 1913-9004 E-ISSN 1913-9012
Published by Canadian Center of Science and Education

# A Bi-objective Simulation-optimization Approach for Solving a No-wait two Stages Flexible Flow Shop Scheduling Problem with Rework Ability

Mohammad Rahmanidoust[1], Jianguo Zheng[1]

[1]Glorious Sun School of Business and Management, DongHua University, Shanghai, China

Correspondence: Mohammad Rahmanidoust, Glorious Sun School of Business and Management, Donghua University, Shanghai, China.

Received: September 26, 2017 Accepted: November 13, 2017 Online Published: November 16, 2017

doi:10.5539/ibr.v10n12p197 URL: https://doi.org/10.5539/ibr.v10n12p197

**Abstract**

The paper suggests a new rule; called no-wait process. The rule has two stages, and is a flexible flow shop scheduling. The process is the subject to maximize tardiness while minimizing the makespan. This hybrid flow shop problem is known to be NP-hard. Therefore, we come to first, Non-dominated Sorting Genetic Algorithm (NSGA-II), then, Multi-Objective Imperialist Competitive Algorithm (MOICA) and finally, Pareto Archive Evolutionary Strategy (PAES) as three multi-objective Pareto based metaheuristic optimization methods. They are developed to solve the problem to approximately figure out optimal Pareto front. The method is investigated in several problems that differed in size and terms of relative percentage deviation of performance metrics. The conclusion, developed by this method is the most efficient and practicable algorithm at the end.

**Keywords*:*** multi-objective optimization, no wait, flexible flow shop, NSGA-II, MOICA, PAES

## 1. Introduction

Scheduling is arranging and planning sequence of event to complete the work. The goal is to utilize the resources optimally while reaching targets. This field has been attracted by many scholars during recent years (Mosheiov and Sidney 2010, Vallada and Ruiz 2011, Shafaei *et al.* 2011, Rabiee *et al.* 2012, Jolai *et al.* 2013, Ullrich 2013, Jolai *et al.* 2014, Yang *et al.* 2014, Tayebi Araghi *et al.* 2014, Xu *et al.* 2015, Rabiee *et al.* 2016, Nesello *et al.* 2017). The flow shop problem is an important subject in scheduling. One of the reasons is that most of the manufacturing systems follow batch shops, flow shop or semi flow shop routings (Baker 1974, Johnson 1954, Lin *et al.* 2006). Flexible flow shop or hybrid flow shop, is the one of the most important classes of scheduling which also defined as a flow shop with parallel machines and flexible flow line. For a literature review in this area, the readers are referred to those of (Richard and Zhang 1999, Ruiz and Vazquez-Rodriguez 2010, Ribas *et al.* 2010)

In a no-wait flow shop, the jobs are processed from one machine to the next one without waiting time (Huang *et al.* 2009). Suppose there are some sequences of jobs, some specified procedures in each that are processed by machine in disciplinary order. No pause or interruption is supposed to occur in the line. In other word, when the process starts, there is no stop for object before or after each machine. No repetition is allowed, one job at a given time by a machine. Therefore, when needed, the start of a job on the first machine must be delayed in order to meet the no-wait requirement (Tasgetiren *et al.* 2007). Nagano et al. (2013) examines the *m* machine no-wait flow shop problem with setup times of a job separated from its processing time. The performance measure considered is the makespan. The hybrid metaheuristic Evolutionary Cluster Search (ECS_NSL) is employed to solve this scheduling problem. Nagano et al. (2015) addressed the problem of scheduling jobs in a no-wait flow shop with sequence-dependent setup times with the objective of minimizing the total flow time. As this problem is well-known for being NP-hard, they presented a new constructive heuristic, named QUARTS, in order to obtain good approximate solutions in a short CPU time. Samarghandi and ElMekkawy (2012) studied the problem of no-wait flow shop and proposed two frameworks based on genetic algorithm and particle swarm optimization to deal with the problem. Samarghandi and ElMekkawy (2014) proposed PSO algorithm to solve the problem of scheduling a no-wait flow-shop system with sequence-dependent set-up times. The application of this problem can be found in industries such as chemical industry, steel production, just-in-time manufacturing process, service industries, etc (Rajendran 1994, Grabowski and Pempera 2000, Raaymakers and Hoogeveen 2000, Aldowaisan and Allahverdi 2004). A detailed survey of the research and applications on this topic has been given by Nagano et al. (2016).

Most of literatures about no wait hybrid flow shop scheduling problem have mainly focused on a single objective. For example, Liu *et al.* (2003) suggests an algorithm known as Least Deviation (LD) in which the focus is on only one machine in each station. The key performance indicator in this point of view is the makespan. The performance is thus high using this algorithm. Also the algorithm is easier in computation and implementation. Having created such values, this is considered favourable.

Xie *et al.* (2004) suggests a new heuristic algorithm named Minimum Deviation Algorithm (MDA) to minimize makespan in a similar method. MDA also performs better than partition method, partition method with LPT, Johnson's and modified Johnson's algorithms. Huang *et al. (2009)* considered a no-wait two stage flexible flow shop with setup times and with minimum total completion time key performance indicator. The author represents an enhanced programming as well as an Ant Colony approach. The solution was satisfactory with the approach and the results were efficient.

To best use of resources, Jolai *et al.* (2009) introduced no-wait flexible flow line scheduling problem with time windows and job rejection which is, in turn, an extension of production and delivery scheduling. He presented a similar method known as integer-linear programming model and genetic algorithm process as well.

In comparison with LINGO, studies show that the GA is a better solution in a computational time. Jolai et al. (2012) introduced a new hybrid algorithm with sequence-dependent setup times to minimize the total completion time. They suggest three algorithms. One, Population Based Simulated Annealing (PBSA), second, Adapted Imperialist Competitive Algorithm (AICA) and finally, hybridization of Adapted Imperialist Competitive and Population Based Simulated annealing (AICA+PBSA) for the problem. All the studies support the hybrid algorithm against the others which are applied in literature for related production scheduling problem. Rabiee et al. (2014) proposed the problem with respect to unrelated parallel machines, sequence-dependent setup times, probable reworks and different ready times to actualize the problem. What they proposed is based on imperialist competitive algorithm (ICA), simulated annealing (SA), variable neighborhood search (VNS) and genetic algorithm (GA) to solve the problem. The result revealed the advantages of our algorithm. To reduce makespan, Ramezani *et al.* (2013) suggested no-wait scheduling problem focused on set up time which is anticipatory and also sequence dependent in a flexible flow shop environment with two sets of same machines in parallel. They introduced it as a novel method since it was NP-hard. Their meta-heuristic method was about invasive weed optimization, adjustable neighborhood exploration and simulated strengthening to attack of the problem. The result showed, hybrid-metaheuristic outperformed in comparison with singular ones.

As said before, most of studies concentrated on single objective problem. But in reality, there is no single objective at all and we need to consider a batch of goals at once. However, there are also some studies working on multi objective in no wait flow shop scheduling problem. Allahverdi and Aldowisan (2004) suggested a method in no wait flow shop scheduling problem in with sum of makespan is important and also the maximum delay as a measure. Their methods for comparison were hybrid simulated annealing and a hybrid genetic heuristics. Also they suggested a dominance relation (DR) and a branch-and-bound algorithm. Herein also, after computation, the heuristic method performs higher and better in comparison with existing heuristics if the makespan and maximum lateness is considered significant. Also the dominance relation and branch and bound algorithm were totally effective. Aiming to minimize average of tardiness and time, Rahimi-Vahed *et al.* (2008) offered a bi-criteria no-wait flow shop scheduling problem. What they suggest is a new method named multi-objective scatter search as a metaheuristic algorithm for finding near optimal Pareto frontier. They were looking for effectiveness of this approach by solving some experimental problems in comparison with SPEA-II. Here also the better performance was appeared in multi-objective scatter search.

Multi-objective immune algorithm also proposed by Tavakkoli-Moghaddam*et al.* (2007) to minimize two goals as weighted average completion time and weighted average tardiness for a no-wait flow shop scheduling problem. He compared the algorithm with a conventional multi-objective genetic algorithm, i.e., SPEA-II. The generic algorithm won the game especially for significant and big problems. Pan *et al.* (2009) offered another algorithm called discrete differential evolution (DDE) for solving the no-wait flow shop scheduling problems with makespan and maximum tardiness measures. The results based on the famous benchmarks and statistical performance comparisons showed that DDE algorithm is much more efficient in comparison to the hybrid differential evolution (HDE) algorithm proposed by Qian *et al.* (2009). For the same objective, i.e. to minimize makespan and tardiness, Khalili (2012) proposed a multi-objective no-wait hybrid flow shop scheduling problem and suggested a novel Multi-Objective Electromagnetism Algorithm (MOEA) to solve the problem. He formulated the problem with mixed integer-linear programming models and proposed an effective Multi-Objective Electromagnetism Algorithm (MOEA) to reach the goal.

As far as multi-objective approach is considered, there are a few studies about no wait flexible flow shop problem. The author presented three multi-objective based algorithms to discover a no wait two stage flexible flow shop scheduling problem with a number of machines in each stage. The goals were minimizing makespan (i.e. $C_{max}$) and maximum tardiness (i.e. $T_{max}$).

In the next section; section 2, the multi objective terms are discussed. Then the author wrote about the bi-criteria no wait two stages flexible flow shop. Then in section 3, the multi-objective optimization search techniques are considered. Investigating the proficiency of suggested multi-objective metaheuristic approaches is presented next section. And at last, the outcome of the research is induced and directions for further researches are depicted in section 5.

## 2. Multi-objective No-wait Two-stage Flexible Flow Shop Optimization

Now we turn to the problem statement. That is the concept of the multi objective optimization and also the structure of the problem.

### *2.1 Multi-objective Optimization*

A multi-objective optimization problem formula is as below:

$$\min_{x \in X^{n_x}} f(x) = \{f_1(x), f_2(x), ..., f_M(x)\} \quad (1)$$

$$s.t. \quad g(x) \le 0, \quad h(x) = 0 \quad (2)$$

Wherein $g(x) \le 0$, $h(x) = 0$ shows the possible solution in $n_x$ dimensional search space and $f(x)$ is a $M$ dimensional vector of objective values. Map between decision variables of $x \in X^{n_x}$ and objective space of $f \in F^M$ is determined by objective functions. In reality, the aim of a multi-objective optimization is to figure out the entire non- dominated solutions of the problem (any solution that is not able to develop an objective function with no effect on other objective).

If any of below conditions are met in an optimization problem with minimum objectives, solutions $x_1$ dominates solution $x_2$.

(1) For every single objective $f(x_1) \le f(x_2)$.

(2) At least in one objective $f(x_1)$ has a lower value compared with $f(x_2)$.

Having defined the dominant solutions, the optimal solution of a multi-objective optimization problem is defined as set of non-dominated solutions known as Pareto-optimal set which forms the Pareto front (CoelloCoello *et al*. 2002).

### *2.2 The Statement of Bi Criteria No-wait two Stage Flexible Flow Shop Problem*

The no-wait two stage flexible flow shop scheduling problem is shown below: Given the processing time $P_{ij}$ of job $j$ on stage $i$($i$=1, 2), each of $n$ job will be sequentially processed in stage 1, 2 respectively. At each stage there are $m_i$ machines. Also at a given time, each machine can process maximum a single job. Likewise, each action needs to be processed on one machine. Once the order of the action at the first stage is cleared the similar order is done for the second stage. To meet the no-wait boundaries, the end time of a job on a machine must be similar to the start time of the job on the next machine. This way, there is no elapse time in the entire operation. The aim here is, reduction of makespan ($C_{max}$). The matching fitness function is considered at below:

$$C_j = \text{Completion time of job j} \quad (3)$$

$$\text{Makespan} = C_{\max} = \max(C_j) \quad (4)$$

$$\min z_1 = C_{\max} \quad (5)$$

Next aim is reduction of maximum tardiness which is calculated below:

$$\min z_2 = T_{\max} \quad (6)$$

Wherein $T_i$ is the tardiness of job i, equal to $\max(0, C_i - d_i)$, and $d_i$ is the due date of job $i$.

## 3. Possible Approaches to Find Solution

There are many classical approaches to solve multi-objective problems among them includes embracing goal programming, integer programming, e-constraint method and weighted sum method. The main features of the classical methods can be described as follows (Deb 2001, CoelloCoello *et al.* 2002):

1. Changing the problem from multi-objective to single objective
2. Experimentally, the methods might be applied randomly to find out the best solution
3. Each typical method includes some user-defined parameters that are not easy to set in an arbitrary problem. Some meta-heuristics have been developed to eliminate such deficiencies. They are genetic algorithms and evolutionary computation. The ability to figure out a reasonable estimate of Pareto frontier in one operation and good computational time, is one of the benefit of this method.

This paper offers three multi-objective metaheuristic methods to solve the problem; NSGA-II (Non-dominated Sorting Genetic Algorithm), MOICA (Multi-Objective Imperialist Competitive Algorithm) and PAES (Pareto Archive Evolutionary Strategy). They are to examine the output of the algorithms in solving the no-wait two stages flexible flow shop problem.

Metaheuristic algorithms are generally, based on a searching system which is random. Here, the problem altered from a phenotype into a genotype that is informally called chromosome. To discover the best solution, it uses intensification as well as diversification where the first intends to use local search area and the second explores the optimal solution globally. The chromosome, fitness evaluation, related operators and structures of applied system are described elaborated respectively:

### *3.1 Solution Representation and Fitness Evaluation*

Some random values that are equals to the number of jobs in length is generated from 0 to 1. This is to show the chromosome. The jobs then are tossed by finding the increasing order of values in vector. See figure 1.

| | | | | | |
|---|---|---|---|---|---|
| Chromosome | 0.45 | 0.63 | 0.13 | 0.33 | 0.77 |
| Job sequence | 3 | 4 | 1 | 2 | 5 |

Figure 1. An example of chromosome representation and its job sequence

After calculation of jobs, procedure of machine assignments is done using a heuristic method. It means in each stage, to assign a job to the machines, the earliest available time machine is chosen and the job with the highest importance is assigned to that machine. If there are two important jobs at the same time, one is chosen randomly. The pseudo code of heuristic procedure is shown in Figure 2.

### *3.1 Genetic Operators*

#### 3.1.1 Crossover

The process in which tow chromosomes are prevented from coupling making progeny is called crossover. The process aims to find a better solution by mixing the chromosomes. The study uses a uniform crossover that first creates a random binary mask with the similar extend as the chromosomes and then substitutes relative gene material of parent chromosome based on created binary mask. This crossover results in a good exploitation of solution space (Syswerda 1989).

#### 3.1.2 Mutation

After crossover, we perform an exchange mutation. For exchange mutation, two different arbitrary genes of the parent chromosome choose and swap the allele values. (Eiben and Smith. 2003).

Set $Q = \emptyset, t_{1i} = 0 \ (i = 1,2,\ldots,m_1),\ t_{2j} = 0 \ (j = 1,2,\ldots,m_2) \min(t_{2j}) = T_2,$
$\min(t_{1i}) = T_1$ and $T = T_2 - T_1$
Set $\pi = \{\pi_1, \pi_2, \ldots, \pi_n\}$ as job sequence and add $q$ to set of scheduled jobs
While $\pi = \emptyset$
    For $k = \pi_1 : \pi_n$
        If $k = \pi_1$
$t_{1i} = t_{1i} + sdst(\pi_{k-1}, \pi_k, 1);$
$t_{2j} = t_{2j} + sdst(\pi_{k-1}, \pi_k, 2);$
        Else
        If $p_{1,\pi_i} - T \geq 0$
$t_{1j} = t_{1j} + p_{1,\pi_k};$
            While $rand \leq P_{re1,\pi_k}$
$t_{1j} = t_{1j} + rt_{1j};$
            End-While
$t_{1j} = t_{1j} + sdst(\pi_{k-1}, \pi_k, 1);$
$t_{2j} = t_{2j} + p_{2,j};$
            While $rand \leq P_{re2,\pi_k}$
$t_{2j} = t_{2j} + rt_{2,\pi_i};$
            End-While
$t_{2j} = t_{2j} + sdst(\pi_{k-1}, \pi_k, 2);$
        Else
$t_{1j} = t_{2j};$
            While $rand \leq P_{re1,\pi_k}$
$t_{1j} = t_{1j} + rt_{1,\pi_k};$
            End-While
$t_{1j} = t_{1j} + sdst(\pi_{k-1}, \pi_k, 1);$
$t_{2j} = t_{2j} + p_{2,\pi_k};$
            While $rand \leq P_{re2,\pi_k}$
$t_{2j} = t_{2j} + rt_{2,\pi_k};$
            End-While
$t_{2j} = t_{2j} + sdst(\pi_{k-1}, \pi_k, 2);$
        End-If
    End-For
$\pi \longleftarrow \pi - \{j\}$ and add $j$ to $q$;
    While-End

Figure 2. Pseudo code of simulator

### *3.3. Multi-objective Algorithms*

#### 3.3.1 Non-dominated Sorting Genetic Algorithm (NSGA-II)

Probability of gaining Pareto-optimal solution using GAs are high, since they work with body of points. This makes it a strong tool for MOOPs too. The Non-dominated Sorting Genetic Algorithm (NSGA-II) is a well-known and extensively used algorithm based on its predecessor NSGA and proposed by Deb et al (2002). Essentially, NSGA-II differs from non-dominated sorting Genetic Algorithm (NSGA) implementation in a number of ways. First, NSGA-II uses an elite-preserving mechanism, thereby assuring the preservation of previously found good solutions. Second, NSGA-II uses a fast non-dominated sorting procedure. Third, NSGA-II does not require any tunable parameter thereby making the algorithm independent of the user (Sivakumar et al, 2011). NSGA-II is a fast and very efficient Multi-objective evolutionary algorithm (MOEA), which incorporates the features of an elitist archive and a rule for adaptation assignment that takes into account both the rank and the distance of each solution regarding others. Salazar and Kishor have applied and compared the efficiency of NSGA-II with existing methods for reliability optimization problems (Kishor et al, 2008).

NSGA-II is an elitist multi-objective evolutionary algorithm which carries out an approximation of the Pareto front, based on the non-dominance concept. For achieving different Pareto fronts, a ranking procedure is performed at each generation. Also, this algorithm takes advantage of an operator called crowding operator for its diversification. NSGA-II starts from a randomly generated population of chromosomes (solutions), P_0 of size N. The population is sorted based on non-domination. Each solution is assigned a fitness (or rank) equal to its non-domination level (1 is the best level) the minimization of the fitness being assumed. A children population (Q_0) of size N is then created by applying the genetic operators: binary tournament selection, recombination, and mutation (Furtuna, et al, 2011). NSGA-II has been used in several prior studies like: Minella et al. (2008), Behnamian et al (2009), Zandieh and Karimi (20111) and Rabiee et al (2012), Asefi et al. (2014). The crossover and mutation operators in this algorithm are as mention in NSGAII. The framework of the proposed NSGAII is generally illustrated in the following pseudo code (Figure 3).

**Begin**
**Input** : $N,\ P_c,\ P_m,\ Max\_Gen;$
Generate $Init\_Pop(P);$
Repair $Init\_Pop(P);$
Evaluate fitness values of the $Init\_Pop;$
Assign rank base on pareto dominance sort;
**for** $i = 1\ to\ Max\_Gen$ **do**
  **for** $j = 1\ to$ round$[(P_c \times N)/2]$
    Select two individuals: $(X_1,\ X_2)$**;**
    Select one scenario for crossover operation;
    perform one point crossover : $(X_1,\ X_2) \to (X_1^{'},\ X_2^{'})$ ;
  **endfor**
  **for** $j = 1\ to$ round$[(P_m \times N)/2]$
    Select an individual: $X$**;**
    Select one scenario for mutation/ neighborhood operation;
    Select one case between (swap, reversion, insertion);
    perform mutation : $X \to X^{''}$ ;
  **endfor**
  Repair $New\_Pop(P);$
  Combime offsprings and parents $\{P \cup Q\}$
  Assign rank based on pareto dominance sorting algorithm;
  Calculate the crowded distance of individuals in each front;
  Select the best N individual base on rank and crowded distance;
**end for**
**Output** : Extract the best pareto front*;*
**end**

Figure 3. Proposed NSGA-II algorithm in pseudo code

### 3.3.2 PAES Algorithm

The study suggests a simple multi-purpose metaheuristic algorithm called PAES. Offered by Knowles and Corne (1999, 2000), the algorithm applies a local search development strategy to a non- dominated solution in a pool of solutions that already applied. Hereunder, the most alternates of PAES which is (1+1) development strategy is discussed.

A solution is randomly generated, evaluated and saved in archived. Here is when the algorithm starts. Procedure at iteration t is continued with creating a new solution by transform current solution and compare it to current solution for dominance. The one with more dominance is accepted. If both solutions have same dominance priority, the new one is compared to archived solution which is archived. The accepted solution in the archive added there and removes the rejected one. Otherwise the new solution replaced by one of the archive member. At last, if there is no dominate member in the archive and archive has still vacancy, the new solution is added to archive. If there is no space in solution pool, the solution in the busiest area is removed and the new one is added. In when considering intricacy of problem to find non-dominated solutions, we consider a limitless archive scope to maintain non-dominated solutions. This is to obtain more Pareto solutions. The structure of proposed PAES is shown in Figure 4.

```
Begin
Input : MaxIteration;
Generate Init_Sol and set it as Current_Sol;
Evaluate fitness value of the Current_Sol;
Add Current_Sol to archive ;
for i = 1 to MaxIteration do
    Generate New_Sol by mutation of Current_Sol;
    Evaluate fitness value of the New_Sol;
    if New_Sol dominates Current_Sol
        Set New_Sol as Current_Sol;
        Update_Archive;
    elseif Current_Sol dominates New_Sol
            Discard New_Sol;
    else Current_Sol and New_Sol don't dominate each other;
            Update_Archive using New_Sol;
            Randomly select next Current_Sol betwen New_sol and Current_Sol ;
    end if
end for
Output : Extract nondominated solution as pareto front;
end
```

Figure 4. PAES algorithm

3.3.3 Multi-objective Imperialist Competitive Algorithm (MOICA)

3.3.3.1 Creating Primary Empires

Every single solution in the imperialist competitive algorithm simulates as an array. The arrays include different values that need to be adjusted. What is called chromosome in GA terminology, is named country here. In an N-dimensional optimization problem, a country is a 1×N array. This array is defined by: $country = [p_1, p_2, p_3, ..., p_N]$, where pi is the variable to be optimized. Each variable in a country denotes a socio-political characteristic of a country. From this point of view, the algorithm searches for the best country that is the country with the best combination of socio-political characteristics, such as culture, language and economic policy (Atashpaz-Gargari and Lucas 2007). After country development, a non-dominance technique and a crowding distance are used to shape the fronts and rank member of each front, respectively. At that point, the members of front one are archived. Non-dominance technique and crowding distance described as below:

1: Non-dominance technique: imagine that there are r objective functions. When the following conditions are satisfied, the solution x1 dominates another solution *x2*. If *x1* and *x2* do not dominate each other, they are placed in the same front.

(1) For all the objective functions, solution *x1* is not poorer than another solution *x2*.

(2) For at least one of the r objective functions *x1* is exactly better than *x2*.

Solutions that are not dominated by others, constitute in front number 1. Meanwhile, the solutions that are only dominated by solutions in front number 1, organize front number 2. The same order applies to crate the other fronts which is shown in Figure 5.

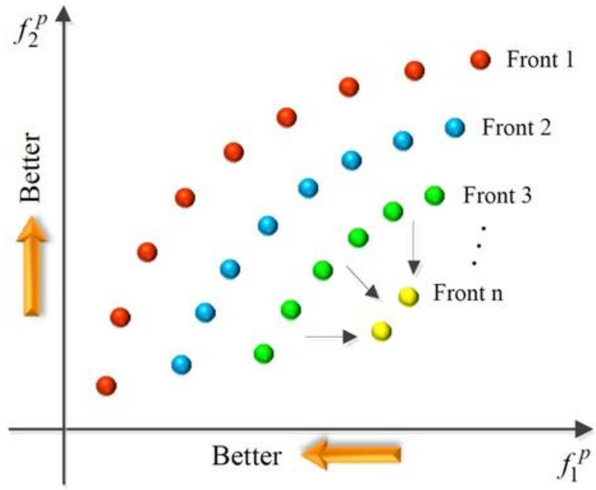



Figure 5. Non-dominance technique

2. Crowding distance: this is a tool to show the quantity of solutions in each step. See below figure. This is an estimate of the solution mass around a given solution.

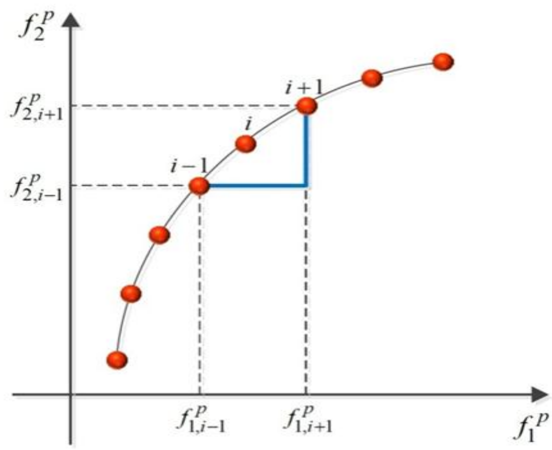



Figure 6. Crowding distance

The crowding distance measure which is used in MOICA is shown in equation (7). The solutions having a lower value of the crowding distance are prioritized over solutions with upper value of the crowding distance.

$$CD_i = \sum_{k=1}^{r} \frac{f_{k,i+1}^{p} - f_{k,i-1}^{p}}{f_{k,total}^{p,\max} - f_{k,total}^{p,\min}} \tag{7}$$

Where:

- the number of objective functions, $f_{k,i+1}^{p}$ is the k-th objective function of the (i+1)-th solution
- $f_{k,i-1}^{p}$ is the k-th objective function of the (i-1)-th solution after sorting the population according to crowding distance of the k-th objective function
- $f_{k,total}^{p,\max}$ and $f_{k,total}^{p,\min}$ are the maximum and minimum value of objective function k, respectively.

Next, the prioritized solution is selected in the mass in terms of non-dominance and crowding distance. The selected solution is imperialists and the rests are colonies.

In order to compute the cost of prioritized solution (the imperialist), each target function is calculated. After that each target function is calculated:

$$cost_{i,n} = \frac{\left| f_{i,n}^{p} - f_{i}^{p,best} \right|}{f_{i,total}^{p,\max} - f_{i,total}^{p,\min}} \tag{8}$$

Where:

- $cost_{i,n}$ is the normalized value of objective function i for imperialist *n*
- $f_{i,n}^{p}$ is the value of the objective function i for imperialist n
- $f_{i}^{p,best}$, $f_{i,total}^{p,\max}$ and $f_{i,total}^{p,\min}$ are the best, maximum and minimum values of objective function i in each iteration, respectively.

At last, the total value of each imperialist is calculated through:

$$Total\ Cost_{n} = \sum_{i=1}^{r} \cos t_{i,n} \tag{9}$$

Where:

- *r* is the quantity of target function

After calculation the cost, the strength of each imperialist is obtained as well as the colonies distributed among the imperialist according to power of each imperialist country.

$$p_n = \left| \frac{Total\ Cost_n}{\sum_{i=1}^{N_{imp}} Total\ Cost_n} \right| \tag{10}$$

At this point, the primary quantity of colonies is calculated as below:

$$NC_n = round\left\{P_n.N_{col}\right\} \tag{11}$$

Where:

- $NC_n$ is the primary quantity of colonies of the *n-th* imperialist
- $N_{col}$ is the number of all colonies

$NC_n$ colonies are selected randomly and assigned to one imperialist. Apparently, the grater quantity of colonies, the stronger imperialist and the less quantity of colonies, the poorer imperialist.

3.3.3.2 Total Strength of an Empire

Imperialist country has the major impact on the total strength of an empire. But the strength of its colonies does not have such effect. Therefore, the equation of the total power of an empire is shown below. (Karimi et al. 2010, Shokrollahpour et al. 2011).

$$TP\ Emp_n = \left(Total\ Cost(imperialist_n) + \xi mean\{Total\ Cost(colonies\ of\ empire_n)\}\right)\left(1 - QE_n\right) \tag{12}$$

Where:

- $TP\ Emp_n$ is the total power of the nth empire
- zeta ( $\xi$ ) is a positive number which is considered to be less than 1
- Total cost of imperialists and colonies are calculated by Eq.8 and Eq.9.
- $QE_n$ is the quality of empire *n*th

$QE_n$ is determined as below:

First: all of the imperialists and colonies are accumulated and then the non-dominated solutions are chosen. The percentage of the non-dominated solution belonging to each empire is calculated as $QE_n$.

Remember that, the total strength of the empire to be determined by just the imperialist when the value of $\xi$ is small and increasing it will increase the role of the colonies in determining the total power of an empire.

3.3.3.3 Moving the Colonies of an Empire toward the Imperialist (Assimilating)

Having distributed colonies among imperialists, the imperialist and relevant colonies go together. Figure 7 shows the movement. There, d is the distance between imperialist and colony. $X$ is a random variable with a uniform (or any proper) distribution between 0 and $\beta \times d$ and $\beta$ is a number greater than 1. Direction of the movement is shown by $\theta$, which is a uniform distribution between $-\gamma$ and $\gamma$.

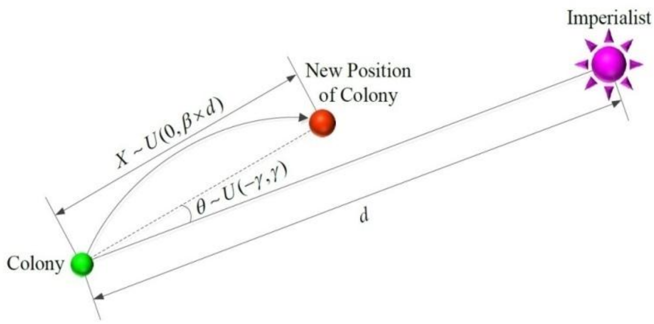



Figure 7. Moving colonies toward the imperialist with a random angle

3.3.3.4 Information Sharing between Colonies

To improve their position, the colonies share their information. To do so, in this part, one of these operators including one-point, two-point and continuous uniform crossover shown in Figure 8 are selected randomly. The mass part that are sharing information are shown by *Pc*. Those colonies with stronger position have more opportunity to share their information since the selection here is by a competition that is describe below:

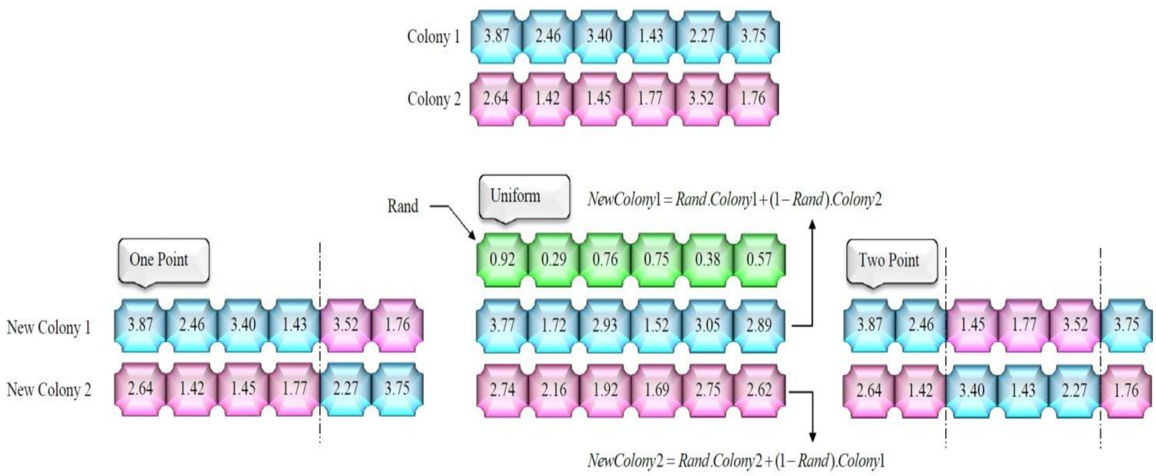



Figure 8. Operators for information sharing between colonies

How come a selection is done?

A binary competition process is used to find out the best solution for both crossover and mutation operators in this way:

Step 1: choose two solutions with the same size

Step 2: the lowest front number needs to be chosen if both populations are from different fronts.

If both of them are in same front, choose the solution with highest crowding distance.

3.3.3.5 Revolution

Some revolutions have been done during last decades on the colonies. To do so, one of the famous policies, swap, reversion and insertion, exchange and local search is randomly chosen. The operators' structures are elaborated as follows: (*Pr* is the revolution rate)

- Swap: a colony's initial numbers are chosen randomly (numbers 1.46 & 2.27 in Figure 9) and their substitute each other.


Figure 9. Swap operator

Reversion: in the current strategy, in addition to conducting substitution, the number that is placed between the substituted numbers are also changed.


Figure 10. Reversion

Insertion: for the insertion strategy, like substitution, two numbers of a colony is chosen by chance (Numbers 1.46 & 2.27 in Figure 11). After that, the next number is places approximate to the number in the first position. The other number is moved right side consequently.

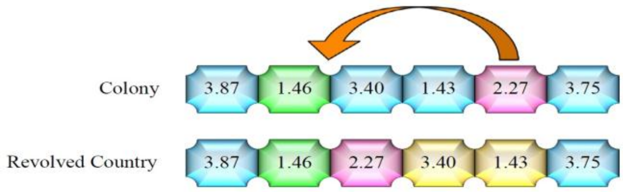



Figure 11. Insertion

Perturbation: in perturbation strategy, one number is chosen by chance, and another number is generated by chance. Then these two are substituted.

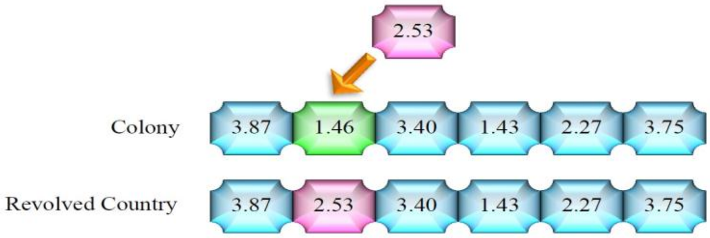



Figure 12. Perturbation

Local search: the strategy refers to randomly selection of the iteration. After that all of the two-point exchanges are investigated.

3.3.3.6 Improve Imperialist

Using the rules describe earlier, the step involves in producing some neighborhoods around each imperialist considering its total power. The less imperialist power, the more neighborhood generation. Similarly, the more imperialist power, the less neighborhood generation. In other word, the quantity of neighborhoods directly depends on the power of the given imperialist. This is a leaner formula ranged between $Ne_{min}$ and $Ne_{max}$. that is illustrated below:

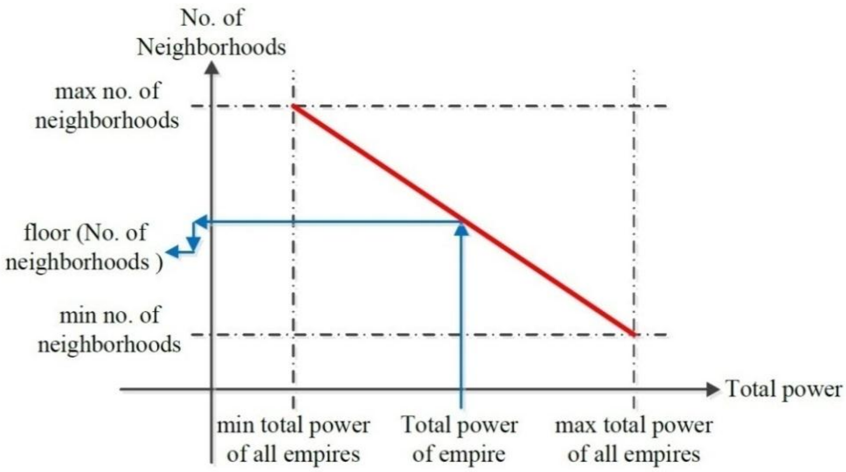
$$Ne_n = floor(Ne_{\min} + \frac{(Ne_{\max} - Ne_{\min})(TP\ Emp_{worst} - TP\ Emp_n)}{(TP\ Emp_{worst} - TP\ Emp_{best})}) \quad (13)$$

Where:

- $TP\ Emp_{best}$ is the value of total power the most power empire
- $TP\ Emp_{worst}$ is the value of total power of the weakest power empire.

How the number of neighborhood and power are related are shown below:

Figure 13. The relationship between power of each imperialist and number of neighborhoods




3.3.3.7 Colonies Updated

The primary mass of colonies, assimilating, information sharing among colonies, revolution and improve imperialist are combined all together in each decades to shape the empire that is called combined mass. Then, based on combined mass, the archive is updated. Then, for any of the imperialists, the best colony is chosen according to non-domination sorting and crowding distance by size of mass of colonies for a given empire (*NC(i)*).

3.3.3.8 Archive Adaption

For the combined mass, the classification is done using non-dominated and crowding distance. To archive, the front one members are chosen. At last, these members are retained and after classifying the solutions in archive, the other members are removed. Meanwhile, the size of archive equals *n Archive*.

3.3.3.9 Exchanging Positions of the Imperialist and a Colony

In this step, the total cost of each imperialist is updated. Next, the best imperialist and colony are combined. Then, this mass is arranged by the non-dominated sorting and crowding distance. At last, the best mass is chosen as imperialist. The step is illustrated below.

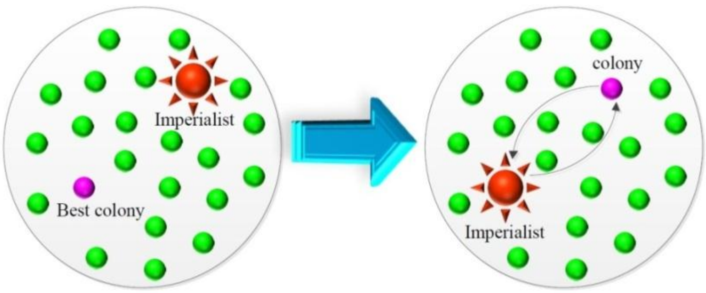



Figure 14. Exchanging positions of the imperialist and a colony

3.3.3.10. Imperialistic Competition

This term refers to a match among imperialist in which the weaker the territories, the more reduction of power would be and the more powerful the territories, the more power it gains. The competition is to take the hegemony of the weakest colony of the weakest territory. The competition is started by first choosing one or more colonies that are the weakest. Then the hegemony of these weak colonies is taken by a stronger territory through the competition. So far, this does not necessarily mean that the strongest territory is the winner. This means that those that are stronger keep more hegemony. This competition is modeled by just selecting one of the weakest colonies of the weakest territory to formulate the hegemony of each territory first is obtained the normalized total cost as follows.

$$NTP\ Emp_n = \max\{TP\ Emp_i\} - TP\ Emp_n \tag{14}$$

Where:

- $NTP_n$ is the normalized total power of nth empire
- $TP_n$ is the total power of nth empire

After calculating normalized total power, the hegemony probability of each territory is obtained by:

$$p_{Pn} = \left| \frac{NTP\ Emp_n}{\sum_{i=1}^{N_{imp}} NTP\ Emp_i} \right| \tag{15}$$

Next, to allocate the abovementioned colony to a territory, a so called roulette wheel method is used that is shown below:

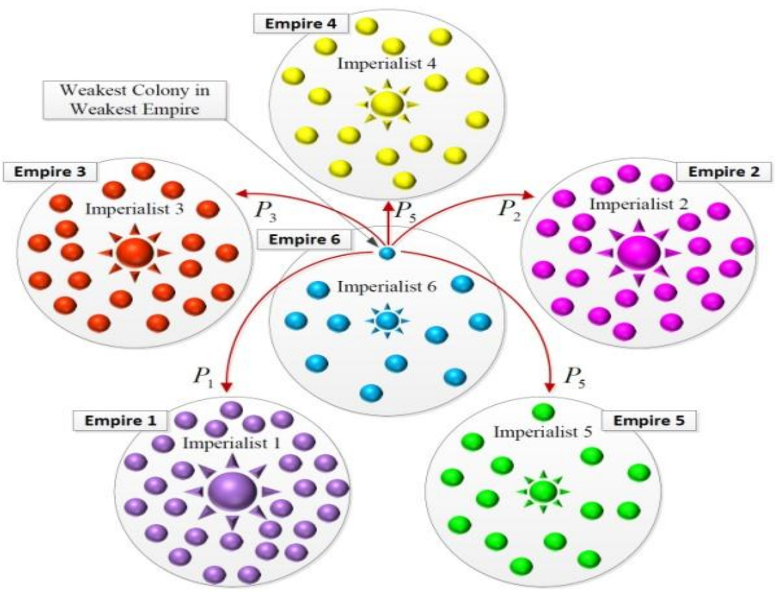



Figure 15. Imperialistic competition

3.3.3.11 Eliminating the Powerless Empires

Through the competition, weak territories will ruin and their members dispense among other territories. The study refers to this ruined territory as collapses as shown below:

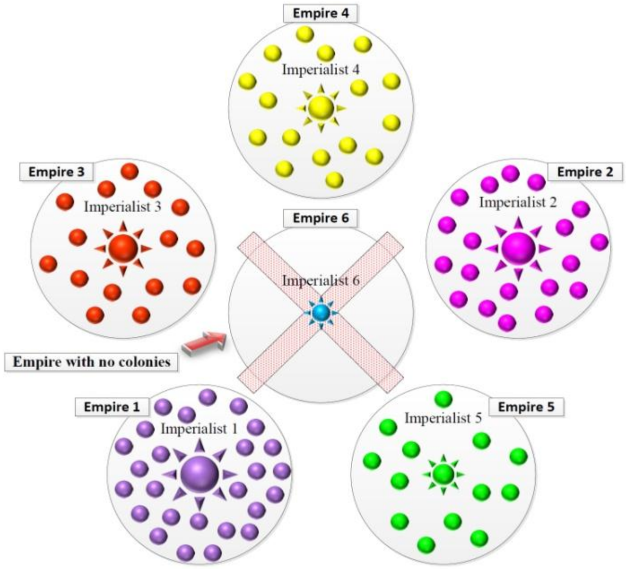



Figure 16. Eliminating the powerless empires

3.3.3.12 Stopping Criteria

The stopping point in competition, in this study, refers to the situation where there is only one territory remained among all countries. The process of territories purification is shown in three spectrums below:

Figure 15. Convergence of algorithm

The process that the study is offered is concisely illustrated below:

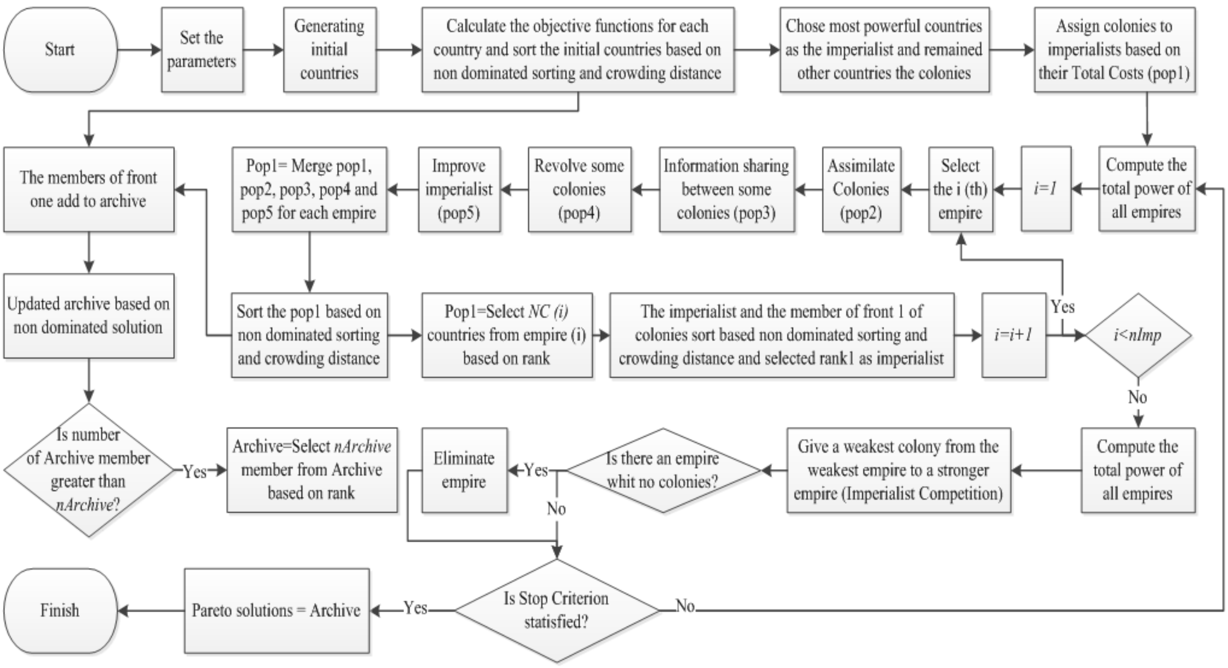



Figure 16. Flowchart of MOICA

## 4. Computational Experiments

### *4.1 Problem Design*

The paper investigates the influence of some approaches for 36 test problems. Data are classified in three segments; quantity of operations, quantity of machines in first and second stages and, the dissemination of operation time in those stages. On the other hand, the abovementioned problems has been classified in two major segments; small and large problems. Table 1 depicts the quantity of operations and machine in small and large scale.

Table 1. Factors and their levels

| Factors | Levels |
|---|---|
| Number of jobs $(N)$ | Small:8,10,14,16,20,24<br>Large:72, 80, 88, 108, 120, 132 |
| No. Machines (in both stages) | M1=3,M2=4; M1=2,M2=2; M1=3,M2=2<br>M1=8,M2=10; M1=10,M2=10; M1=12,M2=10 |
| Processing times $(p_{i,j})$ | $U(4,40)$ |
| Sequence dependent setup times $(S_{jki})$ | $U(4,40)$ |
| Probability of rework $(P_{re})$ | Exponential distribution $(\lambda e^{-\lambda})$ with mean equal to 0.05 |
| Rework times $(RT_{i,j})$ | $Round\left(U(0.3,0.6)\right.$請$\left._{i,j}\right)$ |

In addition the due dates are generated using the following formula:

$$(p_{1j}+p_{2j})+round\left(\frac{U(0,\sum_{j=1}^{n}(p_{1j}+p_{2j})}{(m_1+m_2)}\right) \tag{16}$$

*4.2 Parameter Setting*

To evaluate the performance the suggested process, we need to set some key success factors. To come to this conclusion, some operations are simulated for both sizes of problems. Table 2 shows the tuned values of the proposed algorithms' values.

Table 2. Tuned values of the parameters of the algorithms

| algorithm | parameter | problem size | |
|---|---|---|---|
| | | small | large |
| NSGAII | population size | 100 | 200 |
| | max generation | 200 | 500 |
| | crossover rate | 0.8 | 0.8 |
| | mutation rate | 0.2 | 0.2 |
| MOICA | Pop Size | 150 | 300 |
| | $N_{Imp}$ | 5 | 10 |
| | Maxdc | 250 | 400 |
| | $\varsigma$ | 0.4 | 0.4 |
| | $P_{AS}$ | 0.2 | 0.3 |
| | $P_R$ | 0.2 | 0.3 |
| PAES | max iteration | 50000 | 200000 |

*4.3 Performance Measures*

The so-called Pareto based multi-objective optimization algorithm aims to find an estimate of non-dominated front. The key performance indicators of these methods are different from the single-purpose method. Hence, another method is needed to evaluate the key performance indicators of this algorithm. Then to evaluate the output of multi-purpose algorithms quantitatively, below evaluation methods are applied:

Number of Pareto solutions (NPS): in this method, the number of non-dominated solutions resulted by an algorithm is computed.

Mean ideal distance (MID): The approximation between Pareto solutions and ideal point is defined. The calculation of MID is:

$$MID=\frac{\sum_{i=1}^{n}c_i}{n} \tag{17}$$

Where n is the quantity of non-dominated solutions and $c_i=\sqrt{f_{1i}^2+f_{2i}^2}$ . The lower the value, the better performance the MID has.

- *Diversification metric (DM)*: This performance metric shows the range of solutions resulted by algorithms and is calculated as follows:

$$DM=\sqrt{(\max f_{1i}-\max f_{1i})^2+(\max f_{2i}-\max f_{2i})^2} \tag{18}$$

- *The spread of non-dominance solutions (SNS)*: Regarding to MID, this computes the range of non-dominated solutions. The SNS is defined as follows:

$$SNS = \sqrt{\frac{\sum_{i=1}^{n}(MID - c_i)^2}{n-1}} \tag{19}$$

- *% Domination*: this key performance indicator applies a constructed Pareto combination set. Then the percentage of the solution belonging to each algorithm is calculated.

*4.4 Experimental Results*

This section refers to the result of experiments that are done through all the algorithms. The effectiveness of each algorithm is presented and compared in terms of key performance indicators. All algorithms were coded using MATLAB 2013a and run on personal computer with a 2.66 GHz CPU and 4 GB main memory.

The efficiency of the algorithms was stated by solving 36 variant problems of which 18 are small and 18 are large in scale. The outputs of three algorithms regarding the five key performance indicator for both size are compared and shown in Table 3 to 7 respectively.

Relative Percentage Deviation (RPD) is applied for the best solutions in terms of the key performance indicator. The calculation is shown below:

$$RPD = \frac{|Method_{sol} - Best_{sol}|}{Best_{sol}} \times 100 \tag{20}$$

Where

- $Method_{sol}$ is value of method
- $Best_{sol}$ is the best value between the algorithms

Table 4, 5 and 6 show the output with 95% sureness for the percentage of domination, DM and MID key performance indicator for small size respectively. Deep analysis shows that MOICA beats the others in terms of domination percent. Concerning DM, there is no considerable difference between NSGAII and MOICA. Yet the MOICA still beats PAES. The presented facts disclose that for MID key performance indicator, the three algorithms are similar and there is no considerable variance.

Table 7 presents the output of large scale problems. Alike the small scale problems, MOICA beats NSGAII and PEAS. The more size of the problem, the more considerable the advantages are.

Table 8, 9 and 10 also shows the outputs of algorithms in terms of domination percent in which DM and MID are shown consequently. By a glance at the table 8, MOICA's superiority in terms of performance in comparison to the next two. Outputs of DM are also illustrated in table 9 and disclose that the performance of NSGAII and MOICA identical and both of them outperform PAES. Moreover, when MID is considered, MOICA beats NSGAII and PEAS as shown in table 10.

Table 3. The simulation results for small size problems

| | | | | NPS | | | DM | | | % Domination | | | MID | | | SNS | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Row | N | M1 | M2 | MOICA | NSGAII | PAES | MOICA | NSGAII | PAES | MOICA | NSGAII | PAES | MOICA | NSGAII | PAES | MOICA | NSGAII | PAES |
| 1 | | 3 | 4 | **2** | **2** | 1 | **29.02** | 29.017 | 0 | **100** | **100** | 0 | **81.108** | **81.108** | 90 | **2.9816** | **2.9816** | 0 |
| 2 | 8 | 2 | 2 | **4** | 3 | 2 | 36.5 | **40.447** | 39.01 | **100** | 25 | 0 | **121.23** | 122.11 | 131.92 | 0.7465 | 2.4448 | **9.911** |
| 3 | | 3 | 2 | 3 | **4** | 2 | 16.16 | **21.26** | 9.22 | **100** | 66 | 33 | **117.77** | 121.11 | 120.17 | 2.7998 | **6.9985** | 1.174 |
| 4 | | 3 | 4 | **1** | **1** | **1** | **0** | **0** | **0** | 0 | **100** | 0 | 103 | **101** | 104 | 0 | 0 | 0 |
| 5 | 10 | 2 | 2 | **5** | 3 | 3 | **14.21** | 10.77 | 7.211 | **100** | 20 | 0 | **155.22** | 157.1 | 161.72 | 3.438 | **4.912** | 1.686 |
| 6 | | 3 | 2 | 2 | **3** | 2 | **15.03** | 10.296 | 5.099 | **50** | **50** | 0 | **144.89** | 148.45 | 152.94 | 0.1562 | **4.6258** | 0.157 |
| 7 | | 3 | 4 | **3** | **3** | 1 | **73.06** | 3.6056 | 0 | **100** | 0 | 0 | 117.37 | **109.35** | 114.28 | **12.762** | 1.5098 | 0 |
| 8 | 14 | 2 | 2 | **9** | 5 | 2 | **46.17** | 45.354 | 3.162 | **88.89** | 11.11 | 0 | **168.26** | 169.08 | 184.05 | 2.4101 | **3.3383** | 0.004 |
| 9 | | 3 | 2 | 5 | 3 | 5 | **97.33** | 6.3246 | 40.79 | **57.14** | 42.86 | 0 | 147.51 | **136.07** | 153.8 | **12.833** | 0.9123 | 2.234 |
| 10 | | 3 | 4 | **19** | 13 | 3 | 55.9 | **63.6** | 25.94 | **68.75** | 31.25 | 0 | **150.13** | 150.4 | 153.8 | 2.931 | **5.1041** | 2.234 |
| 11 | 16 | 2 | 2 | **18** | 9 | 2 | 43.19 | **46.519** | 17.89 | **66.67** | 33.33 | 0 | **227.9** | 231.54 | 243.18 | 4.1149 | **10.872** | 6.583 |
| 12 | | 3 | 2 | **9** | **9** | 5 | 67.08 | **72.277** | 26.4 | **100** | 0 | 0 | **184.03** | 188.79 | 187.76 | 5.2406 | **6.5978** | 6.213 |
| 13 | | 3 | 4 | **8** | 3 | 5 | **106.2** | 27.074 | 64.63 | **85.71** | 14.29 | 0 | 185.49 | **179.32** | 196.89 | **11.403** | 0 | 7.002 |
| 14 | 20 | 2 | 2 | 4 | **6** | 5 | 37.12 | 70.007 | **117.7** | 20 | **80** | 0 | **279.59** | 283.57 | 310.07 | 2.9626 | 7.8095 | **18.56** |
| 15 | | 3 | 2 | 2 | 3 | **4** | 6.083 | 10.77 | **11.4** | **100** | 0 | 0 | **236.65** | 240.1 | 241.9 | 0.5632 | 1.3862 | **3.668** |
| 16 | | 3 | 4 | **9** | **9** | 6 | **80.23** | 63.561 | 37.74 | 0 | **100** | 0 | 327.85 | **209.1** | 222.12 | **11.212** | 5.2128 | 9.336 |
| 17 | 24 | 2 | 2 | **9** | 6 | 5 | **93.05** | 10.63 | 73.5 | **100** | 0 | 0 | **211.44** | 321.09 | 352.86 | 7.6681 | 1.5899 | **13.52** |
| 18 | | 3 | 2 | **6** | 4 | 4 | 23.41 | 6.4031 | **74.11** | **100** | 0 | 0 | **266.5** | 266.99 | 299.45 | 1.4816 | 1.5954 | **13.8** |
| AVERAGE | | | | **6.556** | 4.944 | 3.222 | **46.65** | 29.884 | 30.77 | **74.29** | 37.44 | 1.833 | 179.22 | **178.68** | 190.05 | 4.7614 | 3.7717 | **5.337** |
| N.opt | | | | **14** | 8 | 2 | **10** | 7 | 4 | **14** | 5 | 0 | **13** | 6 | 0 | 5 | **8** | 5 |

Table 4. 95% confidence interval for % domination in small size problems

```
                        Individual 95% CIs For Mean Based on
                        Pooled StDev
Level    N   Mean   StDev  +---------+--------+--------+--------
MOICA    18  15.28  35.50   (----*----)
NSGAII   18  54.97  41.66                (----*----)
PAES     18  98.17   7.78                               (----*----)
                           +---------+--------+--------+--------
                           0         30       60       90
```

Table 5. 95% confidence interval for DM in small size problems

```
                                 Individual 95% CIs For Mean Based on
                                 Pooled StDev
Level     N    Mean   StDev   -+---------+---------+---------+-------
MOICA    17   14.34   23.68    (-------*------)
NSGAII   17   33.37   38.95              (-------*------)
PAES     17   48.48   33.74                       (-------*-------)
                              -+---------+---------+---------+-------
                               0        20        40        60
```

Table 6. 95% confidence interval for MID in small size problems

```
                                 Individual 95% CIs For Mean Based on
                                 Pooled StDev
Level     N    Mean   StDev   -----+---------+---------+---------+----
MOICA    18    4.33   13.35     (-----------*----------)
NSGAII   18    3.72   12.05   (-----------*-----------)
PAES     18   10.06   14.67                 (-----------*-----------)
                              -----+---------+---------+---------+----
                                  0.0       5.0      10.0      15.0
```

Table 7. The simulation results for large size problems

| Row | N | M1 | M2 | NPS MOICA | NPS NSGAII | NPS PAES | DM MOICA | DM NSGAII | DM PAES | % Domination MOICA | % Domination NSGAII | % Domination PAES | MID MOICA | MID NSGAII | MID PAES | SNS MOICA | SNS NSGAII | SNS PAES |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 72 | 8 | 10 | 13 | 5 | 2 | 69.64 | 19.723 | 1.414 | 91.67 | 8.333 | 0 | 254.43 | 255.71 | 261.74 | 6.4602 | 2.2159 | 0.389 |
| 2 | | 10 | 10 | 7 | 6 | 3 | 39 | 19.698 | 6.325 | 100 | 0 | 0 | 207.95 | 215.54 | 225.05 | 3.7335 | 1.4171 | 2.592 |
| 3 | | 12 | 10 | 8 | 4 | 2 | 30.61 | 25.397 | 4.123 | 100 | 0 | 0 | 187.14 | 199.16 | 229.2 | 6.6512 | 6.6722 | 2.214 |
| 4 | 80 | 8 | 10 | 2 | 4 | 3 | 1.414 | 7.2111 | 7.616 | 100 | 0 | 0 | 241.45 | 257.83 | 296.67 | 0.5681 | 0.9049 | 2.443 |
| 5 | | 10 | 10 | 2 | 7 | 2 | 31.78 | 31.906 | 4.123 | 100 | 0 | 0 | 206.62 | 226.36 | 239.89 | 4.8939 | 3.7261 | 0.523 |
| 6 | | 12 | 10 | 5 | 4 | 2 | 22.85 | 7.8102 | 4.123 | 100 | 0 | 0 | 198.75 | 204.55 | 218.36 | 8.7505 | 0.5959 | 2.456 |
| 7 | 88 | 8 | 10 | 7 | 10 | 3 | 39.56 | 39.446 | 5 | 100 | 0 | 0 | 289.75 | 303.65 | 305.16 | 2.5704 | 3.6747 | 1.28 |
| 8 | | 10 | 10 | 2 | 4 | 2 | 8.062 | 28.284 | 3.162 | 100 | 0 | 0 | 238.63 | 260.54 | 270.76 | 0.3482 | 3.2601 | 1.666 |
| 9 | | 12 | 10 | 9 | 7 | 2 | 23.71 | 44.385 | 1.414 | 100 | 0 | 0 | 232.16 | 243.69 | 241.74 | 6.8892 | 3.5266 | 0.374 |
| 10 | 108 | 8 | 10 | 5 | 6 | 4 | 19.42 | 32.757 | 10.44 | 100 | 0 | 0 | 359.22 | 383.99 | 386.44 | 1.9322 | 3.2398 | 3.556 |
| 11 | | 10 | 10 | 5 | 6 | 2 | 12.81 | 14.142 | 1.414 | 100 | 0 | 0 | 297.39 | 329.02 | 328.93 | 3.0684 | 2.9776 | 0.312 |
| 12 | | 12 | 10 | 4 | 4 | 2 | 32.39 | 19.723 | 2.236 | 100 | 0 | 0 | 269.43 | 298.69 | 293.28 | 1.5514 | 2.6138 | 1.026 |
| 13 | 120 | 8 | 10 | 7 | 5 | 3 | 84.59 | 36.401 | 4.243 | 100 | 0 | 0 | 421.58 | 425.95 | 451.1 | 12.122 | 4.1824 | 0.654 |
| 14 | | 10 | 10 | 1 | 8 | 3 | 0 | 65.46 | 40.11 | 100 | 0 | 0 | 342.47 | 371.91 | 397.67 | 0 | 7.7346 | 10.44 |
| 15 | | 12 | 10 | 5 | 8 | 1 | 19.42 | 21.954 | 0 | 100 | 0 | 0 | 308.32 | 336.6 | 334.07 | 5.21 | 1.3778 | 0 |
| 16 | 132 | 8 | 10 | 5 | 10 | 1 | 30.68 | 26.833 | 0 | 100 | 0 | 0 | 430.48 | 480.45 | 475.26 | 1.4167 | 3.2612 | 0 |
| 17 | | 10 | 10 | 5 | 6 | 2 | 130.2 | 19.235 | 12.04 | 66.66 | 33.33 | 0 | 389.14 | 365.34 | 383.91 | 22.052 | 4.3077 | 7.471 |
| 18 | | 12 | 10 | 4 | 6 | 3 | 13.45 | 19.235 | 16.97 | 100 | 0 | 0 | 338.8 | 365.34 | 383.33 | 2.5377 | 4.3077 | 3.563 |
| AVERAGE | | | | 5.333 | 6.111 | 2.333 | 33.87 | 26.644 | 6.931 | 97.68 | 2.315 | 0 | 289.65 | 306.91 | 317.92 | 5.042 | 3.3331 | 2.276 |
| N.opt | | | | 7 | 12 | 0 | 9 | 8 | 1 | 18 | 0 | 0 | 17 | 1 | 0 | 9 | 6 | 3 |

Table 8. 95% confidence interval for % domination in large size problems

```
                         Individual 95% CIs For Mean Based on
                         Pooled StDev
Level    N    Mean   StDev   ---+---------+---------+---------+------
MOICA    20   0.00   0.00    (--*--)
NSGAII   20  92.47  22.04                                    (--*--)
PAES     20  95.00  22.36                                   (--*-)
                         ---+---------+---------+---------+------
                            0        30        60        90
```

Table 9. 95% confidence interval for DM in large size problems

```
                              Individual 95% CIs For Mean Based on
                              Pooled StDev
Level    N    Mean   StDev   -------+--------+--------+--------+--
MOICA    18   21.76  32.70   (-----*-----)
NSGAII   18   22.41  30.09   (-----*-----)
PAES     18   77.65  29.76                          (-----*-----)
                             -------+--------+--------+--------+--
                                    25       50       75       100
```

Table 10. 95% confidence interval for MID in large size problems

```
                               Individual 95% CIs For Mean Based on
                               Pooled StDev
Level    N     Mean   StDev    ----+--------+--------+--------+-----
MOICA    18    0.362  1.536    (----*----)
NSGAII   18    6.411  3.650                    (----*----)
PAES     18   10.692  5.756                               (----*----)
                               ----+--------+--------+--------+-----
                                  0.0      4.0      8.0      12.0
```

## 5. Conclusion and Further Researches

The study presents two criteria flow shop that is called two stage flexible flow shop. It aims to shorten the makespan and increase lateness of operations. The paper modeled Pareto optimal solutions and based the process on its similarity. NSGAII, MOICA and PEAS were suggested as the main three metaheuristic Pareto based multi-purpose algorithms. To estimate the efficiency of them, 36 problems, large and small, were answered. In multi-purpose norms, five key performance indicators, NPS, MID, DM, SNS and percentage domination were suggested to disclose the algorithms' efficiency. With similar efficiency, NSGAII beats PAES in both small and large scale of problem, in terms of DM. hence, MOICA is the best algorithm in case of efficiency for all the studied problems.

To Guidance for researchers in similar cases, using other effective metaheuristic algorithm like multi-purpose and colony optimization or multi-purpose invasive weed optimization are suggested to work on. Also there would be valuable result if the stages increase for more than two.

**Acknowledgments**

This research was supported by the National Natural Science Foundation of China (no. 70971020).

**References**

Al Jaddan, O., Rajamani, L., & Rao, C. R. (2008). Non-dominated ranked genetic algorithm for solving multi-objective optimization problem: NSGAII. *Journal of Theoretical and Applied Information Technology*, 4(1).

Aldowaisan T., & Allahverdi A. (2004). A New heuristics for m-machine no-wait flowshop to minimize total completion time. *Omega*, *32*, 345-352. https://doi.org/10.1016/j.omega.2004.01.004

Allahverdi, A., & Aldowaisan, T. (2004). No-wait flowshops with bicriteria of makespan and maximum lateness, *European Journal of Operational Research*, *152*, 132-147. https://doi.org/10.1016/S0377-2217(02)00646-X

Asefi, H., Jolai, F., Rabiee, M., &Araghi, M. T. (2014). A hybrid NSGA-II and VNS for solving a bi-objective no-wait flexible flowshop scheduling problem. *The International Journal of Advanced Manufacturing Technology, 75*(5-8), 1017-1033. https://doi.org/10.1007/s00170-014-6177-9

Atashpaz-Gargari, E., & Lucas, C. (2007). Imperialist competitive algorithm: An algorithm for optimization inspired by imperialist competitive. IEEE Congress on Evolutionary computation, Singapore 2007. https://doi.org/10.1109/CEC.2007.4425083

Bäck, T., Hoffmeister, F., & Schwefel, H. (1991). A survey of evolution strategy. *Proceedings of the 4th International Conference on Genetic Algorithms*, 2-9, (Morgan Kaufmann: San Mateo, CA).

Baker, K. R. (1974). *Introduction to sequencing and scheduling.* New York: John Wiley & Sons.

Behnamian, J., Ghomi, S. F., &Zandieh, M. (2009). A multi-phase covering Pareto-optimal front method to multi-objective scheduling in a realistic hybrid flowshop using a hybrid metaheuristic. *Expert Systems with Applications, 36*(8), 11057-11069. https://doi.org/10.1016/j.eswa.2009.02.080

CoelloCoello, C. A., Van Veldhuizen, D. A., & Lamont, G. B. (2002). *Evolutionary algorithms for solving multi-objective problems.* Singapore: Kluwer Academic. https://doi.org/10.1007/978-1-4757-5184-0

Deb, K. (2001). *Multiobjective optimization using evolutionary algorithms*. Chichester, UK: JohnWiley and Sons Ltd.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002a). Fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182-197. https://doi.org/10.1109/4235.996017

Eiben, A. E., & Smith, J. E. (2003). Introduction to evolutionary computing, Springer 1st edition, ISBN: 3-540-40184-9. https://doi.org/10.1007/978-3-662-05094-1

Furtuna, R., Curteanu, S., & Leon, F. (2011). An elitist non-dominated sorting genetic algorithm enhanced with a neural network applied to the multi-objective optimization of a polysiloxane synthesis process. *Engineering Applications of Artificial Intelligence, 24*(5), 772-785. https://doi.org/10.1016/j.engappai.2011.02.004

Ghiasi H., Pasini D., & Lessard L. (2011). A non-dominated sorting hybrid algorithm for multi-objective optimization of engineering problems, *Engineering Optimization, 43*(1), 39-59. https://doi.org/10.1080/03052151003739598

Grabowski, J., & Pempera, J. (2000). Sequencing of jobs in some production system. *European Journal of Operational Research*, *125,* 535-550. https://doi.org/10.1016/S0377-2217(99)00224-6

Huang R. H., Yang C. L., & Huang Y. C. (2009). No-wait two-stage multiprocessor flow shop scheduling with unit setup. *International Journal of Advanced Manufacturing Technology*, *44,* 921-927. https://doi.org/10.1007/s00170-008-1865-y

Hwang, C., & Yoon, K. (1981). *Multiple Attribute Decision Making: Methods and Applications*. Springer, Berlin. https://doi.org/10.1007/978-3-642-48318-9

Johnson, S. M. (1954). Optimal two and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly*, *1,* 61-68. https://doi.org/10.1002/nav.3800010110

Jolai, F., Rabiee, M., & Asefi, H. (2012) A novel hybrid meta-heuristic algorithm for a no-wait flexible flow shop scheduling problem with sequence dependent setup times. *Int J Prod Res, 50*(24), 7447-7466. https://doi.org/10.1080/00207543.2011.653012

Jolai, F., Sheikh, S., Rabbani, M., & Karimi, B. (2009). A genetic algorithm for solving no-wait flexible flow lines with due window and job rejection. *International Journal of Advanced Manufacturing Technology*, *42,* 523-532. https://doi.org/10.1007/s00170-008-1618-y

Jolai, F., Asefi, H., Rabiee, M., & Ramezani, P. (2013). Bi-objective simulated annealing approaches for no-wait two-stage flexible flow shop scheduling problem. *Scientia Iranica, 20(3),* 861-872. https://doi.org/10.1016/j.scient.2012.10.044

Jolai, F., Tavakkoli-Moghaddam, R., Rabiee, M., & Gheisariha, E. (2014). An enhanced invasive weed optimization for makespan minimization in a flexible flowshop scheduling problem. *Scientia Iranica. Transaction E, Industrial Engineering*, *21(3),* 1007.

Karimi, N., Zandieh, M., & Najafi, A. A. (2010). Group scheduling in flexible flow shops: a hybridised approach of imperialist competitive algorithm and electromagnetic-like mechanism. *International Journal of Production Research, 49*(16), 4965-4977. https://doi.org/10.1080/00207543.2010.481644

Khalili, M. (2012) Multi-objective no-wait hybrid flowshop scheduling problem with transportation times. *Int J ComputSciEng, 7*(2), 147-154. https://doi.org/10.1504/IJCSE.2012.048094

Khalili, M. (2013) A multi-objective electromagnetism algorithm for a bi-objective hybrid no-wait flowshop scheduling problem. Int J AdvManufTechnol, 1-11.

Kishor, A., Yadav, S. P., & Kumar, S. (2009). Interactive fuzzy multiobjective reliability optimization using NSGA-II. *Opsearch, 46*(2), 214-224. https://doi.org/10.1007/s12597-009-0013-2

Knowles, J. D., & Corne, D. W. (1999). Local Search, Multiobjective Optimization and the Pareto Archived Evolution Strategy. In B. Mckay, R. Sarker, X. Yao, Y. Tsujimura, A. Namatame, and M. Gen, editors, *Proceedings of The Third Australia-Japan Joint Workshop on Intelligent and Evolutionary Systems*, 209-216, Ashikaga, Japan, November 1999. Ashikaga Institute of Technology.

Knowles, J. D., & Corne, D. W. (1999). The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Multiobjective Optimization. *1999 Congress on Evolutionary Computation*, 98-105, Piscataway, NJ, July 1999. IEEE Service Center. https://doi.org/10.1109/CEC.1999.781913

Knowles, J. D., & Corne, D. W. (2000). Approximating the non-dominated front using the Pareto Archived Evolution Strategy. *Evolutionary Computation*, *8*(2). https://doi.org/10.1162/106365600568167

Lin, B. M. T., Lin, F. C., & Lee, R. C. T. (2006). Two-machine flow-shop scheduling to minimize total late work, *Engineering Optimization*, *38*(4), 501-509. https://doi.org/10.1080/03052150500420439

Liu Z., Xie J., Li J., & Dong, J. (2003). A heuristic for two-stage no-wait hybrid flowshop scheduling with a single machine in either stage. *Tsinghua Science and Technology, 8,* 43-48.

Minella, G., Ruiz, R., &Ciavotta, M. (2008). A review and evaluation of multiobjective algorithms for the flowshop scheduling problem. *INFORMS Journal on Computing*, *20*(3), 451-471. https://doi.org/10.1287/ijoc.1070.0258

Mosheiov, G., & Sidney, J. B. (2010). Scheduling a deteriorating maintenance activity on a single machine. *Journal of the Operational Research Society*, 61(5), 882-887. https://doi:10.1057/jors.2009.5

Nagano, M. S., & Miyata, H. H. (2016). Review and classification of constructive heuristics mechanisms for no-wait flow shop problem. *The International Journal of Advanced Manufacturing Technology,* 1-14. https://doi.org/10.1007/s00170-015-8209-5

Nagano, M. S., Da Silva, A. A., & Lorena, L. A. N. (2014). An evolutionary clustering search for the no-wait flow shop problem with sequence dependent setup times. *Expert Systems with Applications, 41*(8), 3628-3633. https://doi.org/10.1016/j.eswa.2013.12.013

Nagano, M. S., Miyata, H. H., & Araújo, D. C. (2015). A constructive heuristic for total flowtime minimization in a no-wait flowshop with sequence-dependent setup times. *Journal of Manufacturing Systems, 36,* 224-230. https://doi.org/10.1016/j.jmsy.2014.06.007

Nesello, V., Subramanian, A., Battarra, M., & Laporte, G. (2017). Exact solution of the single-machine scheduling problem with periodic maintenances and sequence-dependent setup times. *European Journal of Operational Research. forthcoming*

Pan Q. K., Wang L., & Qian, B. (2009). A novel differential evolution algorithm for bi-criteria no-wait flow shop scheduling problems, *Computers & Operations Research, 36,* 2498-2511. https://doi.org/10.1016/j.cor.2008.10.008

Qian, B., Wang, L., Huang, D. X., Wang, W. L., & Wang X. (2009). An effective hybrid DE- based algorithm for multi-objective flow shop scheduling with limited buffers, *Computers & Operations Research*, 36(1), 209-233. https://doi.org/10.1016/j.cor.2007.08.007

Raaymakers, W., & Hoogeveen, J. (2000). Scheduling multipurpose batch process industries with no-wait restrictions by simulated annealing. *European Journal of Operational Research, 126,* 131-151. https://doi.org/10.1016/S0377-2217(99)00285-4

Rabiee, M., Zandieh, M., & Jafarian, A. (2012). Scheduling of a no-wait two-machine flow shop with sequence-dependent setup times and probable rework using robust meta-heuristics. *International Journal of Production Research, 50(24),* 7428-7446.

Rabiee, M., Rad, R. S., Mazinani, M., & Shafaei, R. (2014). An intelligent hybrid meta-heuristic for solving a case of no-wait two-stage flexible flow shop scheduling problem with unrelated parallel machines. Int J

AdvManufTechnol, 1-17. https://doi.org/10.1007/s00170-013-5375-1

Rabiee, M., Zandieh, M., &Ramezani, P. (2012). Bi-objective partial flexible job shop scheduling problem: NSGA-II, NRGA, MOGA and PAES approaches. *International Journal of Production Research*, *50*(24), 7327-7342. https://doi.org/10.1080/00207543.2011.648280

Rabiee, M., Jolai, F., Asefi, H., Fattahi, P., & Lim, S. (2016). A biogeography-based optimisation algorithm for a realistic no-wait hybrid flow shop with unrelated parallel machines to minimise mean tardiness. *International Journal of Computer Integrated Manufacturing, 29(9)*, 1007-1024.

Rahimi-Vahed, A. R., Javadi, B., Rabbani, M., & Tavakkoli-Moghaddam, R. (2008). A multi-objective scatter search for a bi-criteria no-wait flow shop scheduling problem, *Engineering Optimization*, *40*(4), 331-346. https://doi.org/10.1080/03052150701732509

Rajendran, C. (1994). A no-wait flow shop scheduling heuristic to minimize makespan. *Journal of the Operational Research Society*, *45*, 472-478. https://doi.org/10.1057/jors.1994.65

Ramezani, P., Rabiee, M., & Jolai, F. (2013) No-wait flexible flowshop with uniform parallel machines and sequence-dependent setup time: a hybrid meta-heuristic approach. J IntManuf, 1-14.

Ribas, I., Leisten, R., & Framinan, J. M. (2010). Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective. *Computers & Operations Research*, *37*, 1439-1454. https://doi.org/10.1016/j.cor.2009.11.001

Richard, L., & Zhang, W. (1999). Hybrid flow shop scheduling - A survey. *Computer and industrial engineering*, *37*, 57-61. https://doi.org/10.1016/S0360-8352(99)00023-6

Ruiz, R., & Vazquez-Rodriguez, J. A. (2010). The hybrid flow shop scheduling problem. *European Journal of Operational Research*, *205*, 1-18. https://doi.org/10.1016/j.ejor.2009.09.024

Samarghandi, H., & ElMekkawy, T. Y. (2012). A genetic algorithm and particle swarm optimization for no-wait flow shop problem with separable setup times and makespan criterion. *The International Journal of Advanced Manufacturing Technology, 61*(9-12), 1101-1114. https://doi.org/10.1007/s00170-011-3766-8

Samarghandi, H., &ElMekkawy, T. Y. (2014). Solving the no-wait flow-shop problem with sequence-dependent set-up times. *International Journal of Computer Integrated Manufacturing, 27*(3), 213-228. https://doi.org/10.1080/0951192X.2013.812802

Shafaei, R., Moradinasab, N., & Rabiee, M. (2011). Efficient meta heuristic algorithms to minimize mean flow time in no-wait two stage flow shops with parallel and identical machines. *International Journal of Management Science and Engineering Management, 6(6)*, 421-430.

Shokrollahpour, E., Zandieh, M., & Dorri, B. (2011). A novel imperialist competitive algorithm for bi-criteria scheduling of the assembly flowshop problem. *International Journal of Production Research*, *49*(11), 3087-3103. https://doi.org/10.1080/00207540903536155

Sivakumar, K., Balamurugan, C., &Ramabalan, S. (2011). Simultaneous optimal selection of design and manufacturing tolerances with alternative manufacturing process selection. *Computer-Aided Design*, *43*(2), 207-218. https://doi.org/10.1016/j.cad.2010.10.001

Sriskandarajah, C., & Ladet, P. (1986). Some no-wait shops scheduling problems: Complexity aspects. *European Journal of Operational Research*, *24*, 424-445. https://doi.org/10.1016/0377-2217(86)90036-6

Steuer, R. E. (1986). *Multiple Criteria Optimization: Theory, Computation, and Application*. John Wiley and Sons, Inc., New York.

Syswerda, G. (1989). Uniform crossover in genetic algorithms. *Proceedings of the Third International Conference on Genetic Algorithms*, edited by J. Schaffer, pp. 2–9, (Morgan Kaufmann: San Mateo, CA).

Tasgetiren, M. F., Pan Q. K., Suganthan, P. N., & Liang, Y. C. (2007). A Discrete Differential Evolution Algorithm for the No-Wait Flow shop Scheduling Problem with Total Flow time Criterion. *Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Scheduling*. https://doi.org/10.1109/SCIS.2007.367698

Tavakkoli-Moghaddam, R., Rahimi-Vahed, A., & Mirzaei, A. H. (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*(22), 5072-5790. https://doi.org/10.1016/j.ins.2007.06.001

Tayebi Araghi, M. E., Jolai, F., & Rabiee, M. (2014). Incorporating learning effect and deterioration for solving a SDST flexible job-shop scheduling problem with a hybrid meta-heuristic approach. *International Journal of Computer Integrated Manufacturing, 27(8),* 733-746.

Vallada, E., & Ruiz, R. (2011). A genetic algorithm for the unrelated parallel machine scheduling problem with sequence dependent setup times. *European Journal of Operational Research, 211(3),* 612-622.

Xie, J., Xing, W., Liu, Z., & Dong, J. (2004). Minimum Deviation Algorithm for Two-Stage No-Wait Flowshops with Parallel Machines. Computer *and Mathematics with Application*, *47,* 1857-1863. https://doi.org/10.1016/j.camwa.2003.08.003

Xu, D., Wan, L., Liu, A., & Yang, D. L. (2015). Single machine total completion time scheduling problem with workload-dependent maintenance duration. *Omega, 52,* 101-106.

Yang, D. L., Cheng, T. C. E., & Yang, S. J. (2014). Parallel-machine scheduling with controllable processing times and rate-modifying activities to minimise total cost involving total completion time and job compressions. *International Journal of Production Research, 52(4),* 1133-1141.

Zandieh, M., &Karimi, N. (2011). An adaptive multi-population genetic algorithm to solve the multi-objective group scheduling problem in hybrid flexible flowshop with sequence-dependent setup times. *Journal of Intelligent Manufacturing, 22*(6), 979-989. https://doi.org/10.1007/s10845-009-0374-7

Zhou, G., Min, H., & Gen, M. (2003). A genetic algorithm approach to the bi criteria allocation of customers to warehouses. *International Journal of Production Economics, 86,* 35-45. https://doi.org/10.1016/S0925-5273(03)00007-0