Journal of Quality Engineering and Production Optimization

Vol. 3, No. 2, PP. 77-98, 2018

Fuzzy Multi-objective Permutation Flow Shop Scheduling Problem with Fuzzy Processing Times under Learning and Aging Effects

Farid Najari[1], Mohammad Alaghebandha[1], Mohammad Mohammadi[1*] and Mohammad Ali Sobhanallahi[1]

1. Department of Industrial Engineering,Faculty of Engineering,Kharazmi University, Tehran, Iran.

[*]***Corresponding Author:*** *Mohammad Mohammadi (E-mail: mohammadi@khu.ac.ir)*

Abstract-- *In industries machine maintenance is used in order to avoid untimely machine fails as well as to improve production effectiveness. This research regards a permutation flow shop scheduling problem with aging and learning effects considering maintenance process. In this study, it is assumed that each machine may be subject to at most one maintenance activity during the planning horizon. The objectives aim to minimize the makespan, tardiness of jobs, tardiness cost while maximizing net present value, simultaneously. Due to complexity and Np-hardness of the problem, two Pareto-based multi-objective evolutionary algorithms including non-dominated ranked genetic algorithm (NRGA) and non-dominated sorting genetic algorithm (NSGA-II) are proposed to attain Pareto solutions. In order to demonstrate applicability of the proposed methodology, a real-world application in polymer manufacturing industry is considered.*

Keywords*: Permutation flow shop; Aging and Learning Effect; Maintenance; Case study.*

I. INTRODUCTION

In classical scheduling models the processing time of jobs are regarded as fixed numbers. However, in real-world applications, processing time of jobs may reduce since the machine and worker performance improve during the production. For instance, workers performance is improved by repeating the same task over time. In industry this phenomenon is referred to as learning effect in manufacturing plants and is proved by scientists (Ghemawat 1985, Webb 1994; Cheng et al., 2010).

(Murata et al. 1996) used a algorithm (GA) in order to solve a flow shop scheduling problem. (Hyun et al. 1998) proposed novel selection roles that can be used in GA. They show that the proposed algorithm can solve the scheduling problem very efficiently. (Reddy & Narendran 2003) introduced a heuristic algorithm for sequence-dependent flow shop group scheduling problem. The authors regarded different conditions such as non-availability of jobs to increase applicability of the proposed model. (Wilson et al. 2004) extended the heuristic algorithm for multiple setups per group. They used the heusistic algorithm in GA and aimed to solve non-similar groups scheduling problem.

(Kuo & Yang 2006a) and (Kuo & Yang2006b) aimed to introduce a scheduling problem with learning effect formulation that considered processing times of processed jobs in order to calculate the processing time of jobs in the sequence. The goal of the authors was to minimize the total completion time and makespan in polynomial time. (Lee & Wu 2004) regarded a two-machine flow shop scheduling problem that targeted minimization of total completion time. (Wu et al. 2007) investigated a two-machine flow shop scheduling problem with effect of learning process. The authors used a Brand and Bound algorithm as well as Simulated Annealing (SA) for optimization. (Biskup 2008) reviewed research papers in order to highlight importance of learning effect in scheduling problems. However, learning has been rarely investigated in flow shop problems. (Eren & Guner 2008) proposed an integer programming formulation for a two-machine flow shop scheduling problem with effect of learning. They aimed at minimization of total completion time and makespan. In order to solve the problem in small and large sizes, two different heuristic approaches were utilized by the authors.

(Wu & Lee 2009) introduced a new algorithm role for solving permutation flow shop scheduling problem with

Manuscript Received:12 -July-2016 and revised: 01-Aug-2017 *ISSN: 2423-3781*
Accepted: 10-Oct-2018

learning while aiming at minimization of total flow times.

Besides non-increasing models, there exist other formulations that the job processing times are calculated using a non-decreasing model. For instance, two types of these kind of models can be highlighted. The first type is "deteriorating effect" in which processing times are non-decreasing formulations of the start times. In the second type which is called "aging effect" the processing times are regarded as non-decreasing functions. The second type attracted a lot of attention among researchers (Chang et al., 2009); (Janiak & Rudek, 2010).

The "aging effect" models can also be classified in two types. In the first type, called "position dependent based aging effect" the processing times are regarded as non-decreasing formulations (Liu et al., 2010). In the second type, processing times are considered as a non-decreasing formulation that uses the normal processing times (i.e. required time to complete a job without considering aging effect) of scheduled jobs (Wang et al., 2009a); (Wang et al. 2009b); (Lai & Lee 2011).

(Rudek & Rudek 2012) analyzed minimization of makespan in a flow shop scheduling problem regarding aging effect and allocation of resources. The authors proved that in some cases the problem can be solved in polynomial time. (Vahedi-Nouri et al. 2013) presented a non-permutation flow shop scheduling model regarding learning effects. They also considered machine availability constraints to increase applicability of the proposed model. The aim was minimizing the total flow time. (Let et al. 2014) regarded a permutation flow shop scheduling problem with deteriorating jobs. The aim was minimization of the total tardiness. A branch-and-bound algorithm with dominance properties was utilized to solve the model. (Yeh et al. 2014) considered a parallel machine scheduling under learning effects in which processing times where considered as fuzzy numbers. They target minimization of the makespan. (Wang & Zhang. 2015) studied the permutation flow shop problem that aimed to minimize the makespan and total completion time in which learning effects affect the processing times. (Rostami et al 2015) investigates a non-identical parallel machine scheduling problem with fuzzy processing times. The authors also considered deterioration and learning effects in the problem. In order to solve the model, a multi-objective branch and bound algorithm was used and efficient Pareto optimal solutions were attained.

In order to highlight contribution of this research, Table I is presented that provides a complete literature review in the field of research.

TABLE I. Characteristics of the most recent and related published models

Reference	Main problem	Learning	Aging	Maintenance
Gawiejnowicz(1996)	SMFS	✓		
Eren and Guner (2003)	TMFS	✓		
Lee and Wu (2004)	TMFS	✓		
Kuoa and Yang(2006)	SMFS	✓		
Wu et al. (2007)	TMFS	✓		
Eren and Guner (2008)	TMFS	✓		
Dirk Biskup(2008)	SMFS	✓		
W-H Kuo and D-L Yang(2008)	SMFS		✓	
Wu and Lee (2009)	PFSS	✓		
Biskup(1999)	SMFS	✓		
Chang et al. (2009)	SMFS	✓	✓	
Janiak and Rudek(2010)	SMFS		✓	
Yang and Yang (2010)	SMFS		✓	✓
Low and Lin(2011)	SMFS	✓		
Lai and Lee(2011)	SMFS	✓		
Rudek and Rudek (2012)	PFSS		✓	
Cheng et al. (2013)	SMFS	✓		
Wang and Zhang. (2015)	PFSS	✓		✓
Rostami et al (2015)	Parallel-machine Scheduling	✓		✓
This paper	PFSS	✓	✓	✓

SMFS: Single-machine flow shop
TMFS: Two-machine flow shop
PFSS: Permutation Flow Shop Scheduling

Based on Table I, most researches have considered learning effects in their mathematical models. However, few

studies have investigated the problem with aging effect. In literature, processing times are considered as known and constant numbers. However, in real-world applications, the processing times is not deterministic. In order to handle uncertainties, fuzzy sets theory introduced by (Zadeh 1965) is applied. Fuzzy sets can contribute to enhance the applicability of the proposed model in real-world situations fulfilling real-world user demands. It is implemented for handling flexible constraints as well as uncertain parameters. This research considers a permutation flow shop scheduling problem with learning and aging while considering maintenance activities. The aim is minimizing makespan, tardiness of jobs and tardiness cost and maximizing net present value, simultaneously in fuzzy environment.

The remainder of this research is organized as follows: In Section 2, the problem is clarified and the model is formulated. Section 3 provides Multi-objective solution methodology. In Section 4 experimental design is illustrated, and Section 5 presents the conclusion.

II. PROBLEM FORMULATION

Flow shop scheduling problem has extensively been studied in recent years. However, most researchers considered the problem as a single objective mathematical model. Considering a single objective is not realistic. Multi-objective flow shop scheduling problem consists of various conflicting objective functions, which often results in more intricate problem (Karimi et al. 2010). In addition, downtime cost, which is obtained by summing up cost elements related to activities required to bring the system back into service is considered. Therefore, in order to consider the total downtime cost, reparation costs of failed elements are regarded. In real-world situations, the precise time and cost of tasks is not known in prior to starting the process. However, it can be estimated by experts, e.g., processing time and maintenance cost. Moreover, fuzzy cash flow states the outward and inward cash flows including resource costs, material costs, and payments from client (owner) to contractor (whose task is to start the activity). Considering possible values for the time/cost, an expert may evaluate whether the intervals appear to be more reasonable. Therefore, fuzzy intervals or fuzzy numbers are the best approaches to formulate such durations. A discount is considered at the begining of activity i with a discount rate of φ. In following subsections notations, parameters, and mathematical formulation is presented. The objective functions are formulated based on cost according to (Rostami et al. 2015) method and triangular fuzzy numbers transformed to crisp numbers by results of this paper.

A. Indices

t Time index

i Machines set

j Jobs set

k Positions index

n Number of jobs

B. Parameters

$\tilde{C}_{Di}$ Fuzzy downtime cost

$\tilde{C}_{Pi}$ Fuzzy maintenance cost

$\widetilde{CF}_{\mathrm{i}}$ Fuzzy cash flow

$D(t)$ Downtime function

G Mean downtime per failure

$\tilde{E}(t)$ Fuzzy mean downtime per inspection interval (t interval)

Λ Rate of arrival of defects from all components

z_{jk} 1 if job j is located as kth job; Otherwise 0

y_{ik} 1 if maintenance is prior to the kth job; Otherwise 0

$\tilde{p}_{ij}$ Fuzzy processing time of job j on machine i

$\tilde{p}_{ik}^{A}$ Actual processing time of the processed job in position k on machine i

A Factor of learning

b_{ik} Aging factor of kth job k on machine i

$\tilde{C}_{ik}$ Fuzzy completion time of the kth job on machine i

$\tilde{t}_i$ Fuzzy Duration of maintenance on machine i

r_{ik} kth job k related to the maintenance activity on machine i

x_{il} 1 if machine i is selected for position l and 0, otherwise

$\tilde{s}_{ij}$ Fuzzy starting time of operation

$\tilde{C}_{Pmax}$ Fuzzy available budget for all maintenance

C. Decision variables

$\tilde{C}_{Tr}$ Fuzzy tardiness cost

$\tilde{T}_r$ Fuzzy tardiness of job

$\tilde{C}_{max}$ Maximum fuzzy completion time

NPV Expected net present value

Therefore, the proposed mathematical model is presented as:

$$Min\left\{\tilde{C}_{max}, \tilde{T}_r, \tilde{C}_{Tr}\right\} \quad (1)$$

$$\text{Max NPV} = \sum_{i=1}^{n} \widetilde{CF}_i \times e^{-\varphi d_i^1} \quad (2)$$

Subject to:

$$\tilde{p}_{ik}^{A} = \left(\sum_{j=1}^{n} z_{jk}\tilde{p}_{ij}\right)\left(1 + \sum_{l=1}^{k-1}\tilde{p}_{il}\right)^{\alpha} + b_{ik}r_{ik} \quad \forall i \tag{3}$$

$$\tilde{C}_{i1} = \sum_{h=1}^{i}\tilde{p}_{h1}^{A} + \tilde{t}_i y_{i1} \qquad \forall i \tag{4}$$

$$\tilde{C}_{1k} = \sum_{l=1}^{k}\tilde{p}_{1l}^{A} + \sum_{l=1}^{k}\tilde{t}_1 y_{1l} \qquad \forall k \tag{5}$$

$$\tilde{C}_{ik} \geq \tilde{C}_{(i-1)k} + \tilde{p}_{ik}^{A} \qquad \forall i,k \tag{6}$$

$$\tilde{C}_{ik} \geq \tilde{C}_{i(k-1)} + \tilde{p}_{ik}^{A} + \tilde{t}_i y_{ik} \qquad \forall i,k \tag{7}$$

$$\tilde{C}_{max} \geq \tilde{C}_{ik} \qquad \forall i,k \tag{8}$$

$$\sum_{j=1}^{n} z_{jk} = 1 \qquad \forall k \tag{9}$$

$$\sum_{k=1}^{n} z_{jk} = 1 \qquad \forall j \tag{10}$$

$$\sum_{k=1}^{n} y_{ik} = 1 \qquad \forall i \tag{11}$$

$$\sum_{i\in m(t)} \tilde{C}_{D_i} + \tilde{C}_{P_i} + \tilde{C}_{Tr_i} \leq \tilde{C}_{Pmax} \tag{12}$$

$$f_i^{\,1} + t_i \leq f_i^{\,2} \qquad \forall i \tag{13}$$

$$\tilde{s}_{ij} + \sum_{k=1}^{n}\tilde{p}_{ij}.y_{ik} \leq \tilde{s}_{i(j+1)} \qquad \forall i,j \tag{14}$$

$$r_{1k} = k - \sum_{l=1}^{k} x_{il} \qquad \forall i,k \tag{15}$$

$$x_{il} \leq +M(1-y_{il}) - 1 \qquad \forall i,l \tag{16}$$

$$x_{il} > +M(1-y_{il}) - 1 \qquad \forall i,l \tag{17}$$

$$x_{il} \leq y_{il} \qquad \forall i,l \tag{18}$$

$$\tilde{s}_{ij} \geq 0 \;\; and \;\; b_{ik} \geq 0 \tag{19}$$

$$\alpha \leq 0 \tag{20}$$

$$z_{jk}, y_{ik} = 0 \;\; or \;\; 1 \;\; ; \;\; \forall i, j, k \tag{21}$$

The first objective function aims at minimization of makespan, tardiness of job and tardiness cost. The second objective function targets maximization of the net present value. Constraints (3) determines the actual processing times considering learning and aging effects, simultaneously.

Constraints (4-5) compute the completion time of the first task at the first position on machines. There constraints also used to calculate completion times on first machine. Constraints (6) estates that each job can be processed on only one machine. Constraints (7) guarantees if machine is not under maintenance, can also handle one job. Constraint (8) determines the makespan. Constraints (9-10) make sure that jobs positions are unique. Constraints (11) shows that the maximum number of maintenance activities is one. Constraints (12) ensures maximum fuzzy cost for encountering with the activity. Constraints (13) indicates that all maintenance activities on machine i are in fuzzy distinct interval. Constraints (14) ensures correct precedence among the operations of a job. Constraints (15) calculates the job and maintenance process positions. Constraints (16-18) determines the number of jobs processed prior to the maintenance. Constraints (19-21) denotes the decision variables.

III. MULTI-OBJECTIVE OPTIMIZERS

In single objective problems, algorithms are exploring the solutions space to find the global optimal solution. However, in multi-objective problems due to conflict among objectives, we are not able to optimize all criteria altogether. Therefore, a set of solutions called Pareto solutions may be determined. Consider a minimization model that consists of conflicting objectives (Rahmati 2012):

$$\begin{aligned} & f(x) = [\, f_1(x), \ldots, f_m(x)] \\ & s.t: \\ & \quad g_i(x) \le 0 \quad ; i = 1, 2, \ldots, m; \quad x \in X \end{aligned} \tag{22}$$

where $x = (x_1, x_2, \ldots, x_n)$ show the decision vector denoting an n–dimensional vector that can obtain any values. In pareto space a dominates b if:

1) $f_i(a) \le f_i(b), \quad \forall i = 1, 2, \ldots, m$

2) $\exists i \in \{1, 2, \ldots, m\} : f_i(a) < f_i(b)$

In this condition, a set of Pareto solutions can be obtained preferably the aim is to obtain Pareto solutions which can not be dominated by other Pareto solutions. The feasible objective space of a multi-objective problem contains Pareto solutions if all the objective cannot be optimized simultaneously by a single solution. In order to obtain Pareto optimal solutions, various techniques can be implemented. In this research, the model is solved using Pareto based multi-objective metaheuristic algorithms. Various efficient metaheuristics in multi-objective optimizations can be used for this aim such as the multi-objective scatter search (MOSS), multi objective genetic algorithm (MOGA), strength Pareto evolutionary algorithm (SPEA2), and non-dominated sorting genetic algorithm (NSGA-II). In this research, non-dominated ranking genetic algorithm (NRGA) and NSGA-II are implemented to solve the model.

A. Non-dominated Sorting Genetic Algorithm (NSGA-II)

NSGA is one of the best multi-objective evolutionary algorithms for optimization of multiple objectives. Despite the superiorities of the algorithm, the algorithm has generally been neglected due to high computational complexity. Another version of multi-objective genetic algorithm is called NSGA-II, which overcomes the disadvantages of NRGA (Deb et al 2002). The pseudo-code of the algorithm is represented as follows (Safari 2011):

- ***Step 1:*** determine the objective functions, the Operations, the stopping criteria, population size and the maximum number of iterations

- ***Step 2:*** Create the initial population.
- ***Step 3:*** while stopping criteria is not meet, do:
    - Create offsprings
    - merge the parents and offspring
    - Sort the solutions to obtain all non-dominated fronts
    - Set $P_{g+1} = \phi$ and i= 1
    - While the parent population size is $|P_{g+1}|+|F_i|< N$ do:

      - calculate crowding-distance

      - add the i*th* non-dominated front

      f -i= i+ 1.

  End while

    - Sort the F_i based on crowding distance.
    - Create parent pop.
    - Generate offspring population.

  End while

- ***Step 4:*** Return non dominated solutions.

Chromosome

The Chromosome in this research, is designed as follow:

We have two different strings of chromosome; the first string is how to allocate works to the situations. In this string we have a row matrix with dimension1*K thus the number of cells in the matrix is equal to the number of positions available; also, numbers in the cells illustrate working number assigned to a specific position. The size of the second string of chromosomes is as same as the first string with the difference that numbers in the cells illustrate the number of machines assigned to a specific position. It should be noted that zero in the second string, which means that the position has not been any real machine. For instance, consider a scheduling problem with 10 jobs and 5 machines then, a chromosome can be defined as follows:

4	11	6	20	12	2	16	19	15	10	8	17	7	5	13	9	1	3	18	14
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	5	3	4	0

Thus, in this chromosome work 4 is assigned to machine 1, work 9 assigned to machine 5, work 1 assigned to machine 2, work 3 assigned to machine 3 and work 18 assigned to machine 4.

Crossover

In GA, offspring are generated using crossover operator that crosses two parents. The aim of crossover operator is to ease optimization process. In the literature, different kinds of crossovers are introduced. In this research, we adapt a single point crossover which is shown in Fig (1). (Moradi et al 2011).

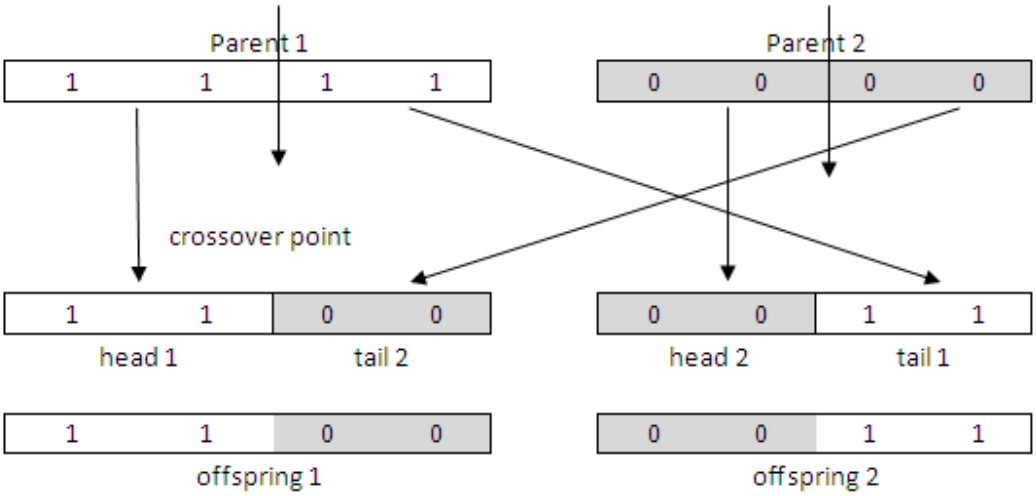

Fig 1. Adjust single point crossover

Mutation

In order to make small changes in the sequences, mutation operator is used. The aim of this operator is to avoid fast convergence. Mutation includes insertion, swapping and reversion of two cells in a given chromosome Fig (2 & 3).

Insertion: Insertion mutation moves a randomly chosen cell to a different randomly chosen cell.

Swap: Swap mutation also exchanges the positions of two randomly selected.

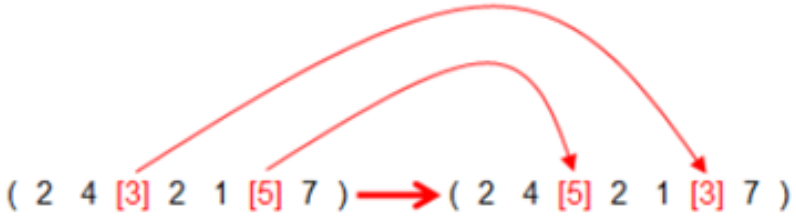

Fig 2. Swap mutation

Reversion: Reversal mutation choses two random cells, and reverses the selected sub-permutation

Fig 3. Reversal mutation

In this research, NRGA and NSGA-II are implemented due to the fact that many research in the literature showed high convergence and coverage of these algorithms that enable them to find a properly distributed Pareto set over Pareto optimal space.

B. Non-dominated Ranking Genetic Algorithm (NRGA)

In this paper, a Non-dominated Ranking Genetic Algorithm (NRGA), presented by Aiello et al (2013), is used, that generates an initial population P and sorts the solutions based on dominance roles. A fitness is assigned to each individual in the population considering its non-domination level. The NRGA solves the problem by repeating the following steps:

in each iteration, the population is sorted based on fitness value, subsequently N/2couples are chosen for crossover, that leads to generate two offspring. The population thus obtained involves 2N elements, that are ranked regarding their dominance level where the first N elements are truncated to shape the population in the next iteration. The mutation operator is used after crossover operator with a given probability. The process is repeated until a specific stopping criterion is reached. The procedure of the proposed NRGA in this paper is summarized in follow:

Initialize population P:

{

Randomly generate a population;
Evaluate objective functions;
Assign rank to each individual regarding dominance roles;
Calculate the crowding distance;
}

Generate child population Q:

{

Rank using roulette wheel;
merge
}

For i=1 : G
Non-dominated sorting;
For for each solution
do
Assign rank
Calculate the crowding distance
end for
select the best members to generate the population of the next generation

Create the next generation:

{

Rank using roulette wheel

merge

}

end for

The scheme of the algorithm is summarized in Fig 4.

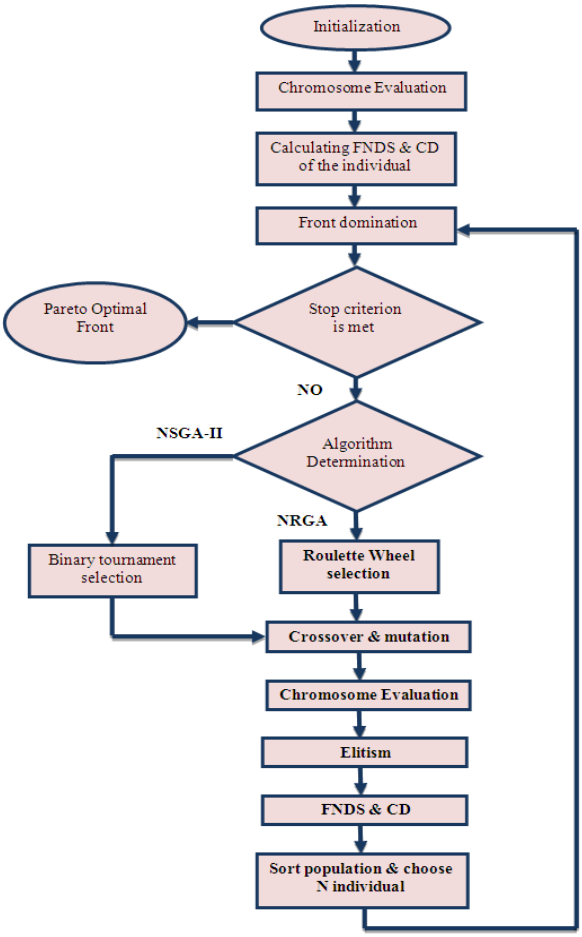

Fig 4. NSGA-II and NRGA procedure

IV. EXPERIMENTAL DESIGN

The subsection aims to obtain the optimal values for input parameters of the algorithms. The values of input parameters may affect the efficiency of the algorithms. For instance, higher population size may help the procedures to find proper solution but this significantly increases computational costs. By considering higher number of chromosomes the diversity of solutions may increase. However, it becomes essential to decrease the number of iterations. Moreover, the crossover, mutation and inversion operators are also considered due to their ability to increase solution quality.

We aim at tuning the parameters of two approaches like (Alaghebandha & Hajipour 2013). In this paper, a Taguchi method is implemented to tune the NRGA and NSGA II.

Regarding the of results of MINITAB software, optimal values of parameters are determined for both algorithms based on higher SNR as presented in Table 2.

TABLE II. Parameter setting of NSGA-II and NRGA

Parameter setting of NSGA-II		**Parameter setting of NRGA**	
Parameter	Value	Parameter	Value
popsize	100	popsize	100
Max_Iteration	[100 200]	Max_ Iteration	[100 150]
Crossover rate	[0.7 0.8]	Crossover rate	[0.65 0.8]
Mutation rate	[0.3 0.4]	Mutation rate	[0.3 0.35]

A. Case study

This research is inspired by a real-world problem treating an ASPC Iranian company in polymer products. In the company, there are 20 products that should be processed on 10 consecutive machines such as HAVER for packing and metal detector machine for identifying metal in bag, check weight machine for adjust bags weight according 25 kg and palletizer machine for used bag placement on pallets and shrink machine for used covering plastic on pallets in Medium density polyethylene and low density polyethylene bagging units that 55 bags of 25 kg polyethylene are packed in the pallets. From each unit of bagging, 300,000 tons of polymer products are imported to the warehouse for export. In this company, in order to reduce aging effect, maintenance is carried out on machines. Furthermore, the maintenance is performed exactly after completion of processing each job in the sequence. The experts demonstrate that the operators' skills are advanced by repeating the same task frequently. We consider Table.A.1 for this case in Appendix 1 and Table.A.2 for the results according to the applied model in the manuscript.

B. Performance metrics for multi-objective algorithms

In order to evaluate efficiency of the algorithms, three different measures are regarded as 1) maximum spread or diversity, 2) spacing, 3) Number of Pareto Solution and 4) mean ideal distance

Maximum Spread or Diversity metric

This criterion is presented by Zitzler (1999), the diameter's length of the cub space by the end values to Non-dominated solutions set. The following relation shows computational procedure.

$$D = \sqrt{\sum_{j=1}^{m} \left(\max_{i} f_i^j - \min_{i} f_i^j \right)^2} \qquad (23)$$

where m is the number of criteria. This measure is calculated using Euclidean distance among solutions. Note that larger D means better solutions.

Spacing metric

In Schott (1995), the spacing metric is defined as follows that shows distance between consecutive solutions.

$$S = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} \left(d_i - \bar{d} \right)^2} \qquad (24)$$

$$d_i = \min_{k \ni nI \; k \neq i} \sum_{j=1}^{m} | f_j^i - f_j^k | \quad ; \quad \bar{d} = \sum_{i=1}^{n} \frac{d_i}{n} \qquad (25)$$

The measured distance is equal to minimum distance of the sum of absolute difference between the measured values of the objective functions between the i*th* solution and solutions in non-dominated set final. Spacing shows standard deviation of the d_i. Consider a set of uniformly distributed solutions, then the value of *s* will be small .Smaller values for S shows higher quality of the obtained Pareto solutions. The maximum spread and spacing metrics illustrated in Fig (5).

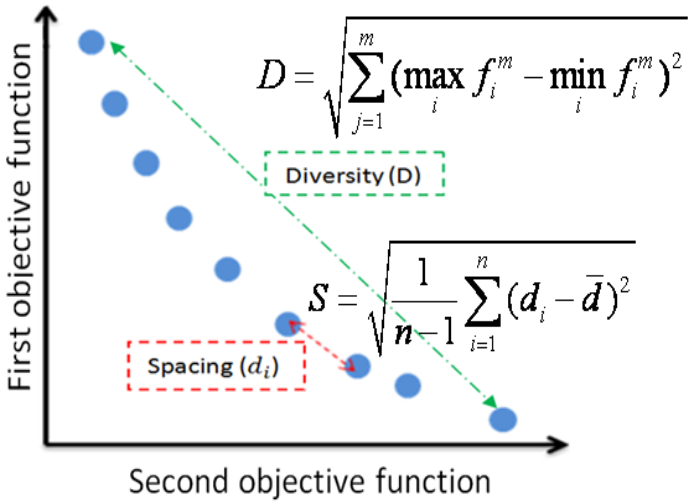

Fig 5. Maximum Spread and spacing

Number of Pareto solutions (NOS) metric

NOS measure shows the number of Pareto optimal solutions determined by each algorithm. Fig.(6) is an example for calculate NOS.

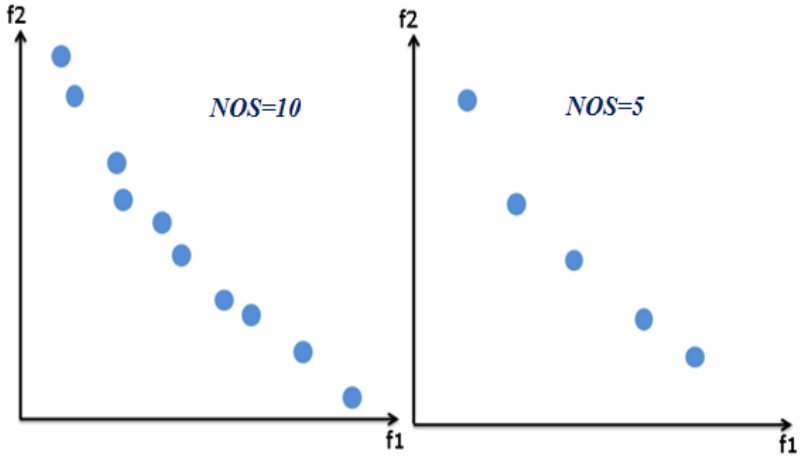

Fig 6. Numberof Pareto solutions for multi-objective problems

Mean Ideal Distance (MID) metric

This measure quantifies the convergence of an algorithm (Karimi et al 2010).

$$MID = \frac{1}{NOS} \sum_{i=1}^{NOS} c_i \qquad (26)$$

where c_i is the distance between each population's member with the best feasible solution. Smaller values for MID shows higher quality of solutions. Fig (7) represents MID metric.

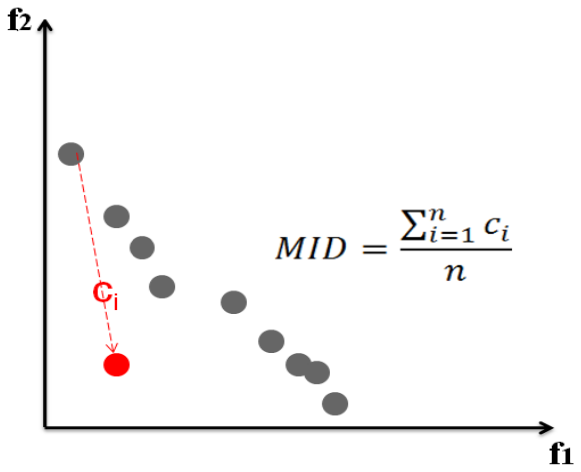

Fig7. distance from ideal solution

We illustrate these metrics in Fig (8) to Fig (11) and Table A.2 for NSGA II and NRGA for our model. In order to evaluate the performances, T-paired statistical test are performed at 95% CL. Then, to determine best solution method, by accepting H0 hypothesis, the p-value must be larger than α. The statistical tests are carried out in MINITAB and summarized in Table A.3 and Table A.4. According to Table A.2 to Table A.4, the proposed NSGA II significantly performs better than NRGA.

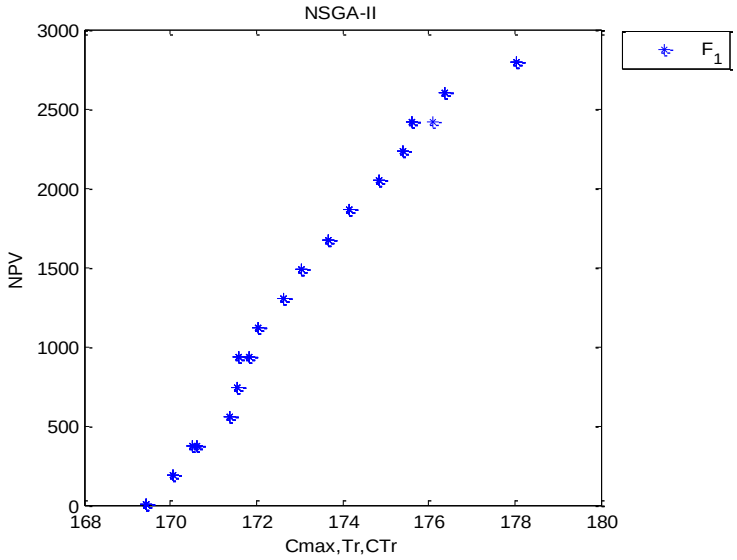

Fig 8. NSGA II in the case study for objectives

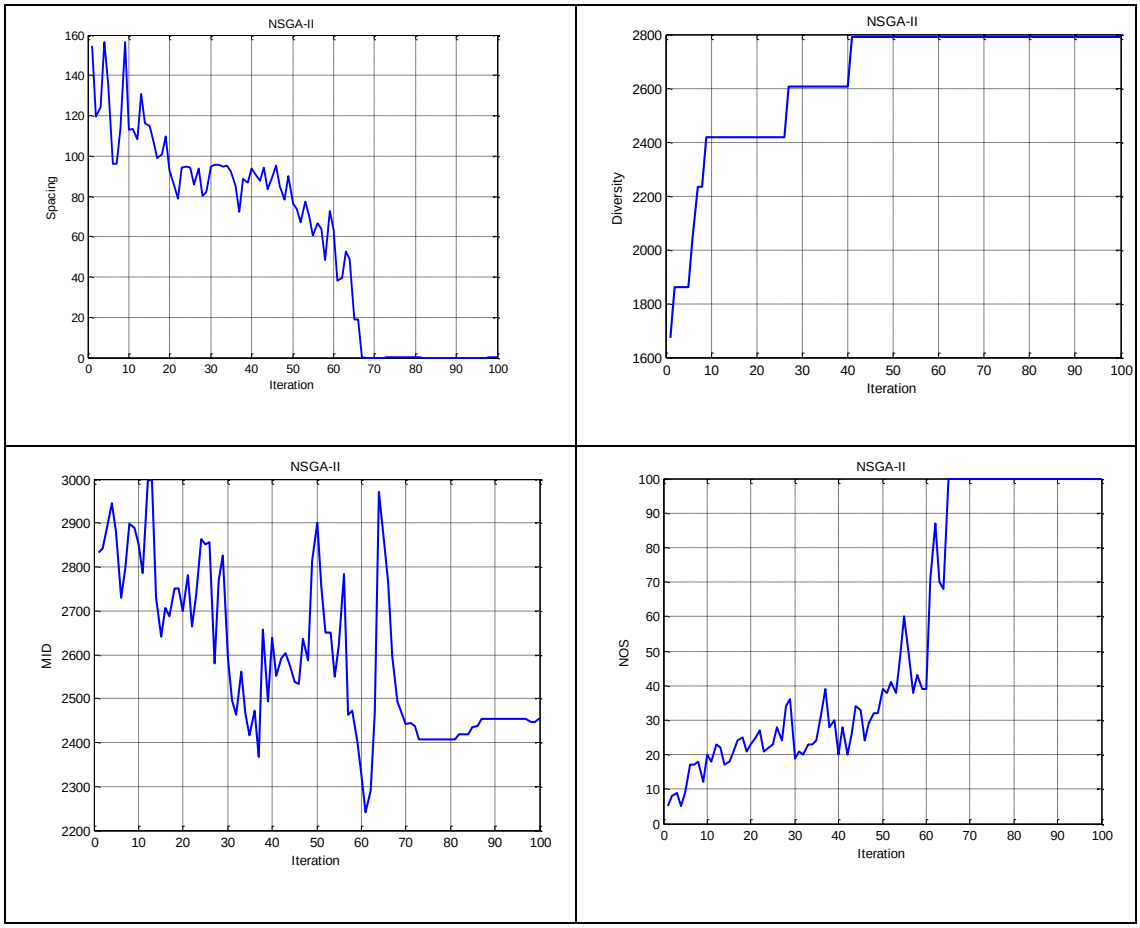

Fig 9. Performance metrics with NSGA II for the case study

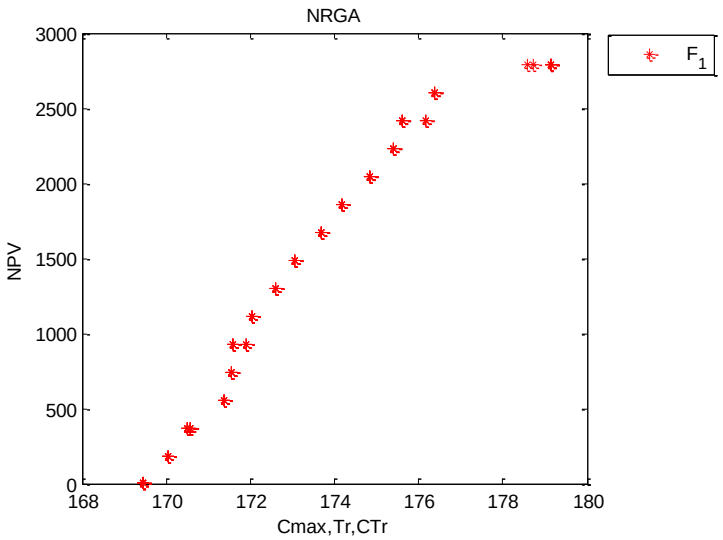

Fig 10. NRGA in the case study for objectives

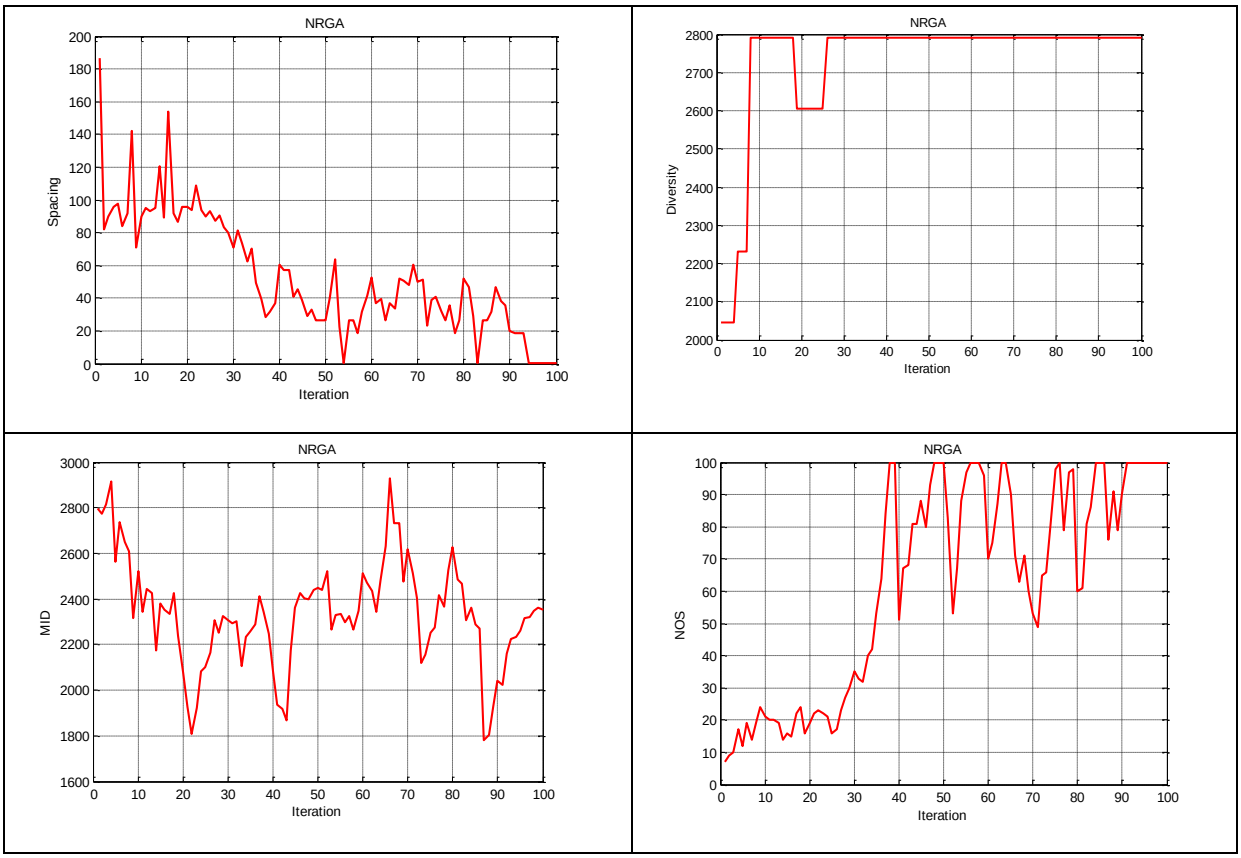

Fig 11. Performance metrics with NRGA for the case study

V. CONCLUSION

In this research, a model was developed for permutation flow shop scheduling problem considering learning and aging effects in which maintenance activity was regarded to increase the applicability of the model. The paper targeted four objectives such as minimization of the makespan and tardiness of job and tardiness cost while maximizing net present value. The importance of the first three objectives is becomes apparent by considering global competition which forces manufacturing enterprises to complete a set of tasks in lowest time and cost. The maximization of the net present value is considered as a new objective function in this field of study in order to increase applicability of the proposed approach. Due to NP-hardness of the problem, two metaheuristic approaches called NSGA II and NRGA, were proposed to solve the problem. To improve the efficiency of the methods, a tuning process based on Taguchi method was carried out to optimize the algorithms. Then both algorithms used to solve the problem 100 times. The results revealed that NSGA II is superior to NRGA. For future studies, different approaches can be introduced in order to solve the problem and obtain better Pareto solutions in less time with lower computational complexity**.**

REFERENCES

Aiello, G., La Scalia,G., Enea, M. (2013). A nondominated ranking Multi Objective Genetic Algorithm and electre method for unequal area facility layout problems. Expert Systems with Applications, 40, 4812–4819.

Alaghebandha, M., Hajipour, V. (2013). A soft computing-based approach to optimise queuing inventory control problem. International Journal of Systems Science. doi.org/10.1080/00207721.2013.809614.

Chang P.C., Chen SH., Mani V. (2009). A note on due-date assignment and single machine scheduling with a learning/aging effect. International Journal of Production Economics, 117:142–149.

Cheng, T.C.E., Lee, W.C., Wu, C.C. (2010). Single-machine scheduling with deteriorating functions for job processing times, 34:4171–4178.

Low, Ch., Lin, W.Y. (2011). Minimizing the total completion time in a single-machine scheduling problem with a learning effect. Applied Mathematical Modelling, 35, 1946–1951.

Dirk, B. (2008). A state-of-the-art review on scheduling with learning effects. European Journal of Operational Research, 188, 315–329.

Dirk, B. (1999). Single-machine scheduling with learning considerations. European Journal of Operational Research, 115, 173-178.

Eren, T, Guner, E. (2008). A bicriteria flow shop scheduling with a learning effect. Applied Mathematical Modelling, 32, 1719-1733.

Ghemawat, P. (1985). Building strategy on the experience curve – a venerable management tool remains valuable – in the right circumstances. HarvBus Rev, 63:143–149.

GawiejnowiczS. (1996). A note on scheduling on a single processor with speed dependent on a number of executed jobs. Information Processing Letters, 57, 297-300.

Hyun, C. J., Kim, Y., & Kin, Y. K. (1998). A genetic algorithm for multiple objective sequencing problems in mixed model assembly. Computers & Operations Research, 25, 675–690.

Wang, J.J., Zhang B.H . (2015). Permutation flow shop problems with bi-criterion make span and total completion time objective and position-weighted learning effects. Computers & Operations Research, 58, Pages 24–31

Jian-Jun Wang., Ya-Jing Liu. (2014). Single-machine bicriterion group scheduling with deteriorating setup times and job processing times. Applied Mathematics and Computation, 242, 309–314.

Kalyanmoy, Deb., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A Fast Elitist Multi objective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 6(2):182 – 197.

Karimi, N., Zandieh, M., Karamooz, HR. (2010). Bi-objective group scheduling in hybrid flexible flowshop: a multi-phase approach, 37:4024–4032.

Kuo, W.H., Yang D.L. (2006). Minimizing the make span in a single machine scheduling problem with a time-based learning effect. Inf Process Lett, 97:64–67.

Kuo, W.H., Yang., D.L. (2006). Minimizing the total completion time in a single-machine scheduling problem with a time dependent learning effect. European Journal of Operational Research, 174:1184–1190.

Lai, P.J., Lee, WC. (2011). Single-machine scheduling with general sum-of-processing-time-based and position-based learning effects. OMEGA The International Journal Manage S, 39:467–471.

Zadeh, L.A. (1965). Fuzzy Sets, Information and control, 8, 3, 338–353.

Lee, W.C., Wu CC. (2004). Minimizing total completion time in a two-machine flow shop with a learning effect. International Journal of Production Economics, 88:85–93.

Lee, W-C., Yeh, W-C., Chung Y-H. (2014). Total tardiness minimization in permutation flow shop with deterioration consideration, Applied Mathematical Modelling, 38, 13.

Liu, P., Zhou, X., Tang, L. (2010). Two-agent single-machine scheduling with position-dependent processing times. The International Journal of Advanced Manufacturing Technology, 48:325–331.

Moradi, H., Zandieh, M., Mahdavi, I. (2011). Non-dominated ranked genetic algorithm for a multi-objective mixed-model assembly line sequencing problem. International Journal of Production Research, 49, 12, 3479–3499.

Murata, T., Ishibuchi, H., & Tanaka, H. (1996). Multi-objective genetic algorithm and its application to flow shop

scheduling. Computers & Industrial Engineering, 30, 957–968.

Chang, P-Ch., Chen, Sh.H., Mani, V. (2009). A note on due-date assignment and single machine scheduling with a learning/aging effect. The International Journal Production Economics, 117, 142–149.

Lai, P.J., Lee, W.Ch. (2011). Single- machine scheduling with general sum-of-processing-time-based and position-based learning effects. Omega, 39, 467–471.

Rahmati, S.H.A. (2012). Proposing a Pareto-based Multi-Objective Evolutionary Algorithm to Flexible Job Shop Scheduling Problem. World Academy of Science, Engineering and Technology, 61: 1160-1165.

Reddy, V., Narendran, T. T. (2003). Heuristics for scheduling sequence dependent set-up jobs in flow line cells. International Journal of Production Research, 41(1), 193–206.

Rostami, M., Ebrahimzadeh Pilerood, A., Mahdavi Mazdeh, M. (2015). Multi-objective parallel machine scheduling problem with job deterioration and learning effect under fuzzy environment. Computers & Industrial Engineering, 85, Pages 206–215.

Rostami, M., Kheirandish, O., Ansari, N. (2015). Minimizing maximum tardiness and delivery costs with batch delivery and job release times. Appl. Math. Modelling, http://dx.doi.org/10.1016/j.apm.2015.03.052.

Rudek, A., Rudek, R. (2012). On flow shop scheduling problems with aging effect and resource allocation. International Journal of Advanced Manufacturing Technology, 62, 135-145.

Safari, J. (2011) A NSGA II for a Multi-objectives Redundancy Allocation Problem. World Academy of Science, Engineering and Technology, 78: 1319-1325.

Schott, J.R. (1995). Fault tolerant design using single and multi-criteria genetic algorithms optimization. Master's thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA.

Suh-Jenq Yang, Dar-Li Yang. (2010). Minimizing the makespan on single machine scheduling with aging effect and variable maintenance activities. Omega, 38, 528–533.

T.C.E. Cheng, Wen-Hung Kuo, Dar-Li Yang. (2013). Scheduling with a position-weighted learning effect based on sum-of-logarithm-processing-times and job position. Information Sciences, 221, 490–500.

Janiak, A., Rudek, R. (2010). Scheduling jobs under an aging\effect. Journal of the Operational Research Society, 61:1041–1048.

Vahedi-Nouri, B., Fattahi, P., Ramezanian, R. (2013). Minimizing total flow time for the non-permutation flow shop scheduling problem with learning effects and availability constraints, Journal of Manufacturing Systems, 32(1), Pages 167–173.

Wang, JB., Wang, LY., Wang, D., Huang, X., Wang, XR. (2009). A note on single-machine total completion time problem with general deteriorating function. The International Journal of Advanced Manufacturing Technology, 44:1213–1218.

Wang, JB., Wang, LY., Wang, D., Wang, XY. (2009). Single machine scheduling with a time dependent deterioration. The International Journal of Advanced Manufacturing Technology, 43:805–809.

Webb GK. (1994). Integrated circuit (IC) pricing. Journal High Technol Manag, Res, 5:247–260.

Lee, W.Ch., Yeh, W.Ch., Chung, Y.H. (2014). Total tardiness minimization in permutation flowshop with deterioration consideration. Applied Mathematical Modelling, 38, 3081–3092.

Kuo, W.H, Yang, D.L. (2006). Minimizing the makespan in a single machine scheduling problem with a time-based learning effect. Information Processing Letters, 97, 64–67.

Wu, C.C., Lee, W.C., Wang, W. C. (2007). A two-machine flow shop maximum tardiness scheduling problem with a

learning effect. International Journal of Advanced Manufacturing and Technology, 31, 743-750.

Wu, C. C., Lee, W.C. (2009) A note on the total completion time problem in a permutation flow shop with a learning effect. European Journal of Operational Research, 192, 343-347.

Wilson, A. D., King, R. E., & Hodgson, T. J. (2004). Scheduling non-similar groups on a flow line: Multiple group setups. Robotics and Computer-Integrated Manufacturing, 20, 505–515.

Yeh, W-C., Lai, P-J., Lee W-C., Chuang, M-C. (2014). Parallel-machine scheduling to minimize make span with fuzzy processing times and learning effects, Information Sciences, 269, 142–158.

Zitzler, E. (1999). Evolutionary Algorithms for Multi-objective Optimization: Methods and Applications. PhD. Thesis, Dissertation ETH No. 13398, Swiss Federal Institute of Technology (ETH), Zürich, Switzerland.

APPENDIX

TABLE A.I. Real data for the model

Number	No. of machines	No. of parts	Pij	ti	α	bik	e
1	3	6	(0.12,0.30,0.49)	(60, 70, 80)	rand[-0.9,-0.01]	rand[0.01,0.9]	(7.45, 8.00, 8.15)
2	4	7	(0.16,0.30,0.44)	(90, 100, 110)			(8.15, 8.30, 8.45)
3	5	8	(0.15,0.30,0.45)	(80, 90, 100)			(8.45, 9.00, 9.15)
4	7	8	(0.13,0.30,0.39)	(70, 80, 90)			(9.15, 9.30, 9.45)
5	10	10	(0.12,0.30,0.43)	(110, 120, 130)			(9.45, 10.00, 10.15)
6	12	10	(0.14,0.30,0.38)	(130, 140, 150)			(10.15, 10.30, 10.45)
7	12	12	(0.15,0.30,0.42)	(130, 140, 150)			(10.45, 11.00, 11.15)
8	12	15	(0.11,0.30,0.48)	(130, 140, 150)			(11.15, 11.30, 11.45)
9	14	12	(0.12,0.30,0.45)	(150, 160, 170)			(11.45, 12.00, 12.15)
10	14	14	(0.18,0.30,0.37)	(150, 160, 170)			(12.45, 12.30, 12.45)

s	(s) r	(s) r	β	λ	g	E(t)	Cdi	Cpi	Cpmax	fi	dIr
(12.45, 13.00, 13.15)	12.31	13.00	(600, 650, 700)	26	8.5	(25, 30, 35)	(1200, 1300, 1400)	(470, 520, 570)	(1500, 1600, 1700)	(150, 160, 170)	(16, 19, 21)
(12.45, 12.30, 12.45)	12.01	12.30	(600, 650, 700)	42	14	(25, 30, 35)	(1200, 1300, 1400)	(470, 520, 570)	(1500, 1600, 1700)	(150, 160, 170)	(6, 7, 8)
(11.45, 12.00, 12.15)	11.31	12.00	(750, 800, 850)	14	3	(60, 65, 70)	(1600, 1700, 1800)	(700, 750, 800)	(1800, 2000, 2200)	(130, 140, 150)	(14, 15, 16)
(11.15, 11.30, 11.45)	11.01	11.30	(750, 800, 850)	28	9	(60, 65, 70)	(1600, 1700, 1800)	(700, 750, 800)	(1800, 2000, 2200)	(130, 140, 150)	(15, 17, 19)
(10.45, 11.00, 11.15)	10.31	11.00	(750, 800, 850)	33	11	(60, 65, 70)	(1600, 1700, 1800)	(700, 750, 800)	(1800, 2000, 2200)	(130, 140, 150)	(10, 11, 12)
(10.15, 10.30, 10.45)	10.01	10.30	(500, 550, 600)	8	4	(50, 55, 60)	(1100, 1200, 1300)	(400, 450, 500)	(1300, 1400, 1500)	(110, 120, 130)	(8, 9, 10)
(9.45, 10.00, 10.15)	9.31	10.00	(450, 500, 550)	17	5	(25, 30, 35)	(970, 1070, 1170)	(320, 370, 420)	(1200, 1300, 1400)	(70, 80, 90)	(9, 9.30, 10)
(9.15, 9.30, 9.45)	9.01	9.30	(550, 650, 650)	42	14	(32, 37, 42)	(1200, 1400, 1600)	(500, 550, 600)	(1700, 1800, 1900)	(80, 90, 100)	(12, 13, 14)
(8.45, 9.00, 9.15)	8.31	9.00	(700, 750, 800)	35	7	(40, 45, 50)	(1500, 1700, 1900)	(600, 650, 700)	(1900, 2000, 2100)	(90, 100, 110)	(9, 10, 11)
(8.15, 8.30, 8.45)	8.00	8.30	(350, 400, 450)	24	8	(32, 37, 42)	(700, 800, 900)	(250, 300, 350)	(700, 800, 900)	(60, 70, 80)	(7, 8, 9)

(21, 22, 23)	(15, 16, 17)	17
(8, 9, 10)	(13, 14, 15)	17
(15, 16, 17)	(10, 11, 12)	17
(18, 19, 20)	(12, 13, 14)	17
(12, 13, 14)	(7, 8, 9)	17
(10, 11, 12)	(14, 15, 16)	17
(11, 11.30, 12)	(12, 13, 14)	17
(14, 15, 16)	(8, 9, 10)	17
(11, 12, 13)	(9, 10, 11)	17
(9, 10, 11)	(11, 12, 13)	17
d2r	**Sij**	**Φ**

TABLE A.II. Computational results for running the model with NSGA II and NRGA

NSGAII									
Objective1	174.161	173.673	170.05	173.052	172.614	176.393	172.055	175.398	
NPV	1860	1674	186	1488	1302	2604	1116	2232	
NRGA									
Objective1	174.848	174.161	173.673	170.05	173.052	172.614	175.398	176.393	
NPV	2046	1860	1674	186	1488	1302	2232	2604	
NSGAII									
Objective1	174.848	174.161	173.673	170.05	173.052	172.614	176.393	172.055	
NPV	2046	1860	1674	186	1488	1302	2604	1116	
NRGA									
Objective1	171.591	179.146	169.438	174.848	174.161	173.673	170.05	173.052	
NPV	930	2790	0	2046	1860	1674	186	1488	
NSGAII									
Objective1	169.438	174.848	174.161	173.673	170.05	173.052	172.614	176.393	
NPV	0	2046	1860	1674	186	1488	1302	2604	
NRGA									
Objective1	171.909	171.591	169.438	179.146	174.848	174.161	173.673	170.05	
NPV	930	930	0	2790	2046	1860	1674	186	
NSGAII									
Objective1	173.052	172.614	176.393	172.055	171.591	171.817	175.398	170.623	
NPV	1488	1302	2604	1116	930	930	2232	372	
NRGA									
Objective1	170.05	173.052	172.614	171.591	171.909	175.398	176.393	170.57	
NPV	186	1488	1302	930	930	2232	2604	372	
NSGAII									
Objective1	169.438	178.035	169.438	178.035	176.393	176.103	171.381	174.848	
NPV	0	2790	0	2790	2604	2418	558	2046	
NRGA									
Objective1	179.146	169.438	169.438	179.146	178.569	176.393	171.381	174.848	
NPV	2790	0	0	2790	2790	2604	558	2046	

171.381	170.623	171.539	169.446	170.501	171.817	175.615	171.591	178.035	169.438	174.848	174.161
558	372	744	0	372	930	2418	930	2790	0	2046	1860
171.381	172.055	170.57	171.539	170.501	175.615	171.591	169.438	179.146	174.848	174.161	173.673
558	1116	372	744	372	2418	930	0	2790	2046	1860	1674
175.398	171.381	170.623	171.539	169.446	170.501	171.817	175.615	171.591	178.035	169.438	174.848
2232	558	372	744	0	372	930	2418	930	2790	0	2046
172.614	175.398	176.393	171.381	172.055	170.57	171.539	170.501	175.615	171.591	179.146	169.438
1302	2232	2604	558	1116	372	744	372	2418	930	2790	0
172.055	175.398	171.381	170.623	171.539	169.446	170.501	171.817	175.615	171.591	178.035	169.438
1116	2232	558	372	744	0	372	930	2418	930	2790	0
173.052	172.614	175.398	176.393	171.381	172.055	170.57	171.539	170.501	175.615	176.178	171.909
1488	1302	2232	2604	558	1116	372	744	372	2418	2418	930
171.539	171.381	170.623	170.501	171.539	169.446	169.446	170.501	171.817	175.615	171.591	178.035
744	558	372	372	744	0	0	372	930	2418	930	2790
171.539	171.381	172.055	170.57	170.501	171.539	169.452	169.452	178.723	170.501	175.615	176.178
744	558	1116	372	372	744	0	0	2790	372	2418	2418
174.161	173.673	173.052	170.05	172.614	172.055	175.398	174.848	174.161	173.673	175.615	170.05
1860	1674	1488	186	1302	1116	2232	2046	1860	1674	2418	186
174.161	173.673	173.052	170.05	175.615	176.178	172.614	172.055	175.398	174.848	174.161	173.673
1860	1674	1488	186	2418	2418	1302	1116	2232	2046	1860	1674

TABLE A.III. Paired T for Objective NSGAII - Objective NRGA

	N	Mean	StDev	SE Mean
Objective NSGAII	100	172.837	2.459	0.246
Objective NRGA	100	173.218	2.689	0.269
Difference	100	-0.381	3.372	0.337

T-Value = -1.13 P-Value = 0.131

TABLE A.IV. Paired T for NPV NSGAII - NPV NRGA

	N	Mean	StDev	SE Mean
NPV NSGAII	100	1272.2	892.9	89.3
NPV NRGA	100	1374.5	900.0	90.0
Difference	100	-102	1176	118

T-Value = -0.87 P-Value = 0.193