Journal of Industrial and Systems Engineering
Vol. 11, Special issue: 14th International Industrial Engineering Conference
Summer (July) 2018, pp. 150-162

JISE

(IIEC 2018)
TEHRAN, IRAN

Minimizing the energy consumption and the total weighted tardiness for the flexible flowshop using NSGA-II and NRGA

Mohammad Mahdi Nasiri[1]*, Mojtaba Abdollahi[1], Ali Rahbari[2], Navid Salmanzadeh Meydani[3], Sadegh Salesi[1]

[1] School of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran
[2] Department of Industrial Engineering, Alborz Campus, University of Tehran, Tehran, Iran
[3] Department of Industrial Engineering, Amirkabir University of Technology, Tehran, Iran
mmnasiri@ut.ac.ir, mojtaba.abdollahi@ut.ac.ir, rahbari.a@ut.ac.ir, navid.salmanzadeh@aut.ac.ir, sadegh.salesi@ut.ac.ir

Abstract

This paper presents a bi-objective MIP model for the flexible flow shop scheduling problem (FFSP) in which the total weighted tardiness and the energy consumption are minimized simultaneously. In addition to considering unrelated machines at each stage, the set-up times are supposed to be sequence- and machine-dependent, and it is assumed that jobs have different release times. Two Taguchi-based-tuned algorithms: (i) non-dominated sorting genetic algorithm II (NSGA-II), and (ii) non-dominated ranked genetic algorithm (NRGA) are applied to solve the model. Six numerical examples with different sizes (small, medium, and large) are used to demonstrate the applicability and to exhibit the efficacy of the algorithms. The results show that the NRGA outperforms significantly the NSGA-II in the performance metrics for all six numerical examples.

Keywords: Flexible flow shop scheduling, energy consumption, weighted tardiness, genetic algorithm, strength Pareto evolutionary algorithm.

1-Introduction

In today's competitive supply chains, an effective sequencing and scheduling is vital in order to survive in the market (Tavakkoli-Moghaddam et al. 2010). The sequencing usually associates with the order of jobs that should be processed on a same machine while, the scheduling usually refers to allocate the jobs to a complex set of machines with respect to time. A suitable scheduling of resources in a company leads to enhance the resource utilization, decrease the time required to complete jobs, and finally increase the profitability (Kerzner and Kerzner , 2017).The resources might be machines in workshops, routes in airports, workers at a construction site, processing units in a computational environment, etc (Pinedo et al. 2015). Although the traditional scheduling approaches considered performance indicators such as life-cycle of product, cost, and quality as optimization objectives in manufacturing systems; recentlya growing body of literature has investigatedon minimizing the energy consumption (Pechmann and Schöler, 2011).

*Corresponding author
ISSN: 1735-8272,

The supply of energy plays a vital role in the continuation of human life, and the minimizing of energy consumption is crucial for the sustainable developemtn of modern societies (Asif and Muneer, 2007). Some of the studies have shown that the consumption of energy in companies can be reduced by alternative operational strategies such as more stringent management and machines scheduling (Dahmus and Gutowski , 2004), (Gutowski et al. 2005) and (Drake et al. 2006).

This paper investigates the reduction of energy consumption in a flexible flow shop scheduling problem. A flexible flow shop scheduling problem (FFSP) is an extension of the classical flow-shop scheduling (LIN and Zhang, 2014).The FFSP is a multi-stage production process that consists of two or more production stages in series. There is at least one machine tool in each production stage, and at least one stage has more than one machine tool. All jobs have to pass every production stage in the same order (each job requires to be processed by only one machine in each stage and every machine can process it) (Dai et al, 2013).The energy consumption (EC) is considered as a new objective and it is optimized concurrently with the other objective of the company.Among the common objectives in the field of prodution scheduling, tardiness is a significant one that has attracted more attentions in recent years (Liu et al. 2014), (Liang et al. 2015), (Liu et al. 2015), (Ruiz et al. 2010), (Lee and Kim, 2004), (Lee et al. 2004), (Naderi et al. 2009) and (Jun and Park, 2015). This is because of the high importance of customers' satisfaction in today's competitive envirnment. Moreover, if a job was not prepared in its due date, not only the service level would be reduced, but also additional costs would be imposed. These costs include specified penalties in the contracts, loss of credibility, increasing the possibility of losing the customers, reducing the attraction of potential customers, and etc. (Choi et al. 2005), (Gupta et al. 2002) and (Davoudpour and Ashrafi, 2009). Since these costs can be different for various jobs (customers), a weight might be consideredfor each job (customer) which represents the relative importance (priority) of it.Therefore, considering a total weighted tardiness (TWT)as an objective function will enable the problem to be more compatible with the conditions of the real world.Therefor, this paper presents a bi-objective MIP model in which the TWT and the EC should be minimized simultanously. Moreover, in order to enhance the applicability of the model two operational conditions are applied: the first, each jobis associated with a release time before which it cannot be processed, and the second, set-up times are considered to be sequence- and machine-dependent.

Table 1. Features of this study in comparison with some other related studies

Study	Production Environment				Machine Type		Evaluation Criteria							Solution Method
	JS	*FJS*	*FS*	*FFS*	*P*	*R*	*EC*	C_{max}	$\sum F$	T_{max}	$\sum T$	$\sum wT$		
(Naderi et al. 2009)				✓	✓			✓		✓				Simulated Annealing
(Rashidi et al. 2010)				✓		✓		✓		✓				Heuristics, MPGA
(Fang et al. 2011)			✓		✓		✓	✓						Ant Colony
(Luo et al. 2013)				✓	✓		✓	✓						Ant Colony
(Song et al. 2014)		✓			✓		✓	✓						NSGA-II
(Liu et al. 2014)	✓				-		✓					✓		NSGA-II
(Yan et al. 2014)				✓	✓		✓	✓						NSGA-II
(Ebrahimi et al. 2014)				✓	✓			✓			✓			MOGA, NSGA-II
(LIN et al. 2014)				✓	✓			✓	✓		✓			NSGA-II
(Li et al. 2015)				✓		✓		✓				✓		HSGA
(Tang et al. 2016)				✓	✓		✓			✓				PSO
(Yan et al. 2016)				✓	✓		✓			✓				CPO, GA
This Study				✓		✓	✓					✓		NRGA, NSGA-II

Since FFSP is known as a NP-hard problem (Dai et al. 2013), two solution algortihms, including (i) non-dominated sorting genetic algorithm II (NSGA-II), and (ii) non-dominated ranked genetic algorithm (NRGA) are adopted to make a suitable trade-off between the TWT and the EC.

Up to the best of our knowledge, there is no comprehensive study which has simultaneously consideredthe EC and the TWTas objective functions for the flexible flow shop scheduling problem and none of them are applied and compared two mentioned solution algorithms(unique combination in Table 1).

2-Problem description and a mathematical model

In a flexible flow shop,there are $n$jobs that should be processed in k production stages. There is at least one machine in each stage, and at least one stage has more than one machine.The problem is to schedule the production so as to minimize the energy consumed by machines and the tardiness penalty cost of jobssimultanously. These two objectives conflict in such a way that minimizing enegry consumption may lead to prefer machines with low velocity at each stage, and minimizing tardiness may lead to prefer ones with high velocity in order to shorten the life-cycle of jobs.

It is assumed that the parallel machines in each stage are unrelated which means that there are at least m^t different parallel machines in stage t and machine i processes job j with v_{ij} velocity. Jobs are independent, and each job has only one operation at each stage, moreover a job must be processed on a machine without break (preemption is not allowed), and each job is associated witha due date and a release time (release times is a period of time when a job becomes available for processing). Set-up times are sequence- and machine-dependent, and a machine consumes energy when it turns on or off, and when it is idle. Moreover, for each job, the amount of consumption of energy by a machine is different during the set-up time and processing time. Other assumptions of the problem are as follows:

- All machines are available at time zero.
- A machine can only process one job at a time, and machines of a stage cannot process the jobs of the other stages.
- Release times, due dates, processing times, set-up times, tardiness penalty costs, and the amount of energy consumptions are deterministic.

2-1-Notations

The indices, parameters and variables used for the formulation of the model are as follows:

Indices:

t	Index of stage	$t = 1,2,\dots,k$
i	Index of machine	$i = 1,2,\dots,m^t$
j,l	Index of job	$j,l = 0,1,2,\dots,n,n+1$

Parameters:

m^t	Number of parallel machines at stage t
d_j	Due date of job j
r_j	Release time of job j
w_j	Weight of job j
M	A very large number
s_{ijl}^t	Set up time between job l and job j on machine i at stage t
ch_{ij}^t	Set up time of job j if the job processes as the first job on machine i at stage t
P_{ij}^t	Process time of job j on machine i at stage t
ER_i^t	Amount of energy used to turn on and off machine i at stage t
EP_{ij}^t	Amount of energy used to process of job j on machine i at stage t in a unit of time
ES_{ij}^t	Amount of energy used to set up of job j on machine i at stage t in a unit of time
EI_i^t	Amount of energy used during idle time of machine i at stage t in a unit of time

Decision variables:

Ini_i^t	Initial start time of machine i at stage t

Id_i^t	Idle time of machine i at stage t
S_{ij}^t	Start time for processing of job j on machine i at stage t
C_j^t	Completion time of job j at stage t
C_j	Completion time of job j at the final stage
$C_{\max i}^t$	Completion time of machine i at stage t
T_j	Tardiness if job j
X_{ijl}^t	A binary variable; 1, if job l immediately allocated and scheduled on machine i at stage t
X_{i0l}^t	A binary variable; 1, if job l allocated and scheduled on machine i at stage t as the first job
$X_{ij(n+1)}^t$	A binary variable; 1, if job l allocated and scheduled on machine i at stage t as the last job

2-2-Mathematical model

The objective functions and the constraints of the model are as follows:

- *Objective functions*

$$min\, Z_1 = \sum_{t=1}^{k}\sum_{i=1}^{m^t}\sum_{j=0}^{n}\sum_{l=1}^{n} EP_{il}^t P_{il}^t X_{ijl}^t + \sum_{t=1}^{k}\sum_{i=1}^{m^t}\sum_{l=1}^{n} ES_{il}^t \left(\sum_{j=1}^{n} s_{ijl}^t X_{ijl}^t + ch_{il}^t X_{i0l}^t\right) + \sum_{t=1}^{k}\sum_{i=1}^{m^t} EI_i^t Id_i^t + \sum_{t=1}^{k}\sum_{i=1}^{m^t}\sum_{l=1}^{n} ER_i^t X_{i0l}^t \quad (1)$$

$$minZ_2 = \sum_{j=1}^{n} w_j \times T_j \quad (2)$$

Equation (1) minimizes the total energy consumption, as the first objective function, during process times, set-up times, idle times, and when the machines turn on and off. Equation (2) minimizes the total weighted tardiness for jobs as the second objective function.

- *Constraints*

$$\sum_{i=1}^{m^t}\sum_{j=0}^{n} X_{ijl}^t = 1 \qquad \forall t, l = 1, \dots, n \quad (3)$$

$$\sum_{i=1}^{m^t}\sum_{l=1}^{n+1} X_{ijl}^t = 1 \qquad \forall t, j = 1, \dots, n \quad (4)$$

$$\sum_{l=1}^{n} X_{i0l}^t \le 1 \qquad \forall i, t \quad (5)$$

$$\sum_{j=1}^{n} X_{ij(n+1)}^t \le 1 \qquad \forall t, i \quad (6)$$

$$X_{ijj}^t = 0 \qquad \forall t, i, j = 1, \dots, n \quad (7)$$

$$\sum_{j=0}^{n} X_{ijl}^t = \sum_{j=1}^{n+1} X_{ilj}^t \qquad \forall t, i, l = 1, \dots, n \quad (8)$$

$$X_{ijl}^t \in \{0,1\} \qquad \forall t, i, j = 0, l = n+1 \quad (9)$$

Constraints (3) – (9) guarantee the feasibility of the considered scheduling on machines at each stage. Constraints (3) and (4) guarantee that only one job must be allocated in each sequence at each stage. Constraints (5) and (6) ensure that only one job must be allocated in initial and final positions of each sequence of jobs on each used machine at each stage. Constraint (7) states that a job cannot process at a stage more than once. Constraint (8) causes a consistent and compatible sequencing at each stage. Constraint (9) determines X_{ijl}^t as a binary decision variable.

$$S_l^t \geq r_l \qquad \forall l, t = 1 \qquad (10)$$

$$S_l^t \geq C_l^{t-1} \qquad \forall l, t \neq 1 \qquad (11)$$

$$S_l^t \geq \sum_{i=1}^{m^t} ch_{il}^t X_{i0l}^t \qquad \forall l, t \qquad (12)$$

$$S_l^t - C_j^t \geq \sum_{i=1}^{m^t} s_{ijl}^t X_{ijl}^t + \left(\sum_{i=1}^{m^t} X_{ijl}^t - 1 \right) M \qquad \forall t, j, lj \neq l \qquad (13)$$

$$S_j^t \geq 0 \qquad \forall t, j \qquad (14)$$

Constraints (10) – (14) find the initial start time for each job. Constraints (10) and (11) present the initial start time of each job respectively at first stage and next stage. At the first stage, the start time of each job must be at least equal to the release time of it. At the next stage, the start time determines by completion time in previous stages. Constraint (12) ensures that the start time of each job on each machine must be at least equal to its release time. Constraint (13) ensures that if job j and l are assigned to the same machine, the start time of job j (that processes after job l) must be at least set up time of job l plus completion time of job j. Constraint (14) guarantees the non-negativity of the start time of each job at each stage.

$$C_l^t = S_l^t + \sum_{i=1}^{m^t} \sum_{j=0}^{n} P_{il}^t X_{ijl}^t \qquad \forall l, t \qquad (15)$$

$$C_{\max i}^t \geq C_j^t + \left(X_{ij(n+1)}^t - 1 \right) M \qquad \forall j, i, t \qquad (16)$$

$$C_{\max i}^t \leq C_j^t - \left(X_{ij(n+1)}^t - 1 \right) M \qquad \forall j, i, t \qquad (17)$$

$$int_i^t \geq S_l^t - ch_{il}^t + \left(X_{i0l}^t - 1 \right) M \qquad \forall l, i, t \qquad (18)$$

$$int_i^t \leq S_l^t - ch_{il}^t - \left(X_{i0l}^t - 1 \right) M \qquad \forall l, i, t \qquad (19)$$

$$Id_i^t = C_{\max i}^t - int_i^t - \sum_{l=1}^{n} \left(ch_{il}^t X_{i0l}^t + \sum_{j=1}^{n} s_{ijl}^t X_{ijl}^t + \sum_{j=0}^{n} P_{il}^t X_{ijl}^t \right) \qquad \forall i, t \qquad (20)$$

$$T_j \geq C_j^t - d_j \qquad \forall j, t = k \qquad (21)$$

$$T_j \geq 0 \qquad \forall j \qquad (22)$$

Constraint (15) determines completion time of each job as total start time and process time. Constraints (16) and (17) calculate completion time of each machine at each stage that equals to the completion time of the last job processed on that machine. Here, the equality constraint converted to two inequality constraints by using M. This constrains would be active while X_{ijl}^t becomes 1, otherwise, they become inactive constraints. Constraint (18) and (19) represent initial start time of each machine at each stage. Constraint (20) determines the idle time of each machine at each stage. Constraints (21) and (22) include set of constraints that determine the amount of tardiness for each job.

3-Solution algorithms

Regarding the complexity of the model and the NP-hardness nature of FFSP (Dai et al. 2013), two solution algortihms, including (i) non-dominated sorting genetic algorithm II (NSGA-II), and (ii) non-dominated ranked genetic algorithm (NRGA) are developed to solve the model. In contrast to classic solution methods that convert a multi-objective model to a single-objective one, we solve the model as a multi-objective to make a suitable trade-off between the TWT and the EC. One of the most popular and applicable algorithms to solve a multi-objective optimization problem is NSGA-II that was introduced by Deb, et al. (2002). The implementation steps of NSGA-II is shown in Fig1.The steps of NRGA, first introduced by Al Jadaan, et al. (2009), is similar to NSGA-II with the exception in the step of parent selection.

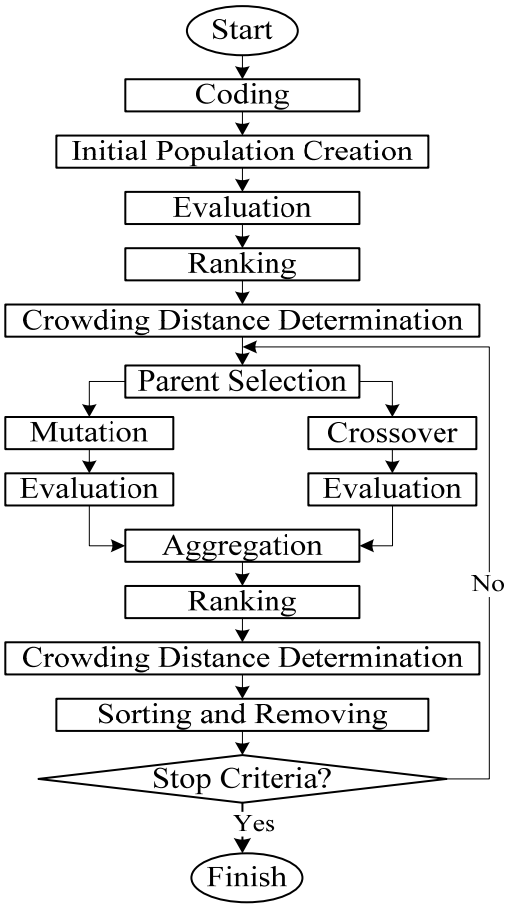

Fig1. Flowchart of NSGA-II and NRGA

Step 1: Coding and creation of initial population

Each chromosome is composed by "number of machines + (number of jobs - 1) × number of stages" genes. So that each chromosome will have the number of workshop parts. Each part includes a sequence of "number of jobs + number of related stage jobs - 1" which are placed randomly. Minus the number of 1 is because of the lack of need to display the latest job at each stage. Also, the sequence of jobs at all stages cannot be considered the same due to consider of energy consumption function in the proposed chromosome. Based on the proposed solution-representation, a set of chromosomes (as much as $nPop$) are generated randomly as the initial population.

Step 2: Evaluation and ranking

The objective values of the initial population are evaluated using an evaluation function. Then, the population is ranked based on the non-domination sorting procedure to create Pareto fronts. Each chromosome of the population under evaluation obtains a rank equal to its non-domination level, where the first front level as the best one contains chromosomes with the smallest rank, the second front as the next-best one corresponds to the chromosomes with the second rank, and so on.

Step 3: Crowding distance (CD) determination

The CD is calculated for members of each front level in comparison with other members of the same level. For solution i, the average distance from two adjacent solutions is calculated regard to each objective function and the summation is CD_ias shown in equation (23), where m is the number of objective functions.

$$CD_i = \sum_{m=1}^{N} \frac{f_m^{i+1} - f_m^{i-1}}{f_m^{max} - f_m^{min}} \tag{23}$$

Step 4: Parent selection

Since the only difference between NSGA-II and NRGA algorithms is in the step of parent selection, the implementation of these two algorithms in this step is presented as follows separately.

➢ *Parent selection for NSGA-II*

In the modified algorithm for NSGA-II, there is no limitation on use of two members in order to selection of parent and the tournament size may be considered more than two members. This will cause high probability of selecting better chromosomes in order to create high quality offspring. Therefore, each chromosome participates with better solution in the tournament. In equality of ranks of two solutions, the second criteria (CD) will be evaluated to select the best one.

Fig2 shows the structure of chromosome with 4 jobs and 3 stages. The number of unrelated parallel machines at stages is 3, 1, and 2 respectively. The numbers that are greater than the number of jobs are chosen as parallel machines at desired stage. Thus, the first number that is greater than the size of the set of jobs represents the first machine, and the second number represents the second machine, and so on. Jobs located before each machine is allocated to that machine, and the remained jobs are allocated to the last machine. Two parts are considered for evaluation of the chromosome. These two parts represents two different results with the same sequence and allocation. The first and second results are related to values of objective functions in the lack of idle time and in the existence of idle time respectively. After performing pairwise comparisons, all chromosomes are classified based on frequency of dominated solutions in the Pareto frontier. Then a rank is assigned for each solution based on its Pareto counter due to its location in the Pareto frontier.

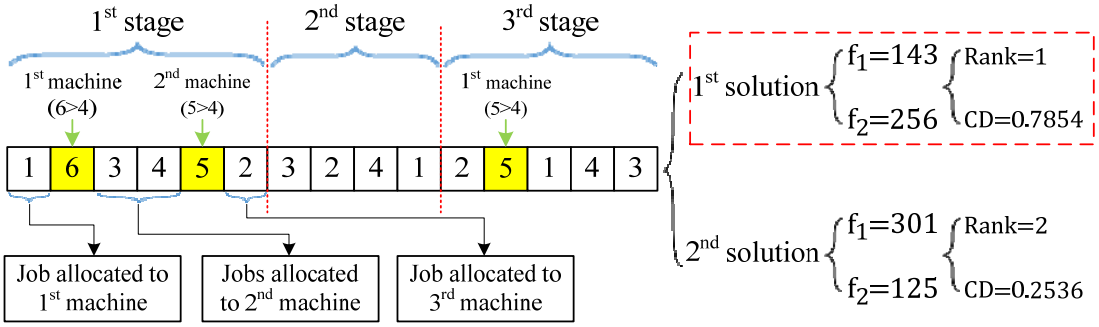

Fig2. How to allocate jobs to machines, evaluate and chose a chromosome for tournament.

➢ *Parent selection for NRGA*

In the modified NRGA algorithm, the parent selection is performed by roulette wheel. Here, modified Boltzmann method is used as equation (24).

$$p_i = \frac{e^{-\beta \times \frac{R_i}{R_{max}}}}{\sum_{i=1}^{R_{max}} e^{-\beta \times \frac{R_i}{R_{max}}}} \tag{24}$$

where,

p_i Probability of selecting ith frontier

R_i Number of ith frontier

R_{max} Maximum number of frontiers

β Selection pressure

If the value of β becomes high, the chance of selecting worse frontier will decrease. Else, the chance will increase. Therefore, the value of β will be determined in parameter tuning for different test problems. After determining the probability of solutions, the parent is specified by performing roulette wheel and consequently, the related chromosome is used to generate offspring.

Step 5: Crossover and evaluation

The point that should be noticed is the need to search more space because of the properties of the problem of this paper and the lack of equality of jobs at all stages. At the first, this process is performed at each stage separately. Next, all performed crossovers at each stage will be added to the sum of generated offspring in the form of two offspring from these parents in order to apply an overall crossover (at all stages). Fig 3 shows the offspring from a single-point crossover for a sample with

two stages as an example. After performing crossover on parents, first, each chromosome of offspring will be decoded and objective functions will be calculated for them. Then, pairwise comparisons will be done among offspring to determine ranking and CD. Two best chromosomes will be accepted as two offspring created by these parents and the rest will be removed.

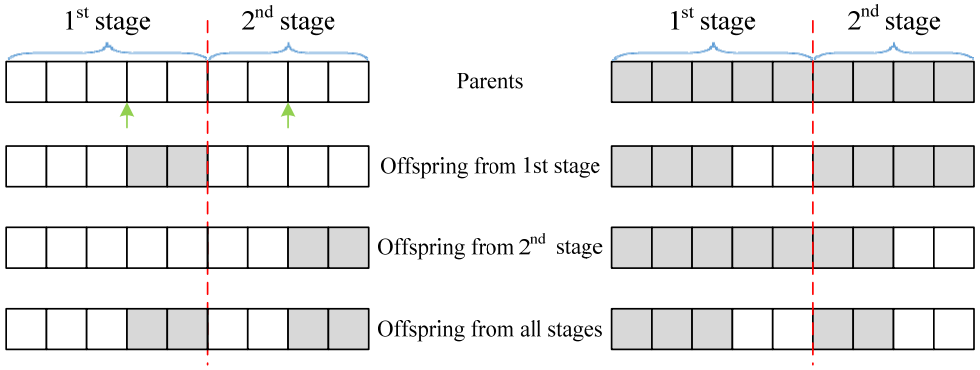

Fig 3. Offspring from a single-point crossover for a sample with two stages

Step 6: Mutation and evaluation

Four types of mutations are used in the proposed algorithm. These four types of mutations are insertion, swap, reversion, and flip. Fig 4 shows types of mutations used in this paper. To increase the search space (mainly done locally), first, the mutation process is performed at each stage separately. Then, all mutations made at each stage will be added to the offspring of considered parent in the form of new offspring in order to apply an overall mutation (at all stages). Fig 5 shows the offspring from a swap mutation for a sample with two stages as an example. After performing mutation on parents, first, each chromosome of offspring will be decoded and objective functions will be calculated for them. Then, pairwise comparisons will be done among offspring to determine ranking and CD. Two best chromosomes will be accepted as two offspring created by these parents and the rest will be removed.

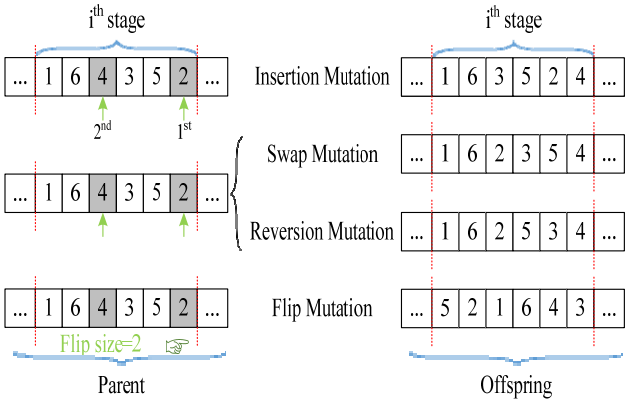

Fig 4. Types of mutations used in this paper

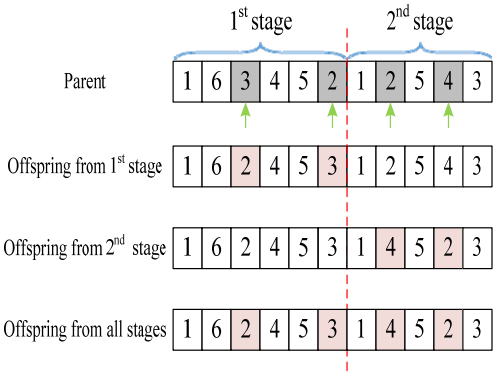

Fig 5. Offspring from a swap mutation for a sample

Step 7: Aggregation, ranking, CD determination, sorting, and removing

Offspring generated by crossover and mutation are aggregated into the main population and constitute a bigger population named R_t.All solutions obtained by R_t chromosomes are compared and classified within the Pareto frontier levels. Then, due to the location of each solution at the frontier a rank is assigned for it based on frontier level. CDs are calculated for each member of R_t population (like Step 3).Each chromosome must do its upmost and proof its competency to survive in the next generation. So, each chromosome will participate in sorting with its best solution. First, the chromosomes are sorted by their ranks. In the case of ranks equality, the second criteria (CD) will be the sorting base. Then, members are chosen from the top of ordered R_t population in the size of initial population ($nPop$) and make the next generation.

Step 8: Investigating stop criteria

In order to recognize the end of the implementation of the proposed algorithms, the number of function evaluation (NFE) criteria is used. NFE is considered 2.5×10^5 in this paper. If it is realized, the proposed algorithm stops, else, it will return to step 4.

4-Computational results

The computer coding of both algorithms are implemented using the MATLAB and the experiments are performed on a computer with 2.60 GHz of 3537U CPU and 4.00 GB of RAM. Six random test problems with different sizes (small, medium, and large) have been generated based on the parameters illustrated in Table 2.

Table 2. Parameter value for test problems

Parameter	Class of the problem		
	Small size	**Medium size**	**Large size**
Number of jobs	5	10	15
Number of stages	2,3	4,5	6,7
Number of machines at each stage	U(1,3)	U(2,4)	U(3,5)
Weights of jobs	U(1,9)	U(1,9)	U(1,9)
Due dates	U(50,120)	U(80,150)	U(120,220)
Release times	U(1,15)	U(1,15)	U(1,15)
Process time	U(25,95)	U(25,95)	U(25,95)
Set up time of first job	U(2,14)	U(2,14)	U(2,14)
Set up time between jobs	U(3,25)	U(3,25)	U(3,25)
On/off energy	U(40,100)	U(40,100)	U(40,100)
Process time energy	U(1,10)	U(1,10)	U(1,10)
Set-up time energy	U(5,15)	U(5,15)	U(5,15)
Idle time energy	U(10,20)	U(10,20)	U(10,20)

In order to minimize the effect of noise factors on the performance of the proposed algorithms, a Taguchi procedure is utilized to set the parameters of the algorithms (see Table 3).

Table 3. Parameter ranges and levels with the selected values for NSGA-II and NRGA

Algorithm parameters	**Parameters range**	**Low**	**Medium**	**High**	**Selected for NSGA-II**	**Selected for NRGA**
Population size	50-100	50	70	100	50	100
Cross-over ratio	0.5 - 0.9	0.5	0.7	0.9	0.5	0.5
Mutation ratio	0.05 – 0.15	0.05	0.1	0.15	0.1	0.05

Each algorithm runs six times for each of the test problems and the performances are compared in terms of the means of four indicators; (i) the quality metric (QM), (ii) the mean ideal distance (MID) introduced by Zitzler and Thiele (1998), (iii) the spacing metric (SM), and (iv) the diversification metric (DM).Moreover, the small sized test problems have been solved using the ε constrained (ε-c) method to evaluate the efficiency of the proposed algorithms. Finally, for each of the test problems, a total weighted normalized indicator (TWNI) is calculated and is used to compare the results of two proposed algorithms. The indicators are obtained from the following formulas:

$$\mathrm{QM} = \frac{\sum_i \alpha_i}{nPop}$$

$$MID = \frac{\sum_{i=1}^{n} C_i}{n}$$

$$SM = \sqrt{\frac{\sum_{i=1}^{n-1} (\bar{d} - d_i)^2}{(n-1)}}$$

$$TWNI_i = \sum_{j=1}^{4} R_{ij} w_j$$

$$DM = \sqrt{\sum_{i=1}^{n} d_i^{'}} \ , d_i^{'} = \max_{j} \left\{ \sum_{m=1}^{N} (f_i^m - f_j^m)^2 \right\}$$

where,

α_i 1, if solution i belongs to Pareto front; 0, otherwise

n The number of non-dominated solutions

C_i The Euclidian distance of i[th] non-dominated solution from the ideal point

d_i The Euclidian distance for two adjacent non-dominated solutions ($\bar{d}$ is the mean value of d_i)

f_i^m The value of objective function for the i[th] non-dominated solution

R_{ij} The normalized value of indicator j for solution i

w_j The weight of indicator j , here $(w_{\mathrm{QM}}, w_{\mathrm{MID}}, w_{\mathrm{SM}}, w_{\mathrm{DM}}) = (10^2, 10, 1, 1)$

For a test problem with two stages, the convergence patterns of the number of Pareto solutions obtained from a run of algorithms are displayed in Fig 6 for 50,000 number of function evaluations (NFE). The number of Pareto solutions has reached a maximum value in a little more than 12,000 NFEs for the NSGA-II, while this value is less than 17,000 NFEs for the NRGA.

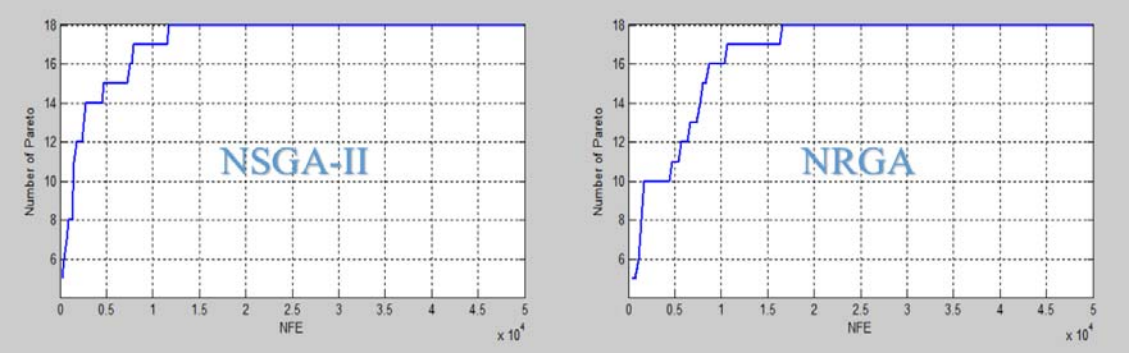

Fig 6. Convergence patterns of a run of algorithm for a test problem with two stages

The results obtained for each of the test problems are shown in Table 4. For the small sized problems, the results show while the difference among the epsilon-constrained method and the two proposed algorithms is not significant (less than 12%) in terms of the means of MID, SM, and DM, there are a difference between 14% to 33% in term of the means of QM. Moreover, while there are no significant differences among the two proposed algorithms for the small sized problems, there are significant differences among the proposed algorithms for medium and large sized problems in terms of the means of QM, MID, SM, and DM, and the NRGA significantly outperforms the NSGA-II in all four indicators.

Table 4. Computational results obtained from the NSGA-II and NRGA for the FFSP test problems

Problem size			**QM**			**MID**			**SM**			**DM**		
Job	Stage	Machine	NSGA-II	NRGA	ε-c	NSGA-II	NRGA	ε-c	NSGA-II	NRGA	ε-c	NSGA-II	NRGA	ε-c
5	2	U(1,3)	0.742	0.773	0.909	1132.1	1131.4	1073.5	353.257	353.738	341.364	231.925	231.952	245.543
5	3	U(1,3)	0.575	0.644	0.862	1816.8	1759.1	1614.6	261.971	259.615	256.672	294.698	311.244	333.035
10	4	U(2,4)	0.054	0.128	---	7431.2	7275.1	---	651.384	541.359	---	951.603	1033.10	---
10	5	U(2,4)	0.046	0.144	---	10179	8860.9	---	457.838	515.381	---	1032.60	1176.40	---
15	6	U(3,5)	0.039	0.137	---	18088	16171	---	619.273	483.563	---	1347.20	1694.00	---
15	3	U(1,3)	0.020	0.242	---	31563	29013	---	1065.60	1455.60	---	1780.80	2574.20	---

Fig 7shows the value of TWNI for six test problems which demonstrates the better performance of the NRGA than NGSA-II.

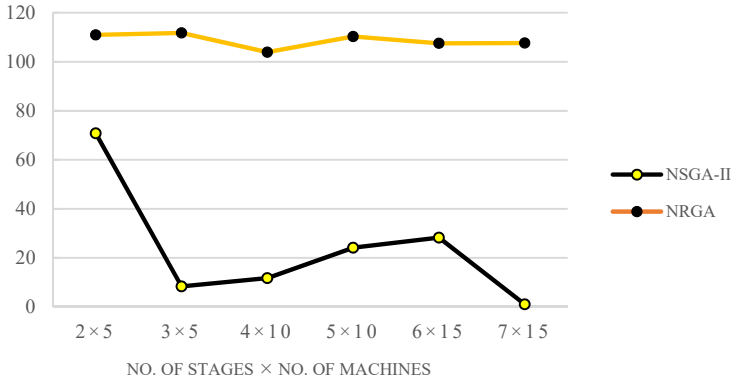

Fig 7. Values of indicator of TWNI obtained from proposed algorithms for six test problems

5-Conclusions and future research directions

In this paper, a more applicable to reality model for the flexible flow shop scheduling problem was investigated with the goals of minimizing both the total weighted tardiness and the energy consumption. The set-up times are considered to be sequence- and machine-dependent, and a release time and a due date is applied for each of the jobs. The problem was first formulated into a bi-objective model and then two multi-objective evolutionary algorithms, namely NSGA-II and NRGA were utilized to find Pareto front solutions of the model. Taguchi method was employed to tune the parameters of algorithms. Six numerical examples in small, medium, and large sizes were next generated randomly to demonstrate the application of the proposed algorithms. The results showed that the NRGA has significant better performance in comparison with the NSGA-II for all six test problems in terms of all four indicators, i.e., (i) the quality metric, (ii) the mean ideal distance, (iii) the spacing metric, and (iv) the diversification metric. In this paper, it was supposed that all parameters of the model are deterministic. As a future research suggestion, one can consider the uncertainty of some parameters, especially the release time of the jobs, the set-up time of the machines, and the processing time of the jobs, using stochastic programming, robust optimization or simulation approaches. Moreover, as another research direction, one can employ some other meta-heuristic or matheuristic algorithms, and he can compare them with the NRGA.

References

Asif, M., & Muneer, T. (2007). Energy supply, its demand and security issues for developed and emerging economies. *Renewable and Sustainable Energy Reviews*, *11*(7), 1388-1413.

Choi, S. W., Kim*, Y. D., & Lee, G. C. (2005). Minimizing total tardiness of orders with reentrant lots in a hybrid flowshop. *International Journal of Production Research*, *43*(11), 2149-2167.

Dahmus, J. B., & Gutowski, T. G. (2004, January). An environmental analysis of machining. In *ASME 2004 international mechanical engineering congress and exposition*(pp. 643-652). American Society of Mechanical Engineers.

Dai, M., Tang, D., Giret, A., Salido, M. A., & Li, W. D. (2013). Energy-efficient scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm. *Robotics and Computer-Integrated Manufacturing*, *29*(5), 418-429.

Davoudpour, H., & Ashrafi, M. (2009). Solving multi-objective SDST flexible flow shop using GRASP algorithm. *The International Journal of Advanced Manufacturing Technology*, *44*(7-8), 737-

747.
Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, *6*(2), 182-197.

Drake, R., Yildirim, M. B., Twomey, J. M., Whitman, L. E., Ahmad, J. S., & Lodhia, P. (2006). Data collection framework on energy consumption in manufacturing.

Ebrahimi, M., Ghomi, S. F., & Karimi, B. (2014). Hybrid flow shop scheduling with sequence dependent family setup time and uncertain due dates. *Applied Mathematical Modelling*, *38*(9-10), 2490-2504.

Fang, K., Uhan, N., Zhao, F., & Sutherland, J. W. (2011). A new approach to scheduling in manufacturing for power consumption and carbon footprint reduction. *Journal of Manufacturing Systems*, *30*(4), 234-240.

Gupta, J. N., Krüger, K., Lauff, V., Werner, F., & Sotskov, Y. N. (2002). Heuristics for hybrid flow shops with controllable processing times and assignable due dates. *Computers & Operations Research*, *29*(10), 1417-1439.

Gutowski, T., Murphy, C., Allen, D., Bauer, D., Bras, B., Piwonka, T., ... & Wolff, E. (2005). Environmentally benign manufacturing: observations from Japan, Europe and the United States. *Journal of Cleaner Production*, *13*(1), 1-17.

JADAAN, O. A., RAJAMANI, L., & Rao, C. R. (2009). NON-DOMINATED RANKED GENETIC ALGORITHM FOR SOLVING CONSTRAINED MULTI-OBJECTIVE OPTIMIZATION PROBLEMS. *Journal of Theoretical & Applied Information Technology*, *5*(5).

Jun, S., & Park, J. (2015). A hybrid genetic algorithm for the hybrid flow shop scheduling problem with nighttime work and simultaneous work constraints: A case study from the transformer industry. *Expert Systems with Applications*, *42*(15-16), 6196-6204.

Kerzner, H., & Kerzner, H. R. (2017). *Project management: a systems approach to planning, scheduling, and controlling*. John Wiley & Sons.

Lee, G. C., & Kim*, Y. D. (2004). A branch-and-bound algorithm for a two-stage hybrid flowshop scheduling problem minimizing total tardiness. *International Journal of Production Research*, *42*(22), 4731-4743.

Lee, G. C., Kim, Y. D., & Choi, S. W. (2004). Bottleneck-focused scheduling for a hybrid flowshop. *International Journal of Production Research*, *42*(1), 165-181.

Li, D., Meng, X., Liang, Q., & Zhao, J. (2015). A heuristic-search genetic algorithm for multi-stage hybrid flow shop scheduling with single processing machines and batch processing machines. *Journal of Intelligent Manufacturing*, *26*(5), 873-890.

Liang, P., Yang, H. D., Liu, G. S., & Guo, J. H. (2015). An ant optimization model for unrelated parallel machine scheduling with energy consumption and total tardiness. *Mathematical Problems in Engineering*, *2015*.

LIN, Z. Y., & Zhang, W. (2014). Study on Multi-Objective Optimization of Hybrid Flow Shop Scheduling Problem. *Advanced Materials Research*, *1082*.

Liu, Y., Dong, H., Lohse, N., & Petrovic, S. (2015). Reducing environmental impact of production during a rolling blackout policy–a multi-objective schedule optimisation approach. *Journal of Cleaner*

Production, *102*, 418-427.

Liu, Y., Dong, H., Lohse, N., Petrovic, S., & Gindy, N. (2014). An investigation into minimising total energy consumption and total weighted tardiness in job shops. *Journal of Cleaner Production*, *65*, 87-96.

Luo, H., Du, B., Huang, G. Q., Chen, H., & Li, X. (2013). Hybrid flow shop scheduling considering machine electricity consumption cost. *International Journal of Production Economics*, *146*(2), 423-439.

Naderi, B., Zandieh, M., & Roshanaei, V. (2009). Scheduling hybrid flowshops with sequence dependent setup times to minimize makespan and maximum tardiness. *The International Journal of Advanced Manufacturing Technology*, *41*(11-12), 1186-1198.

Pechmann, A., & Schöler, I. (2011). Optimizing energy costs by intelligent production scheduling. In *Glocalized Solutions for Sustainability in Manufacturing* (pp. 293-298). Springer, Berlin, Heidelberg.

Pinedo, M., Zacharias, C., & Zhu, N. (2015). Scheduling in the service industries: An overview. *Journal of Systems Science and Systems Engineering*, *24*(1), 1-48.

Rashidi, E., Jahandar, M., & Zandieh, M. (2010). An improved hybrid multi-objective parallel genetic algorithm for hybrid flow shop scheduling with unrelated parallel machines. *The International Journal of Advanced Manufacturing Technology*, *49*(9-12), 1129-1139.

Ruiz, R., & Vázquez-Rodríguez, J. A. (2010). The hybrid flow shop scheduling problem. *European journal of operational research*, *205*(1), 1-18.

Song, W., Zhang, C., Lin, W., & Shao, X. (2014). Flexible job-shop scheduling problem with maintenance activities considering energy consumption. *Applied Mechanics & Materials*, (521).

Tang, D., Dai, M., Salido, M. A., & Giret, A. (2016). Energy-efficient dynamic scheduling for a flexible flow shop using an improved particle swarm optimization. *Computers in Industry*, *81*, 82-95.

Tavakkoli-Moghaddam, R., Javadian, N., Khorrami, A., & Gholipour-Kanani, Y. (2010). Design of a scatter search method for a novel multi-criteria group scheduling problem in a cellular manufacturing system. *Expert Systems with Applications*, *37*(3), 2661-2669.

Yan, J., Li, L., Zhao, F., Zhang, F., & Zhao, Q. (2016). A multi-level optimization approach for energy-efficient flexible flow shop scheduling. *Journal of Cleaner Production*, *137*, 1543-1552.

Yan, J. H., Zhang, F. Y., Li, X., Wang, Z. M., & Wang, W. (2014). Modeling and Multiobjective Optimization for Energy-Aware Hybrid Flow Shop Scheduling. In *Proceedings of 2013 4th International Asia Conference on Industrial Engineering and Management Innovation (IEMI2013)* (pp. 741-751). Springer, Berlin, Heidelberg.

Zitzler, E., & Thiele, L. (1998, September). Multiobjective optimization using evolutionary algorithms—a comparative case study. In *International conference on parallel problem solving from nature* (pp. 292-301). Springer, Berlin, Heidelberg.