International Journal of Academic Research
ISSN: 2348-7666 : Vol.2, Issue-4(4), October-December, 2015
Impact Factor : 1.855

A multi-objective stochastic programming model for hybrid flow shop scheduling problem with uncertain processing time

M.Ebrahimi, Department of Industrial Engineering, Amirkabir University of Technology, 424 Hafez Avenue, 15916-34311, Tehran, Iran

Abstract: *This paper develops a new mathematical model for hybrid flow shop scheduling (HFS) problem in uncertain environment. The flow shop scheduling problem is made up of n jobs that have to be processed on m machines. But a HFS problem should have more than one machine in at least one stage. To make this problem more realistic, it is assumed that the processing time is stochastic and family setup time is sequence-dependent. Also to approach to the problems of real world, a bi-objective problem is considered. In other words, the goal of this paper is to find the best sequence of groups and jobs inside the groups to minimize the makespan and total tardiness. Firstly, the problem is modeled based on chance-constraint. Then, by changing the problem to a deterministic problem and non-linear constraints to linear constraints, a linear problem is solved. Since this problem is NP-hard, in order to obtain the pareto solutions set, two meta-heuristic algorithms based on Genetic Algorithm (GA) are developed, namely: Sub-Population Genetic Algorithm (SPGA-II) and Non-dominated Sorting Genetic Algorithm (NSGA-II). The quality of the solution sets obtained from the algorithms is evaluated by some appropriate performance evaluation metrics. Experimental results show that algorithms proposed for the problem have high performance.*

Keywords: Chance-constraint method, Metaheuristic algorithms, Multi-objective scheduling optimization, Sequence-dependent setup time, Stochastic programming.

1. Introduction: Since the flow shop scheduling problem was first introduced by Johnson (1954), many studies have been conducted in this field. In the classical flow shop problem, a set of jobs flows through all stages in the same order. In each stage, there is only one machine. The first paper in the field of hybrid flow shop scheduling (HFS) problem was published by Salvador (1973). HFS problems are the extended form of flow shop problems. In HFS problems, there is at least one stage that has at least two parallel machines. HFS problems are classified into different categories according to machines characteristics. In this paper, the HFS problem with identical parallel machines is studied. In this type of problems, all machines existing in each stage are considered to be similar. So, the processing time of a job in each stage does not dependent the specific machine assigned to that job. In this paper, it is assumed that parts being similar in terms of processing procedure or characteristics physical, are placed in a same group. In this condition, the setup time is considered only for one job of each group (the first job of each group being processed). So, in this paper, the sequence-dependent family setup time assumption is considered.

Many problems of real environment include the simultaneous optimization of different objective functions. These objective functions are in competition and conflict with themselves Tavakkoli Moghadam et Al. (2007). So, it is impossible to find a solution that its objective functions are better compared to other solutions. In multi-objective problems, some solution sets are obtained that by taking all objective functions into consideration, are better compared to other existing solution sets. These solutions are called pareto-optimal. In this paper, the following objective functions are considered to be minimized:

f_1 =makespan: $C_{max} = max\{C_i\}$,$i = 1,\ldots,n$

f_2 =total weighted tardiness= $TWT = \sum_{i=1}^{n} W_i T_i$

Where$T_i = max(0, C_i - d_i)$, d_i is the due dates of job i, W_i is a weight related to job i, n is the number of jobs.

Many uncertain events happen in production environment, such as changing the due date, processing time, availability of resources, machine breakdown… Hence, considering the scheduling problems in uncertain situations makes the problem conditions closer to the real environment. In this paper, the processing time is assumed to be stochastic and its data follow normal distribution. Also, the chance-constraint method is used to model the problem. To solve the model, two meta-heuristic algorithms called Sub-Population Genetic Algorithm (SPGA-II) and Non-dominated Sorting Genetic Algorithm (NSGA-II) based on Genetic Algorithm (GA), are developed. The paper is organized as follows. In section 2, a review on scheduling problems with considered assumptions is made. Section 3 introduces the mathematical model and section 4 gives a description of two metaheuristic algorithms. Section 5 gives the results obtained from the algorithms implementing. Finally, section 6is devoted to conclusions and recommendations for future studies.

2. Literature review

Rahmani and Heydari(2014) studied a new approach to achieve stable and robust schedule despite uncertain processing times and unexpected arrivals of new jobs. This approach was a proactive–reactive method which used a two-step procedure. In the first step an initial robust solution was produced proactively against uncertain processing times using robust optimization approach. In the next step, when an unexpected disruption occurs, an appropriate reactive method was adopted to deal with this unexpected event.

Several heuristic and metaheuristic algorithms have been suggested to solve multi-objective scheduling problem. Wei-Chang Yeh (2014) expressed parallel machine scheduling with learning effects. The objective was to minimize the makespan. To satisfy reality, they considered the processing times as fuzzy numbers. The possibility measure used to rank the fuzzy numbers. Two heuristic algorithms, the simulated annealing algorithm and the genetic algorithm, were proposed. Computational experiments have been conducted to evaluate their performance. Jun-qing Li and Quan-ke Pan (2015) presented a novel hybrid algorithm (TABC) that combines the artificial bee colony (ABC) and tabu search (TS) to solve the hybrid

flow shop (HFS) scheduling problem with limited buffers. The objective is to minimize the maximum completion time. A novel decoding method is embedded to tackle the limited buffer constraints in the schedules generated. A TS-based self-adaptive neighborhood strategy was adopted to impart to the TABC algorithm a learning ability for producing neighboring solutions in different promising regions. Furthermore, a well-designed TS-based local search was developed to enhance the search ability of the employed bees and onlookers. Through a detailed analysis of the experimental results, the highly effective and efficient performance of the proposed TABC algorithm was contrasted with the performance of several algorithms reported in the literature.

3. Stochastic modeling: This section deals with the modeling the hybrid flow shop scheduling problem considering the sequence-dependent family setup times and stochastic processing times. The chance-constrained method is used to model the problem. After stochastic modeling, firstly, the stochastic multi-objective model changes into a deterministic multi-objective model by transforming stochastic constraints into deterministic equivalents. Then, the nonlinear constraints of the problem are converted into linear constraints by linearization methods. So, the final model proposed in this paper is a model with deterministic and linear constrains which is solved in the next section

3.1. Chance-constraint programming

In this paper, the processing times are considered to be uncertain. It is assumed that the stochastic variables follow normal distribution. So, constraints including stochastic variables are called chance constraints, in other words, these constraints are not always satisfied. As an example, consider the following constraint:

$X \geq P$,P is a stochastic variable,

Using a satisfactory level 1-ε selected by a decision maker, the above constraint can be changed as follows:

$P_r [X \geq P] \geq 1-\varepsilon$,

The constraint is rewritten as $X \geq F_p^{-1}(1-\varepsilon)$, where $F_p^{-1}(1-\varepsilon)$ is the inverse distribution function of P.

3.2. Notations

The following notations are used to develop the model.

n : number of jobs to be scheduled
g: number of serial stages
n_G: number of groups
g_i: last stage visited by job i, $i \in \{1, \ldots, n\}$
m^t:the number of machines in parallel at stage t, $t \in \{1, \ldots, g\}$

S_l : set of stages visited by group l , $l \in \{1, \ldots, n_G\}$
e_i: set of stages visited by job i , $i \in \{1, \ldots, n\}$
S^t : set of groups visited by stage t ,

P_k^t: processing time for all jobs in group k at stage t(assumed to be stochastic), $k \in \{1, \ldots, n_G\}$

P_{ij}^t : processing time for job j,if job j is processed immediately after job i(assumed to be stochastic) , $i,j \in \{1, \ldots, n\}$

$\mu_{p_k^t}$:the mean of processing times for all jobs in group k at stage t ,

$\sigma_{p_k^t}$:the standard deviation of processing times for all jobs in group k at stage t ,

$\mu_{p_{ij}^t}$:the mean of processing times for job j, if job j is processed immediately after job i,

$\sigma_{p_{ij}^t}$:the standard deviation of processing times for job j, if job j is processed immediately after job i,

d_i: due date of job i ,

S_{lk}^t :setup time for group k, if group k is processed immediately after group l,l,$k \in \{1,\ldots, n_G\}$

C_i^t:completion time of job i at stage t ,

CG_l^t:completion time for group l at stage t ,

x_{ij}^t : 1 if job i is scheduled immediately before job j at stage t and 0 otherwise,

y_{lk}^t : 1 if group l is scheduled immediately before group k at stage t and 0 otherwise,

$(1-\varepsilon)$: probability value of chance constraint,

For jobs 0 and n+1, the processing time is considered to be zero. The end of processing of job 0 in each stage is considered to be the start of setup time in the same stage. Following inequality shows the constraint of the number of jobs in each stage considering machines in that stage.

$$|S^t| \geq m^t \ , t=1, \ldots, g \ , \ so \ n \geq max_{1,\ldots,g}\{m^t\},$$

3.3. Assumptions

This paper studies the hybrid flow shop scheduling problem considering sequence-dependent family setup times. In this problem, there are one or more machines in at least one stage. Following assumptions are considered to model the hybrid flow shop problem:

1. All of n jobs that must be scheduled have one flow in k stages. In other words, each job must be processed firstly in stage 1 then in stage 2 and so on.
2. In stage t, there are m^t machines (assume $m^t \geq 1$). In each stage, parallel machines are similar in terms of capacity and rate of processing jobs.
3. All jobs and machines are simultaneously available at the beginning of scheduling period.
4.
5. Job processing cannot be interrupted.
6. The buffer capacity in stages is unlimited before the first stage and after the last stage. So the machines are not blocked.
7. Machines are always available and never break down.
8. The travel times of jobs are negligible.
9. All analysis related to a hybrid flow shop scheduling problem are static.
10. Processing time is stochastic and its data follow normal distribution.
11. Each machine can process only one job at the same time.
12. As the processing of a group starts and as long as all jobs of the group are being processed, the operation cannot stop.

One of the characteristics of this research is the classification of jobs in groups for processing. This is performed in such a way that jobs are classified in different groups according to some pre-determined characteristics, resulting Group Technology (GT). Group technology is a management theory that its purpose is to group parts according to similarity of the production process or production characteristics or both of them.

With respect to the above assumptions, the stochastic model with nonlinear constraints is presented as follows.

3.4.Stochastic mathematical model

Constraint (1) defines objective function, includes minimizing f_1 the makespan (C_{max}) and f_2, total tardiness. Constraint set (2) shows that m^t machines are scheduled in each stage. Both constraint sets (3) and (4) guarantee that processing of each group is performed only on one machine in each stage. Inequalities (5) to (9) show four chance constraint sets. Constraint set (5) guarantees that job j is processed after job i with minimum processing time of job i and satisfactory level 1-ε .

Also, constraint set (6) guarantees with satisfactory level 1-ε that the processing completion time of job j at stage t would be completing its process at stage t-1, plus its processing time in stage t. M_j^t is calculated from the following equation, $M_j^t = P_j^t$. In fact, constraint sets (5) and (6) together indicate that as long as job j in stage t is not available, its setup time cannot be considered, in other words, this job must be completed in stage t-1 and its previous job (job i) must be completed in stage t. Similar to constraint sets (5) and (6) for jobs, constraint sets (7) and (8) for groups, together guarantee that as long as the groups are not available, the setup time cannot be started. In other words, in order to consider the setup time for group k in stage t, the group l in stage t and group k in stage t-1 must be completed; group k is scheduled after group l.

$$Min\ Z = min\ (f_1, f_2) \quad (1)$$

$$s.t.\ \sum_{j=1}^{n} x^{t}_{0j} = m^{t}, \quad t = 1,\ldots,g, \quad (2)$$

$$\sum_{k \in \{S^{t}, n+1\}} y^{t}_{lk} = 1, \quad l = 1,\ldots,n_G,\ t \in S_l, \quad (3)$$

$$\sum_{l \in \{0, S^{t}\}} y^{t}_{lk} = 1, \quad k = 1,\ldots,n_G,\ t \in S_l, \quad (4)$$

$$\Pr\left[C^{t}_{j} - C^{t}_{i} + M^{t}(1 - x^{t}_{ij}) \geq P^{t}_{ij}\right] \geq 1 - \varepsilon, \quad i = 0,\ldots,n,\ j = 1,\ldots,n,\ t \in e_t, \quad (5)$$

$$\Pr\left[C^{t}_{j} - C^{t-1}_{j} + M^{t}_{j}(1 - x^{t}_{ij}) \geq P^{t}_{ij}\right] \geq 1 - \varepsilon, \quad i = 0,\ldots,n,\ j = 1,\ldots,n,\ t \in e_t - \{1\}, \quad (6)$$

$$\Pr\left[CG^{t}_{k} - CG^{t}_{l} + M^{t}(1 - y^{t}_{lk}) - S^{t}_{lk} \geq P^{t}_{k}\right] \geq 1 - \varepsilon, \quad l = 0,\ldots,n_G,\ k = 1,\ldots,n_G,\ t \in S_l, \quad (7)$$

$$\Pr\left[CG^{t}_{k} - CG^{t-1}_{k} + M^{t}_{k}(1 - y^{t}_{lk}) - S^{t}_{lk} \geq P^{t}_{k}\right] \geq 1 - \varepsilon, \quad l = 0,\ldots,n_G,\ k = 1,\ldots,n_G,\ t \in S_t - \{1\}, \quad (8)$$

$$\begin{cases} \Pr\left[x^{t}_{ij} \leq P^{t}_{ij}\right] \geq 1 - \varepsilon, \\ \Pr\left[x^{t}_{ji} \leq P^{t}_{ij}\right] \geq 1 - \varepsilon, \end{cases} \quad i,j \in \{0,\ldots,n, n+1\}, \quad (9)$$

$$C^{t}_{j} \geq C^{t}_{0}, \quad j = 1,\ldots,n, \quad t = 1,\ldots,g, \quad (10)$$

$$f_1 \geq C^{g_j}_{j}, \quad j = 1,\ldots,n, \quad (11)$$

$$T_j = max\{0, C^{g_j}_{j} - d_j\}, \quad j = 1,\ldots,n, \quad (12)$$

$$f_2 = \sum_{j=1}^{n} W_j T_j, \quad (13)$$

$$\begin{cases} x^{t}_{ij} \in \{0,1\}, & i,j \in \{0,\ldots,n,n+1\},\ t = 1,\ldots,g \\ x^{t}_{ij} = 0, & i = j,\ t = 1,\ldots,g \end{cases} \quad (14)$$

$$\begin{cases} y^{t}_{lk} \in \{0,1\}, & l,k \in \{0,\ldots,n_G,n_{G+1}\},\ t = 1,\ldots,g \\ y^{t}_{lk} = 0, & l = k,\ t = 1,\ldots,g \end{cases} \quad (15)$$

$$C^{t}_{j} \geq 0, \quad j = 1,\ldots,n, \quad t = 1,\ldots,g, \quad (16)$$

The value of M^{t}_{k} equals to $max_i(S^{t}_{lk}) + (P^{t}_{k})$. Constraint set (9) shows that the processing time with satisfactory level 1-ε is considered only for those jobs entered in a stage. In other words, the processing times of jobs not entered in a stage are considered to be zero. Constraint set (10) guarantees that the completion time of job not belonged to station t yet, equals to setup time in station t-1. Constraint set (11) shows the relationship between $C^{g_j}_{j}$ and f_1. The value of tardiness can be calculated by constraint set (12). Constraint set (13) shows the relationship between total tardiness and f_2. Constraint sets (14) to (16) represent the state of the decision variables.

3.5. Transforming the stochastic model to a deterministic model

In the above model, the processing time is a stochastic variable with normal distribution and has generated chance constraint sets (5) to (9). Chance constraints can be converted to deterministic constraints. Chance

constraint sets (5) to (9) change as deterministic constraint sets (17) to (21).

$$C_j^t - C_i^t + M^t(1-x_{ij}^t) \geq \mu_{p_{ij}^t} + \sigma_{p_{ij}^t} F^{-1}_{p_{ij}^t}(1-\varepsilon), \quad i=0,\ldots,n, \quad j=1,\ldots,n, \quad t \in e_i, \tag{17}$$

$$C_j^t - C_j^{t-1} + M_j^t(1-x_{ij}^t) \geq \mu_{p_{ij}^t} + \sigma_{p_{ij}^t} F^{-1}_{p_{ij}^t}(1-\varepsilon), \quad i=0,\ldots,n, \quad j=1,\ldots,n, \quad t \in e_i-\{1\} \tag{18}$$

$$CG_k^t - CG_l^t + M^t(1-y_{lk}^t) - S_{lk}^t \geq \mu_{p_k^t} + \sigma_{p_k^t} F^{-1}_{p_k^t}(1-\varepsilon), \quad l=0,\ldots,n_G, \quad k=1,\ldots,n_G, \tag{19}$$

$$CG_k^t - CG_k^{t-1} + M_k^t(1-y_{lk}^t) - S_{lk}^t \geq \mu_{p_k^t} + \sigma_{p_k^t} F^{-1}_{p_k^t}(1-\varepsilon), \quad l=0,\ldots,n_G, \quad k=1,\ldots,n_G, \tag{20}$$

$$\begin{cases} x_{ij}^t \leq \mu_{p_{ij}^t} + \sigma_{p_{ij}^t} F^{-1}_{p_{ij}^t}(1-\varepsilon) \\ x_{ji}^t \leq \mu_{p_{ii}^t} + \sigma_{p_{ij}^t} F^{-1}_{p_{ii}^t}(1-\varepsilon) \end{cases} \quad i,j \in \{0,\ldots,n,n+1\}, \tag{21}$$

3.6. Linearization Max{0,x}

The only nonlinear constraint set in this problem is the constraint set (12). As Ghezavati and Saidi-Mehrabad (2010) expressed, in order to linearization such terms as $Max\{0,x\}$, one additional variable and two auxiliary constraints must be used to replace with this term. Following example shows the linearization operation of these constraint sets:

$Minimize\ Z$

$s.t.\ Z = Max\{0,X\}$ linearization :

$Minimize\ Z$

$s.t.\ Z \geq 0$

$Z \geq X$

4. Proposed algorithms for the model

Many optimization problems are *NP-hard*. It is impossible to solve these problems with exact method in a reasonable time. So, it is recommended to apply metaheuristic methods when there are time limits for solving the optimization problem and it is hard to use exact solution methods or the problem is very complex. In this section two metaheuristic algorithms SPGA-II and NSGA-II are introduced to solve our problem.

4.1.SPGA-II: Cheng et al. (2005) proposed the Sub-Population Genetic Algorithm (SPGA) for the first time. The main idea of SPGA is based on dividing the main population to some sub-populations. Then, different weights are allocated to these sub-populations. All solutions existing in a sub-population use a similar weight allocated to that sub-population. The main reason for allocating different weights to different sub-populations is to extend search solution space to various directions as well as better convergence. SPGA-II was expressed by Cheng and Chan (2009) which is a development of SPGA. Some characteristics distinguishing the SPGA-II from SPGA are: SPGA-II uses a global pareto archive that must be updated immediately after finding a new set of pareto. In addition, SPGA-II exerts a two phase approach that uses the best solutions of the first phase to generate initial solutions of the second phase.

SPGA-II developed in this paper is acts based on the following steps:

Step 1. Solution representation: In this paper the matrix method is used to code each chromosome. Each row of the matrix representative of a group of that chromosome. To allocate groups to machines in the first step, the method presented by Ebrahimi et al. (2014) is applied. For each group, a random number is generated in the interval [1,m^1+1], where$m^1$is the number of machinesin the first stage. The integer part of the generated number is the number of the machines that group must be processed on. The fractional part of this number is desired to determine the sequence of jobs inside each group for the specific machine. In order to determine the sequence of jobs in each group, random numbers are generated in interval [0,1]. The job with a smaller number is processed more quickly. In the next stage, allocation of groups and jobs to the machines is done according to their completion times in the previous stage.

The sequence of jobs and groups on the first machine will be as follows:

(Machine 1:$G_3(J_4,J_3$, $J_1,J_2)$, $G_2(J_2,J_1)$ *)(Machine 2:* $G_1(J_3, J_1,J_2)$*), (Machine 3:* $G_4(J_3, J_2,J_1)$ *),*

Step 2. Start: A population including N chromosomes is generated.

Step 3.Dividing the population to sub-populations: All the solutions of this population must be divided to several sub-populations.

Step 4. Weight allocation: A weight is allocated to each sub-population. These weights differ for different sub-populations. w_t is the defined weight for sub-population t. The value of w_t is obtained from the following equation:

$w_t = \left| Sin\left(\frac{2\pi t}{\alpha}\right)\right|$, in this equation, $\alpha = 40$ and t is the number of the sub-population.

Step 5. Start of the first phase: The value of the objective function is calculated for all solutions. In SPGA-II, this phase is exerted on sub-populations that have weights near to the boarders of the weight vector. As an example, if 20 sub-populations are generated, sub-populations 1 to 5 and 16 to 20 are performed in the first phase and the others are performed in the second phase.

Stage 6. Evaluation: The value of fitness is calculated using determined weights and normalized objective functions. The objective functions may have different units, requiring the use of their normalized values expressed as:

$$f_1(x) = \frac{C_{max}(x) - best(C_{max})}{worst(C_{max}) - best(C_{max})} \quad , \quad f_2(x) = \frac{TT(x) - best(TT)}{worst(TT) - best(TT)}$$

For solution x, $C_{max}(x)$ and $TT(x)$ are representative of numerical value of first and second objective functions. Worst (C_{max}) and best (C_{max}) are representative of the worst and best values of $C_{max}(x)$. Best (TT) and worst(TT) are

representative of the best and worst values of objective function TT. $f_i(x)$ is the normalized value of the i objective function to x. In this paper, the value of fitness function for the solution x is obtained from the following expression:

$$f_{it}(x) = W_t.f_1(x) + (1- W_t).f_2(x)$$

Step 7.Update the global pareto archive: After calculating the values of fitness, the non-dominated solutions set of each sub-population is determined and updating of the global pareto archive is done.

Step 8. New solutions generation: Binary tournament is applied as the selection strategy. In order to replace current sub-populations, the reproduction operator is exerted. Also, crossover and mutation operations are used as follows.

Crossover: The crossover creates two new children by combining both parents chromosomes' gens. For this study we perform uniform crossover. Suppose that the probability of crossover would be r_c. Now, the chromosome with a better fitness value is considered to be the parent 1 and the other, parent 2. Then a new chromosome with random number between (0,1) should be generated and called mask. The mask size should be equal to the maximum length of groups. If the random number is more than 0.7 for a specific gene, the corresponding value is copied in it from the parent 1 otherwise, parent 2 is used.

Mutation: Firstly, two groups of a chromosome are selected randomly and values related to these two groups are replaced with each other. In the next step, two jobs of each group are selected and their related values are replaced with each other.

Step 9. Second phase: After completion of the first phase, initial solutions for sub-populations of the second phase must be generated. For this purpose, the tournament selection strategy must be applied for the current solutions in order to generate the sub-populations of the second phase.

After the sub-populations of the second phase are generated, the above process is continued for these sub-populations till stopping criterion is met.

4.2. NSGA-II

Another algorithm developed for the problem in this paper is the NSGA-II. The form of solution representation and mutation and crossover operations of this algorithm are similar to SPGA-II algorithm explained in section 4.1. In this section, a brief explanation of steps and general structure of NSGA-II is presented. First of all, an initial population P_0 is generated randomly. All offspring chromosomes Q_t and population P_t are created by selection, mutation and crossover operators and then they are evaluated. Afterward, all individuals of P_t and Q_t will be ranked and put in varying fronts. First pareto front not dominated by other fronts is generated. This front includes non-dominated solutions. To generate next fronts, only remaining solutions are used. This process continues till all solutions are ranked and assigned to different fronts. Then, the best solutionin the best front with most crowding distance is selected to generate population P_{t+1}.

4.3. Evaluation of non-dominated solution

To compare the pareto solution sets of various algorithms, quantitative metrics are required. There are various methods to measure performance of the multi-objective algorithms. In this paper, the following methods have been applied:

1. Number of pareto set members(N)
2. Spacing
3. Mean Ideal Distance (MID)
4. The rate of achievement to two objectives simultaneously (RAS)

In designing these metrics, the following three principles contribute:

- The number of non-dominated solutions obtained from algorithm
- Crowding distance of obtained solutions
- Coverage, distribution and scattering of non-dominated solutions

Number of pareto set members (N): The more the number of solutions, the more the possibility of spreadness and scattering of solutions in the set.

Spacing: Although non-dominated solutions may be appropriate in terms of scattering, it is possible that non of them be different structurally or many of them be similar. In this condition, the spacing metric is calculated as follows:

$Spacing = \sqrt{\frac{\sum_{i=1}^{n}(d_i - \bar{d})^2}{n-1}}$, n is the number of pareto set members. $\bar{d}$ is the mean of d_i calculated from the equation: $d_i = min_j \{| f_1^i(x) - f_1^j(x) | + | f_2^i(x) - f_2^j(x) |\}, i,j=1,...,n$

Mean Ideal Distance (MID): Point (0,0) as the ideal point and the distance between each pareto solution and this ideal point is being calculated. The average of these distances is called MID and calculated as follows: $MID = \frac{\sum_{i=1}^{n} C_i}{n}, C_i = \sqrt{f_{1i}^2 + f_{2i}^2}$

Where n is the number of non-dominated sets, C_i is the distance between the ith non-dominated solution and ideal point. f_{1i}, f_{2i} are the values of ith non-dominated solution for the first and second objective functions respectively.

The rate of achievement to two objectives simultaneously (RAS): A suitable solution is a solution set that has reasonable performance with respect to all objectives. RAS is defined to measure the achievement to this balance between various objectives and their scattering. The equation of RAS is presented in the below:

$RAS = \frac{\sum_{i=1}^{n}(\frac{f_{1i} - F_i}{F_i}) + (\frac{f_{2i} - F_i}{F_i})}{n}$, where $F_i = min\{f_{1i}, f_{2i}\}$

5. Implementation and comparing the algorithms

This section compares our proposed algorithms with each other. These algorithms have been coded with MATLAB R2008band run with an Intel Pentium IV dual core 2 GHz PC at 2 GB RAM under a Microsoft Windows XP environment.

5.1. ***Data generation:*** In this paper, different sizes have been used for the problem. Tables 1-3have been created to demonstrate the generated example problems used in evaluating the performance of the algorithms.

Table 1 Characteristics of test problem.

Factors	Levels	
	Small	Large
Number of stages	U(5,20)	U(5,20)
Number of groups	U(5,10)	U(20,50)
Number of jobs	(20,50)	U(100,500)
Setup time	U(5,25)	U(5,30)
	U(5,50)	U(5,100)
	U(5,75)	U(5,150)
	U(5,100)	U(5,200)
	U(5,150)	U(5,250)
Maximum number of machines in each stage	4	5

Table 2: Jobs processing time.

Levels	$\mu_{p_{ij}^t}$	$\sigma_{p_{ij}^t}$	$1-\varepsilon$
Small	70	20	0.9
	100	25	0.9
	200	30	0.9
	300	28	0.9
	350	30	0.9
Large	150	50	0.9
	300	65	0.9
	350	70	0.9
	400	70	0.9
	500	85	0.9

Table 3: Groups processing time.

Levels	$\mu_{p_k^t}$	$\sigma_{p_k^t}$	$1-\varepsilon$
Small	280	80	0.9
	400	100	0.9
	800	130	0.9
	1200	135	0.9
	1400	125	0.9
Large	750	250	0.9
	1500	275	0.9
	1750	280	0.9
	2000	295	0.9
	2500	350	0.9

In this paper it is supposed that the number of jobs is equal in all groups and it is 4 jobs in each group.

5.2. Parameter setting: The value of different parameters is one of the effective factors on the performance of

each algorithm. Different combination of parameter values may result in obtaining a non-dominated solution set with different ability. A number of experiments have been done using different sets of parameters in order to set the best values for parameters of the two developed algorithms. For this purpose, we considered three problems with different sizes and determined the best level of each parameter using practical study and an empirical testing approach. These parameters are shown in Table 4.

Table 4: Parameter setting results.

Algorithms	*Problem size*	*Population size*	*Number of iterations*	*Crossover rate*	*Mutation rate*	*Number of sub-populations*
SPGAII	*Small*	100	400	0.8	0.1	15
	Large	200	400	0.85	0.05	20
NSGAII	*Small*	100	400	0.82	0.08	
	Large	200	400	0.9	0.1	

5.4. Experimental results: The most common condition of stopping the algorithm is the specific number of repetitions determined according to experimental results obtained from various implementations and problem size. The equal number of repetitions of two compared algorithms does not lead in equality of implementation times of both algorithms and the results obtained from these implementations cannot prepare fair conditions to compare the algorithms. Because of this, in recent years, some researchers have considered the stop condition of the compared algorithm to be the equal times. This method was applied in this paper. The NSGA-II algorithm was performed while a specific number of repetitions were met where this number has a direct relationship with the number of jobs. Experimentally, the result was that providing more repetitions, no considerable improvement was obtained. Then, the time of NSGA-II algorithm was considered as the stop condition of SPGA-II. SPGA-II was implemented at least to the NSGA-II time. The implementation times of the algorithms are approximately equal. For each set of problems four performance metrics are investigated and shows the average of results of proposed algorithms. The pareto set of both algorithms for a specific problem. In this specific problem, the pareto set of NSGA-II algorithm is closer to the coordinates origin. However, the number of members of non-dominated solution set of SPGA-II is larger. The important point in these diagrams is that the increase in the problem size although constructs more complex problems, does not decrease the algorithms performance.

6. Conclusions and recommendation for future studies

In this paper, the hybrid flow shop (HFS) scheduling problem was studied. The HFS problem is amongst the common problems of production and industry systems. In order to increase the performance of these problems, it was assumed that family parts are placed in one group and only for each group the setup time was considered. Also, to make the problem closer to real conditions of production environment, the stochastic processing time and sequence-dependent family setup time were considered. In other hand, since most of the

optimization problems in real environment have more than one objective, the makespan and total tardiness were considered as objective functions, simultaneously. According to assumptions and objective functions, a stochastic mathematical model of the problem was proposed. Then, the model of the problem was changed to a deterministic model using the methods found in literature. Since the proposed model was *NP-hard* and it is impossible to find a solution that satisfies all objective functions, the NSGA-II and SPGA-II algorithms, as two metaheuristic algorithms, were developed to obtain pareto solutions set. These algorithms were implemented for various sizes and numerous implementations. To study the performance of proposed algorithms, four evaluation metrics were applied. Comparing the results obtained from these algorithms with those of methods found in literature, confirms the proper performance of algorithms developed in this paper. We use four criteria to compare the results between NSGA-II and SPGA-II in this paper; *Pareto number*, *Spacing*, *MID* and *RAS*.

Pareto Number: The results of NSGA-II are better than SPGA-II.
Spacing: The results from two algorithms are close together in small size, but in large size NSGA-II is better than SPGA-II.
MID: The results from two algorithms are almost identical.
RAS: The results of NSGA-II are slightly better than SPGA-II.
In general the results of NSGA-II are better than SPGA-II.
The future work is to change the objectives that we used in this paper, assumption of uncertainty for other parameters and extension of proposed algorithms to other environments of scheduling.

References

Ahmadizar, F., Ghazanfari, M., Fatemi Ghomi, S.M.T., (2009),"Application of chance-constrained programming for stochastic group shop scheduling problem,"International Journal of Advanced Manufacturing Technology, Vol. 42, pp. 321-344.

Almeder, C., Hartl, R.F., (2013)"A metaheuristic optimization approach for a real-world stochastic flexible flow shop problem with limited buffer,"International Journal of Production Economics, Vol. 145(1), pp. 88-95.

Amin-Tahmasbi, H., Tavakkoli-Moghaddam, R., (2010)"Solving a bi-objective flow shop scheduling problem by a Multi-objective Immune System and comparing with SPEAII+ and SPGA," Advances in Engineering Software, Vol. 42(10), pp. 772-779.

Aydilek, H., Allahverdi, A., (2010)"Two machine flow shop scheduling problem with bounded processing times to minimize total completion time,"Computers and Mathematics with Applications, Vol. 59, pp. 684-693.

Baker, K.R., Altheimer, D., (2012) "Heuristic solution methods for the stochastic flow shop problem,"European Journal of Operation Research, Vol. 216, pp. 172–177.

Boloori Arabani, A., Zandieh, M., Fatemi Ghomi, S.M.T., (2011) "Multi-objective genetic-based algorithms for a cross-docking scheduling problem,"Applied Soft Computing, Vol. 11(8), pp. 4954-4970.

Garey, M.R., Johnson, D.S., (1979) "A guide to the theory of NP-completeness,"

Computers and intractability San Francisco: Freeman, Vol. 11, pp.20-26.

Ghezavati, V., Saidi-Mehrabad,M., (2010) "Designing integrated cellular manufacturing systems with scheduling considering stochastic processing time,"International Journal of Advanced Manufacturing Technology, Vol. 48, pp. 701–717.

Ghezavati, V.R., Saidi-Mehrabad, M., (2011) "An efficient hybrid self-learning method for stochastic cellular manufacturing problem: A queuing-based analysis, "Expert Systems with Application, Vol. 38, pp. 1326–1335.

Golenko-Ginzburg, D., Aharon Gonik, Zohar Laslo, (2003) "Resource constrained scheduling simulation model for alternative stochastic network projects,"Mathematics and Computers Simulation, Vol. 63, pp. 105–117.

Gourgand, M., Grangeon, N. Norre, S., (2003) "A contribution to the stochastic flow shop scheduling problem,"European Journal of Operation Research, Vol. 151, pp. 415–433.

Gupta, J.N.D., (1988) "Two-stage, hybrid flow shop scheduling problem,"Journal of the Operation Research Society, Vol. 39, 1988 pp. 359–364.

Rahmani, D., Heydari, M., (2014) "Robust and stable flow shop scheduling with unexpected arrivals of new jobs and uncertain processing times,"Original Research Article, Vol. 33, pp. 84-92.

Rajendran, C., Holthaus, O., (20015) "A comparative study of dispatching rules in dynamic flow shops and job shops,"European Journal of Operation Research, Vol. 116, pp. 361-370.

Salvador, M.S., (1973) "A solution to a special class of flow shop scheduling problems," In: Elmaghraby SE editor. Symposium on the theory of scheduling and its applications. Berlin: Springer, Vol. 24, pp. 83–91.

Seo, K.D., Cerry, M. Wooseung, J., (2005) "Single machine stochastic scheduling to minimize the expected number of tardy jobs using mathematical programming models,"Computers and Industrial Engineering, Vol. 48, pp. 153–161.

Soroush, H.M., (2007) "Minimizing the weighted number of early and tardy jobs in a stochastic single machine scheduling problem,"European Journal of Operation Research, Vol. 181, pp. 266–287.

Tavakkoli-Moghaddam, R., Rahimi-Vahed, A., Mirzaei, 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, Vol. 177, pp. 5072–5090.

Ulungu, E.l., Techem, J.P.H., Fortemps, D., Tuytten, (1999) "MOSA method: A tool for solving MOCO problems," Journal Multi-Criteria DecisionAnalysis, Vol. 21, pp. 221-236.

Wang, K., Choi, S.H., (2014) "A holonic approach to flexible flow shop scheduling under stochastic processing times,"Computers and Operation Research, Vol. 43, pp. 157-168.

Wei-Chang Yeh, Peng-Jen Lai, Wen-Chiung Lee, Mei-Chi Chuang, (2014) "Parallel-machine scheduling to minimize makespan with fuzzy processing times and learning effects,"Original Research Article, Vol. 269, pp.142-158.