International Journal of Industrial Engineering & Production Research

March 2014, Volume 25, Number 1
pp. 1-12

pISSN: 2008-4889

http://IJIEPR.iust.ac.ir/

Multi-objective scheduling problem in a three-stage production system

P. Fattahi, S.M.H. Hosseini, F. Jolai* & A.D. Safi samghabadi

Parviz Fattahi, *, Department of Industrial Engineering, Bu-Ali Sina University, Hamedan, Iran, fattahi@basu.ac.ir*
Seyed Mohammad Hassan Hosseini, *Department of Industrial Engineering, University of Shahrood, Shahrood, Iran, sh.hosseini51@gmail.com,*
Fariborz Jolai, *Department of Industrial Engineering, University of Tehran, Tehran, Iran, fjolai@ut.ac.ir,*
Azam dokht Safi Samghabadi, *Department of Industrial Engineering, Payame Noor University, Tehran, Iran, safi@tehran.pnu.ac.ir,*

KEYWORDS

Scheduling, Multi-objective,
Three-stage production system,
Assembly,
Genetic algorithm

ABSTRACT

A three-stage production system is considered in this paper. There are two stages to fabricate and provide the parts and an assembly stage to assemble the parts and complete the products in this system. Suppose that a number of products of different kinds are ordered. Each product is assembled with a set of several parts. At first the parts are produced in the first stage with parallel machines and then they are controlled and provided in the second stage and finally the parts are assembled in an assembly stage to complete the products. Two objective functions are considered: (1) minimizing the completion time of all products (makespan), and (2) minimizing the sum of earliness and tardiness of all products ($\sum_i (E_i / T_i)$). Since this type of problem is NP-hard, a new multi-objective algorithm is designed for searching local Pareto-optimal frontier for the problem. To validate the performance of the proposed algorithm, various test problems are designed and the reliability of the proposed algorithm, based on some comparison metrics, is compared with two prominent multi-objective genetic algorithms, i.e. NSGA-II and SPEA-II. The computational results show that the performance of the proposed algorithms is good in both efficiency and effectiveness criteria.

1. Introduction

Scheduling problem plays a key role in a decision making and production planning system. Hence, there has been an increasing interest in solving scheduling problem during the recent half century. Nevertheless, most of research in production scheduling is concerned with the minimization of a single criterion. Up to the 1980s, scheduling research was mainly concentrated on optimizing single performance measures such as makespan (C_{max}), total flow time (F), maximum tardiness (T_{max}), total tardiness (T) and number of tardy jobs (n_T) [1]. C_{max} and F are related to maximizing system utilization and minimizing work-in-process inventories respectively, while the remaining measures relate to job due dates. However, scheduling problems often involve more than one aspect and therefore require multiple criteria analysis. In every productive or service organization, each particular department decision maker wants to minimize a special criterion. For example in a company, the commercial manager is interested in satisfying customers and then minimizing the tardiness. On the other hand, the production manager wishes to optimize the use of the machine by minimizing the makespan or the work in process by minimizing the maximum flow time. Each of these objectives is valid from a general point of view. Since these objectives are conflicting, a solution may perform well for one objective, but giving bad results for others. For this reason, scheduling problems have often a multi-objective nature [2]. An assembly type production system with multi-objective function is studied in this paper. There are two stages to fabricate and provide the parts and an assembly stage to join the parts into the products. Assembly type production system that has many applications in industry, has received an increasing attention of researchers recently [3, 4, 5, and 6]. For example Lee et al. [3] described an application in a fire engine assembly plant while Potts et al. [7] described an application in personal computer manufacturing. In particular, manufacturing of almost

* **Corresponding author: Fariborz Jolai**
Email: *fjolai@ut.ac.ir*
Paper first received March 13, 2013, and in accepted form Oct. 26, 2013.

all items may be modeled as a two-stage assembly scheduling problem including machining operations and assembly operations [4]. Despite the importance of this problem, the review studied shows that scant attention has been given to solve it, especially in the case of multiple criteria.

The first study in assembly-type flow shop scheduling problem was done by Lee et al. in 1993. They studied a two stage assembly flow shop scheduling problem considering a single objective function (C_{max}). They show that the problem is strongly NP-complete and identified several special cases of the problem that can be solved in polynomial time and suggested a branch and bound solution and also three heuristics. After that Potts et al. [7] studied and extend the problem with the same objective function as [3]. Hariri and Potts [8] also studied the same problem as [7] with the same objective function and proposed a branch and bound algorithm. Cheng and Wang [9] consider minimizing the makespan in the two-machine flow shop scheduling with a special structure and develop several properties of an optimal solution and obtain optimal schedules for some special cases. In most studies of assembly type scheduling problem, it is assumed that the preassembly stage has a parallel machines format followed by an assembly stage. This production system is named two-stage assembly flow shop. The objective function of these studies is the single criterion such as the completion time of all jobs. For example see Koulamas and Kyparisis [10], Sung and Kim [11], and Allahverdi and Al-Anzi [4]. In all of these studies it is shown that the problem is NP-Hard, and hence most of them present some approximately solution based on metaheuristic algorithms. Some studies have been done by Yokoyama [12, and 13] and Yokoyama et al. [14] in the assembly type production system in which that the reassembly stage is a two or three stage flow shop. The objective function is still the single criterion such asmakespan, mean completion time for all products, and weighted sum of completion time of each product. This problem is also NP-Hard and they present some heuristic solution or B&B algorithm for the special cases of the problem. Fattahi et al. [15] introduce the hybrid flow shop with assembly operation for the first time in 2012.

Their objective function is makespan and they considered the preassembly stage as a two stage hybrid flow shop and present a mathematical model for the considered problem. Since the considered problem is strongly NP-Hard, they present some heuristic solution that can solve the problem up to 150 products and 16 parts for each product. The majority of papers on these problems have concentrated on single-objective or criterion problems, while consideration of multiple objectives or criteria is more realistic.

According the literature survey, there has been an increasing interest in multi-objective scheduling problem (MOSP). Hence, there has been a noticeable increase in published the MOSP especially multi-objective evolutionary algorithms (MOEA). Konak et al. [16] and Sun et al. [17] presented a review and prospects of multi-objective optimization algorithms. Also Coello et al [18] presented a comprehensive evolutionary algorithm for solving multi-objective problems. According these studies, being a population-based approach, genetic algorithms (GA) are well suited to solve multi-objective optimization problems. Therefore, GA has been the most popular heuristic approach to multi-objective design and optimization problems. The first multi-objective GA, called vector evaluated GA (or VEGA), was proposed by Schaffer (1985).

Afterwards, several multi-objective evolutionary algorithms were developed including Multi-objective Genetic Algorithm (MOGA), Niched Pareto Genetic Algorithm (NPGA), Weight-based Genetic Algorithm (WBGA), Random Weighted Genetic Algorithm (RWGA), Nondominated Sorting Genetic Algorithm (NSGA), Strength Pareto Evolutionary Algorithm (SPEA), improved SPEA (SPEA2), Pareto-Archived Evolution Strategy (PAES), Pareto Envelope-based Selection Algorithm (PESA), Region-based Selection in Evolutionary Multiobjective Optimization (PESA-II), Fast Nondominated Sorting Genetic Algorithm (NSGA-II), Multi-objective Evolutionary Algorithm (MEA), Micro-GA, Rank-Density Based Genetic Algorithm (RDGA), and Dynamic Multi-objective Evolutionary Algorithm (DMOEA) [16, and 18].

The rest of this paper is organized as follows: In section 2, the problem is described completely. The proposed solving algorithm is presented in section 3. In section 4 design of the problems and computational experiment and results is presented. Finally, a Concluding remarks and summary of the work and direction for the future research are given in section 5.

2. Problem Description

A three-stage production system with multi-objective criteria is considered in this paper. This system contains two stages for fabrication and providing the parts that followed by an assembly stage. Suppose that several products of different kinds (H) are ordered and each product needs a set of parts ($J_h = 1, 2, 3, \ldots, n_h$) to complete. At first, the parts are manufactured in the first two stages (can be considered as a hybrid flow shop). Each part *j* has a certain operation time P_{lj} on stage *l l* ($l = 1, 2$). The first stage contains some identical parallel machines to fabricate the parts and the number of machines is *m*. The second stage control and provide the fabricated parts. After manufacturing the parts, they are assembled into the products on an assembly stage. The assembly operation cannot be started for a product until the set of parts are completed in machining operations. The considered objective is to minimize makespan and the sum of earliness and tardiness (C_{max}, $\sum_h E_h/T_h$). Decision variables are sequence of the products to be assembled and also sequence of the parts in the first stage.

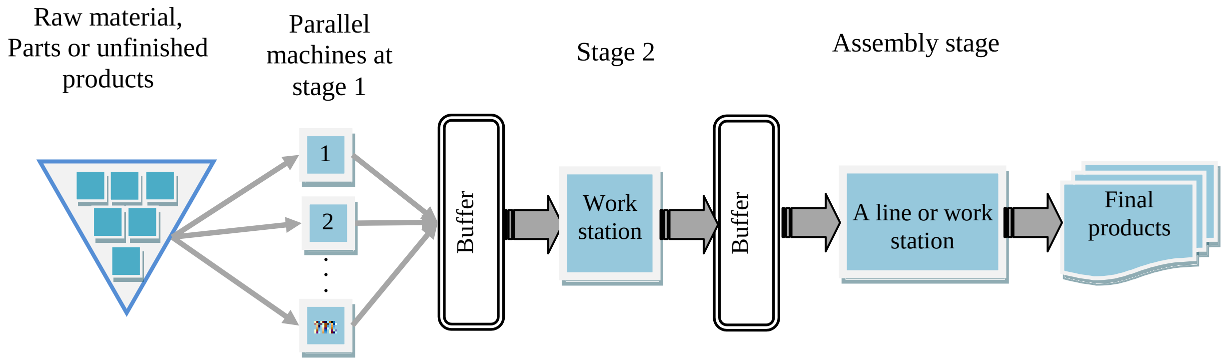

Fig. 1. A schematic view of the considered problem

The considered problem in this paper has many applications in manufacturing industries. Figure 1 shows a schematic view of the considered problem. The inputs contain raw material, parts or unfinished products that are processed by the parallel machines in the first stage. Then in the second stage some activity such as control and reading the parts are done. When the set of parts of a product complete, they joined on assembly stage. Typically, buffers are located between stages to store intermediate products and it is supposed that there is no limited in buffer storages.

2-1. Notations

We introduce the following notation for this problem:

H Total number of products

h Product index (h = 1, 2, . . . , H)

n Total number of parts

j Part index (j = 1, 2, . . . , n)

n_h Total number of parts of product h (h = 1, 2, . . ., H)

l Stage index of the preassembly stage (l=1,2)

P_{lj} Processing time for part j in stage l (l=1, 2)

m Number of parallel machines in stage 1

k Machine index of stage 1 (k=*1, 2, …, m*),

A_h Assembly time of product h

d_h Due date for delivery of product h

M A very big and positive amount

Also variables of the mathematical model are as follow:

x_{ijkl} 1, if job j is processed directly after job i on machine k in stage l, 0 otherwise,

x_{0ikl} 1, if job i is the first job on machine k in stage l, 0 otherwise,

x_{i0kl} 1, if job i is the last job on machine k in stage l, 0 otherwise,

$C_j^{(l)}$ Completion time of job j in stage l,

F_h Finish time of the parts for h th product and ready to assemble

$S_{h'h}$ 0, if all parts of product h' is provided to assemble before the parts of product h, a positive amount otherwise,

C_h Completion time of assembly the product h

E_h Earliness of Completion time of the product h

T_h Tardiness of Completion time of the product h

D Sum of earliness and tardiness of products ($D = \sum_{h=1}^{H}(E_h/T_h)$)

The problem is to decide about sequencing of the products and their parts, and the objective function of the considered problem is expressed as:

Min $Z = \{max(C_h), \sum_{h=1}^{H}(E_h/T_h)\}$ or $\{C_{max}, D\}$

While the parameter D is computed as (1)

$$D = \left\{\sum_{h=1}^{H}[max(0, C_h - d_h) + max(0, d_h - C_h)]\right\} \quad (1)$$

2-2. Assumptions

(1) All parts are available at time zero.

(2) The parallel machines in stage 1 are uniform.

(3) If product h is going to be assembled before product h', then, on each stage, processing of any part of product h' doesn't start before starting the processing of all parts for product h.

(4) Assembly operation for a product will not start until all parts of its product are completed.

(5) When assembly operation of a product is started, it doesn't stop until completed (no preemption in assembly stage)

(6) There is no limited in buffer storages

2-3. Mathematical Modeling

The mathematical definition of a multi-objective problem (MOP) is important in providing a foundation of understanding between the interdisciplinary nature of deriving possible solution techniques (deterministic, stochastic); i.e., search algorithms. Fattahi et al. [15] presented a mathematical model for the assembly flexible flow shop scheduling problem with a single objective. Their model is developed for the considered in this study. The single objective formulation is extended to reflect the nature of multi-objective problems where there is not one objective function to optimize, but many. Thus, there is not one unique solution but a set of solutions. This set of solutions is found through the use of Pareto Optimality Theory [18]. Note that multi-objective problems require a decision maker to make a choice of $\mathbf{x}^*$ values. The selection is essentially a tradeoff of one complete solution x over another in multi-objective space.

Based on the present problem and notations, the mathematical formulation of the problem is presented as follows:

$$\text{Min } Z_1 = C_{max} \qquad (2)$$

$$\text{Min } Z_2 = D \qquad (3)$$

Subject to:

$$\sum_{i=0,i\neq j}^{n}\sum_{k=1}^{K_l} x_{ijkl} = 1 \qquad j=1,2,3,\dots,n \quad K_1 = m \qquad l=1,2 \quad K_2 = 1 \qquad (4)$$

$$\sum_{j=0}^{n} x_{ijlk_l} \le 1 \qquad i=0,1,2,3,\dots,n \quad k_1 = 1,2,\dots,m \qquad l=1,2 \quad k_2 = 1 \qquad (5)$$

$$\sum_{i=0,i\neq z}^{n} x_{izlk_l} - \sum_{j=0,j\neq z}^{n} x_{zjlk_l} = 0 \qquad z=1,2,3,\dots,n \quad k_1 = 1,2,\dots,m \qquad l=1,2 \quad k_2 = 1 \qquad (6)$$

$$C_i^{(l)} + \sum_{k=1}^{K_l} x_{ijkl} \times P_{lj} + \left(\sum_{k=1}^{K_l} x_{ijkl} - 1\right) \times M \le C_j^{(l)} \qquad i=1,2,3,\dots,n \quad j=1,2,3,\dots,n \quad l=1,2 \quad k_1 = 1,2,\dots,m \quad k_2 = 1 \qquad (7)$$

$$C_j^{(1)} + P_{2j} \le C_j^{(2)} \qquad j=1,2,3,\dots,n \qquad (8)$$

$$C_j^{(2)} \le F_h \qquad \forall j \in \{J_h\}, \quad h=2,3,4,\dots,H \qquad (9)$$

$$F_h + A_h \le C_h \qquad h=1,2,3,\dots,H \qquad (10)$$

$$C_{h'} + A_h - S_{h'h} \times M \le C_h \qquad h, h'=1,2,3,\dots,H \qquad (11)$$

$$S_{h'h} = max\{0, (F_{h'} - F_h)\} \qquad h, h' = 1,2,3,\dots,H, \quad h' \neq h \qquad (12)$$

$$C_h \le C_{max} \qquad h=1,2,3,\dots,H \qquad (13)$$

$$E_h \ge max\{0, (d_h - C_h)\} \qquad h=1,2,3,\dots,H \qquad (14)$$

$$T_h \ge max\{0, (C_h - d_h)\} \qquad h=1,2,3,\dots,H \qquad (15)$$

$$D = \sum_{h=1}^{H} [E_h + T_h] \qquad h=1,2,3,\dots,H \qquad (16)$$

$$x_{ijkl} \in \{0,1\} \qquad i=1,2,3,\dots,n, \quad j=1,2,3,\dots,n, \quad l=1,2, \quad k \le 1,2,\dots,K_l \qquad (17)$$

$$C_j^{(l)} \ge 0 \qquad j=1,2,3,\dots,n \quad l=1,2 \qquad (18)$$

$$E_h \ge 0 \qquad h=1,2,3,\dots,H \qquad (19)$$

$$T_h \ge 0 \qquad h=1,2,3,\dots,H \qquad (20)$$

Equations (2) and (3) determine the objective functions of the given problem that are minimizing the maximum completion time (makespan) and the sum of earliness and tardiness, respectively. Constraints (4), (5) and (6) ensure that each part is processed precisely once at each stage. In particular, constraint (4) guarantees that at each stage l for each part j there is a unique machine such that either j is processed first or after another part on that machine. The inequalities (5) imply that at each stage there is a machine on which a part has a successor or is processed last. Finally, at each stage for each part there is one and only one machine satisfying both of the previous two conditions by (6).

Constraints (7) and (8) take care of the completion times of the parts on stage 1, 2. Inequalities (7) ensure that the completion times C_i^l and C_j^l of parts i and j scheduled consecutively on the same machine respect this order. Inequality (8) implies that the parts go through the stages in the right order, i.e. from stage 1 to stage 2. Inequalities (9) take care of the start times of the products at the assembly stage. The inequalities (10), (11), and (12) express the completion time of products. Inequalities (11) and (12) ensure that the completion time of product h and h' scheduled consecutively on the assembly stage respect this order.

The constraint that the makespan is not smaller than the completion time of any product is expressed by constraints (13). Calculating the earliness and tardiness is shown in equations (14), and (15), respectively and the equation (16) presents the sum of earliness and tardiness for all products. The last four constraints specify the domains of the decision variables.

3. The Proposed Solving Algorithm

Multi-objective optimization was originally conceived with finding Pareto-optimal solutions. Such solutions are non-dominated, i.e., no other solution is superior to them when all objectives are taken into account. Since in GA a population-based approach is used, it is well suited to solve multi-objective optimization problems, and hence several multi-objective evolutionary algorithms were developed after presentation the first multi-objective GA by Schaffer [18]. Therefore in this section a new solving algorithm is proposed for the considered problem based on GA is proposed.

3.1. Solution Representation

Implementation a meta-heuristic needs to decide how to represent and relate solutions in an efficient way to the searching space. Representation should be easy to decode and calculated to reduce the run time of the algorithm. In the considered problem, several products (H) of different kinds are ordered to be scheduled and produced. Each product needs a set of parts ($J_h = 1, 2, 3, \dots, n_h$) to complete and the parts are fabricated in a hybrid flow shop.

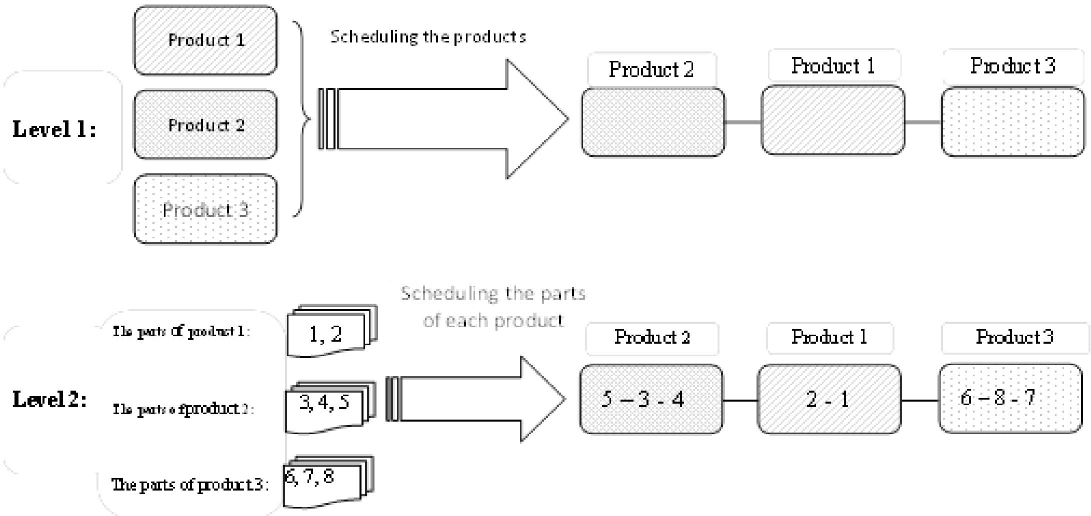

Fig. 2. An example of two level scheduling

According the assumption (3), If product *h* is going to be assembled before product *h*', then, process operation of all parts of the product *h*' doesn't start before processing of all parts of the product *h*. Hence in the proposed algorithm the products and the parts are scheduled in two levels separately. For example consider an example that three products must to be produced. Product 1 needs two parts, and both products 2 and 3 need three parts to complete. Two levels of an example scheduling of this problem can be done as figure 2.

In order to coding the solutions as chromosomes in GA proposed algorithm, each solution (sequence) is considered in a two-row matrix that the above row shows the number of products and the below indicates the parts of the above product. For example the sequence of the figure 2 is shown in figure 3.

Product number:	2	2	2	1	1	3	3	3
Part number:	5	3	4	2	1	6	7	8

Fig. 3. Presentation the products and their parts as a chromosome

The schedule of the parts in each product is considered only on stage 1. After that, each part that release earlier from stage 1, will process sooner on stage 2.

The chromosome and probability of crossover and mutation is defined as below:

Chromosome: Each sequence of all products including sequence of their parts (similar to fig. 3).

P_{CPr}: Probability of crossover operation on products in each chromosome.

P_{CPa} : Probability of crossover operation on the parts of each product in every chromosome.

P_{MPr}: Probability of mutation operation on products in each chromosome.

P_{MPa} : Probability of mutation operation on the parts of each product in every chromosome.

Crossover and mutation operation is implemented on product and their parts in each chromosome separately.

3-2. Initialization

Implementation of several experiments showed that population with a variable size of N presents the optimum results. Hence, the experiments showed that the population size is better to be depended on the number of products H and must be at least 20. Also, in order to reduce the run time of algorithm, it is better to neglect doing the crossover on the parts and so only the mutation is done on sequencing the parts. The stopping criterion is the number of iterations and is considered as a variable and is determined as (21).

$$ITER = \max(50\ , 1.5 \times H) \tag{21}$$

As for the result of the experiments the parameters are determined as below:

$N = \max(40\ , H)$

$P_{CPr} = 0.9$

$P_{CPa} = 0$

$P_{MPr} = 0.2$

$P_{MPa} = 0.1$

$ITER = \max(50\ , 1.5 \times H)$

3-3. Generation the Initial Population P_0

Most evolutionary algorithms use a random procedure to generate an initial set of solutions. However, since the output results are strongly responsive to the initial set, it is better that some of the initial solution is identified as suitable rules. Hence, in initial population three solutions are determined in regulative as below and the others generate randomness.

- One solution is determined based on the earliest due date (EDD) of the product.
- The second solution is determined according non-increasing in assembly time.
- The third is determined according non-decreasing in assembly time.

After generation the initial sequencing for the products, all of the parts are scheduled randomly.

3-4. Calculation the Fitness Value and Sorting the Solutions

In order to calculate the fitness value of each solution, the dominance ranking is used. That is, for each solution it is important that how many individuals is an individual dominated by (plus 1).

As an example of dominance ranking see figure 4. The green circles show the non-dominated solutions that are dominated by no other solutions. The blue triangles show the solutions that are dominated by only one solution and their rank will be 2, and so on:

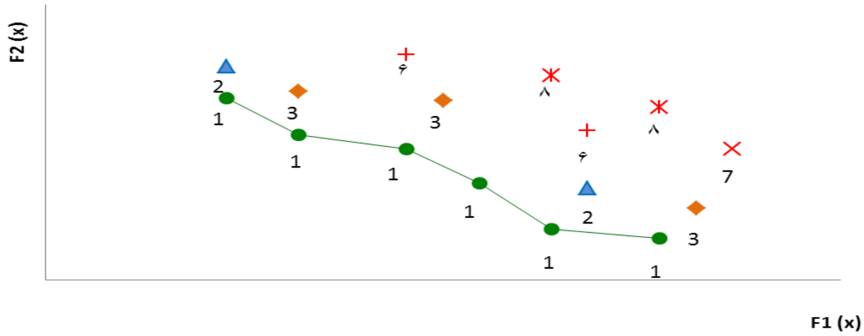

Fig. 4. The Dominance ranking for sorting

3-5. Selection the Parents

The parents are selected based on the roulette-wheel rule. After calculation the fitness value ($fv(x)$) For each solution, the operation of the roulette-wheel is done to select the parents.

3-6. Recombination and Mutation

There exist a variety of crossover operators for recombination that are suitable for the scheduling problems. We tested some of them and finally two operators that were selected for the proposed algorithm are: one-point crossover (1PX), and two-point crossover (2PX). The results showed that the crossover operation on the parts doesn't any significant improve on objective functions, therefore in order to increase the efficiency of the algorithm; the crossover is done only on the products. The mutation operator used here is the insertion operator, which randomly selects a product or a part in the sequence and inserts it in a random position of the sequence.

3-7. Eliminating Solutions as the Rule of Tabu Radius (TR)

In order to provide the diversity of solutions and prevention of creation the clusters of solutions, a tabu space is defined around any non-dominated solution. That is, a circle with a radius of R is considered for each non-dominated solution. After crossover and mutation operations, the offspring that are not non-dominated solution and are placed in a circle around one non-dominated solution with a radius of R will eliminated and then the new population will be selected from the remained solutions. Figure 5 and 6 show an example of this rule.

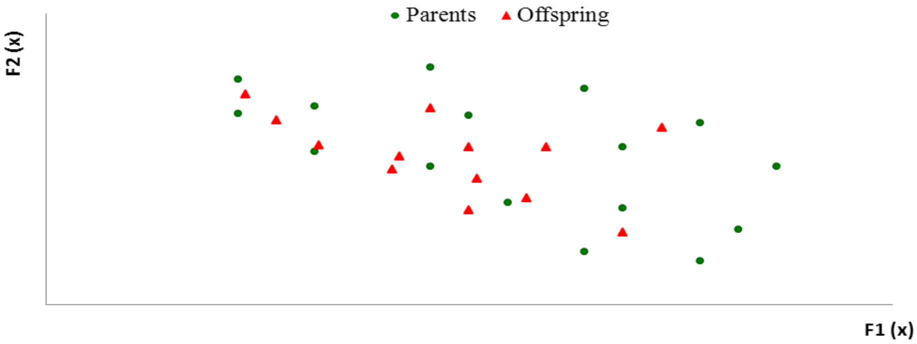

Fig. 5. an example of generation offspring

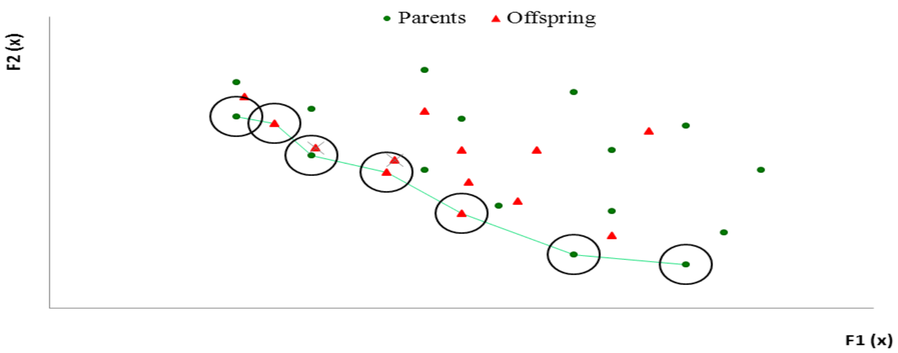

Fig. 6. an example of the rule of Tabu radius (TR)

3-8. Selection the New Generation P_{t+1}

The new generation P_{t+1} from the parents and offspring $(P_t \cup Q_t)$ is selected based on the fitness values that were illustrated in section 3.4. When more than N population members of the combined population exist in the non-dominated set or the last dominance rank, only those that are maximally apart from their neighbors in the same rank are chosen.

The proposed algorithm that uses the Tabu radius and genetic algorithm is called as TRGA in this article and the flowchart of the proposed algorithm is presented in figure 7.

4. Computational Experiments and Results

In this section, the computational experiment is carried out in order to evaluate the performance of the proposed algorithm. The tests have been performed on various condition of the problem.

4-1. Design of Problems

To show the efficiency of the proposed algorithm, it is necessary to design problems in a variety wide of conditions and test the proposed algorithm by them.

In the scheduling problems that the earliness and tardiness are considered in the objective function, the problems are designed in a variety wide of due date. Hence the researchers have considered two significant factors consisting the tardiness (τ) and the range of due date (R) in these problems [20]. Generally, by considering these two factors, the due dates can be obtained as (22):

$$d = \left[\left(1-\tau-\frac{R}{2}\right)\times M, \left(1-\tau+\frac{R}{2}\right)\times M\right] \qquad (22)$$

Researchers usually design the problems by changing the factors τ and R. Moslehi et al. [20], present that when $\tau = 0.2$ and $R = 0.6$, the primary jobs of the sequence have earliness, and the remaining ones often have tardiness. This combination is considered in this study and so due date of the product is defined within a discrete uniform distribution with range [0.5M, 1.1M]. M is the maximum completion times of all jobs that usually is obtained from an exist algorithm. We use the GRASP algorithm to obtain M. The testing data is divided into the small problems, the mediocre problems, and the large problems by changing the parameters. The following parameters are considered to design and generate these problems totally:

- H: 10, 15, 25, 50, 100, and 200.
- n_h:generated from the discrete uniform distribution with range [2 , 7].
- m: 2, 3, and 4.
- P_{1j}:generated from the discrete uniform distribution with range [25 , 75].
- P_{2j}:generated from the discrete uniform distribution with range [15 , 20].
- A_h:generated from the discrete uniform distribution with range [50 , 100].
- d_h:generated from the discrete uniform distribution with range [0.5M, 1.1M].

By a combination of all parameters, the problems and their data are defined as table 1.

4-2. Comparisons of Results

Evaluation the result of the proposed algorithm is done in this section. Each problem has been run ten times and the best and the average of results are evaluated. The performance of the proposed tabu radius and genetic algorithm (TRGA) is compared with two well-known multi-objective genetic algorithms NSGA-II and SPEA-II. These algorithms have been coded in the MATLAB 7/10/0/499 (R2010a) and executed on a Pc with a 2.0GHz Intel Core 2 Duo processor and 1GB of RAM memory. First, a brief discussion is presented on the implementation of these algorithms.

4-2-1. Two Well-Known Multi-Objective Genetic Algorithms

Non-dominated sorting genetic algorithm II (NSGA-II). Deb et al. [21] suggested an elitist multi-objective genetic algorithm in which the parent and offspring population are combined together and evaluated using: (i) a fast non-dominated sorting method; (ii) an elitist approach; and (iii) an efficient crowded-comparison mechanism. When more than population sizes of the combined population exist in the non-dominated set, only those that are maximally apart from their neighbors according to the crowding distance are chosen. *Strength Pareto evolutionary algorithm II (SPEA-II).* Zitzler et al. [22] proposed a Pareto-based method, the strength Pareto evolutionary algorithm II

(SPEA-II), which is an intelligent enhanced version of SPEA. In SPEA-II, each individual in both the main population and the elitist non-dominated archive is assigned a strength value, which incorporates both dominance and density information. On the basis of the strength value, the final rank value is determined by the summation of the strengths of the points that dominate the current individual. Meanwhile, a density estimation method is applied to obtain the density value of each individual. The final fitness is the sum of the rank and density values. Additionally, a truncation method is used to maintain a constant number of individuals in the Pareto archive.

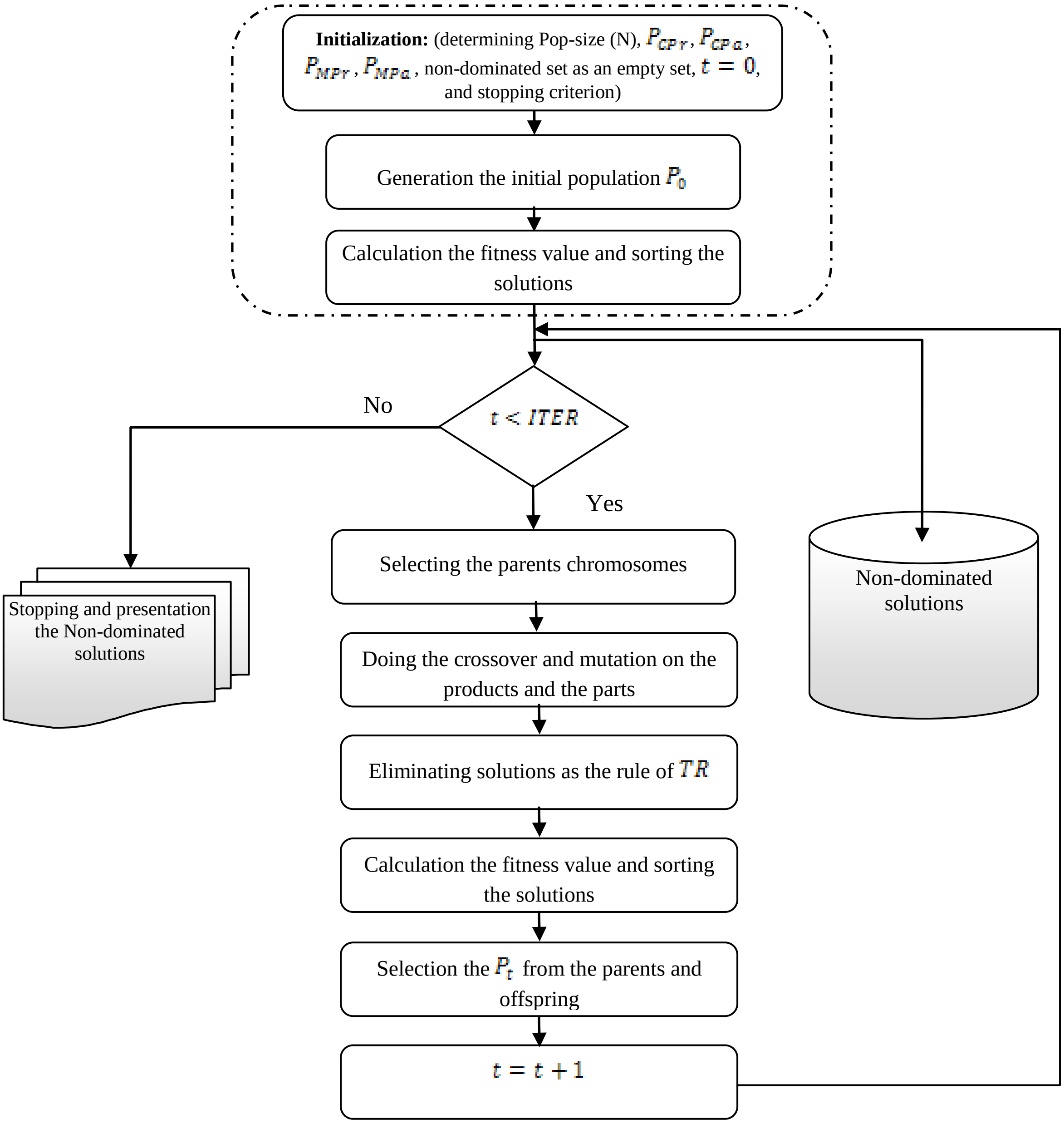

Fig. 7. The scheme of the proposed algorithm (TRGA)

4-2-2. Small-Sized Problems

At the first, experiment is carried out on the small-sized problems. The proposed algorithm is applied to the small-sized problems and its performance is evaluated. In this evaluation the Pareto-optimal is needed that is obtained from the mathematical model. There are a number of methods available to compare the performance of different algorithms. Rahimi-Vahed et al. [23, and 24] and many other researchers use the number of Pareto solutions as a quantitative measure of the performance of the algorithms studied.

The number of Pareto solutions, Overall Non-dominated Vector Generation (ONVG), the Overall Non-dominated Vector Generation Ratio (ONVGR), the error ratio (ER), and the generational distance (GD) are also used as the performance measure indicators when the Pareto-optimal solutions are known [18]. The comparison metrics that we implemented are explained in the next section.

Tab. 1. The test problems

Problem size	Problem name	Number of products	Number of parts	m	P_{1j}	P_{2j}	A_h
Small	HS-I	10	[2 , 7]	2	[25 , 75]	[15 , 20]	[50 , 100]
	HS-II	10	[2 , 7]	3	[25 , 75]	[15 , 20]	[50 , 100]
	HS-III	10	[2 , 7]	4	[25 , 75]	[15 , 20]	[50 , 100]
	HS-IV	15	[2 , 7]	2	[25 , 75]	[15 , 20]	[50 , 100]
	HS-V	15	[2 , 7]	3	[25 , 75]	[15 , 20]	[50 , 100]
	HS-VI	15	[2 , 7]	4	[25 , 75]	[15 , 20]	[50 , 100]
Mediocre	HM-I	25	[2 , 7]	2	[25 , 75]	[15 , 20]	[50 , 100]
	HM-II	25	[2 , 7]	3	[25 , 75]	[15 , 20]	[50 , 100]
	HM-III	25	[2 , 7]	4	[25 , 75]	[15 , 20]	[50 , 100]
	HM-IV	50	[2 , 7]	2	[25 , 75]	[15 , 20]	[50 , 100]
	HM-V	50	[2 , 7]	3	[25 , 75]	[15 , 20]	[50 , 100]
	HM-VI	50	[2 , 7]	4	[25 , 75]	[15 , 20]	[50 , 100]
Large	HL-I	100	[2 , 7]	2	[25 , 75]	[15 , 20]	[50 , 100]
	HL-II	100	[2 , 7]	3	[25 , 75]	[15 , 20]	[50 , 100]
	HL-III	100	[2 , 7]	4	[25 , 75]	[15 , 20]	[50 , 100]
	HL-IV	200	[2 , 7]	2	[25 , 75]	[15 , 20]	[50 , 100]
	HL-V	200	[2 , 7]	3	[25 , 75]	[15 , 20]	[50 , 100]
	HL-VI	200	[2 , 7]	4	[25 , 75]	[15 , 20]	[50 , 100]

4-2-2-1. The Number of Pareto Solutions. This metric shows the number of Pareto-optimal solutions that each algorithm can find. The number of Pareto-optimal solutions corresponding to each algorithm is compared with the total Pareto-optimal solutions found by exhaustive enumeration.

4-2-2-2. Error Ratio (ER). After finishing the solving process, the number of solutions on the final Pareto-front (PF_{known}) is termed as $|PF_{known}|$ and the number of solutions on the optimum Pareto-front (PF_{true}) is termed as $|PF_{true}|$. The Error Ratio (ER) metric reports the number of solutions on the final Pareto-front ($|PF_{known}|$) that are not members of the optimum Pareto-front ($|PF_{true}|$) [23]. This metric which is Pareto compliant, requires that PF_{true} is known and that the proposed algorithm approaches the Pareto-front. In this study the lingo is used to obtained the PF_{true} for the small problems according the proposed mathematical modeling and varying the w_i. After determining the PF_{known} by the proposed algorithm and the PF_{true} by lingo, ER is calculated as (23).

$$ER = \frac{\sum_{i=1}^{|PF_{known}|} e_i}{|PF_{known}|} \tag{23}$$

Where e_i is one if the i^{th} vector of PF_{known} is not an element of PF_{true}. When $ER = 1$, this indicates that none of the points in PF_{known} are in PF_{true}, that is none solutions outcome from the proposed algorithm is positioned on the optimum Pareto-front. On the other hand, when $ER = 0$, the PF_{known} is the same as PF_{true}.

4-2-2-3. Generational Distance (GD). The Generational Distance (GD) reports how far, on average, PF_{known} is from PF_{true}. This indicator is mathematically defined as equation (24).

$$GD = \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{|\boldsymbol{PF_{known}}|} \tag{24}$$

Where $|PF_{known}|$ is the number of vectors in PF_{known}, and d_i is the Euclidean phenotypic distance between each member, i, of $\boldsymbol{PF_{known}}$ and the closest member in $\boldsymbol{PF_{true}}$ to that member. The values of the above indicator according the best solutions of ten run for each small problems is presented as table 2 and 3. Table 2 shows the number of Pareto-optimal solutions that three algorithms have found. Each algorithm has been run ten times and the average values are shown in table 2. The best result of each algorithm during ten times of running is shown in table 3. These two tables show that the results of the proposed algorithm TRGA is near to NSGA-II and better than SPEA-II. In other words, the proposed algorithm TRGA provides a higher number of diverse local non-dominated solutions which are closer to the true Pareto-optimal frontier that algorithm SPEA-II.

4.2.3. Medium and Large-Sized problems

It is impossible or very time complexity for the medium and large-sized problems to find the Pareto-optimal solutions. Therefore, the comparison metrics which are used in the Medium and large-sized problems must be restricted to indicators that don't need to Pareto-optimal solutions. Hence, in this section two indicators Overall Non-dominated Vector Generation (ONVG) and Spacing (S) are used to evaluate performance of the proposed algorithm in solving the Medium and large-sized problems.

Tab. 2. Comparison of the number of Pareto-optimal solutions that algorithms found

Problem name	Number of Pareto solutions	Algorithm		
		TRGA	NSGA-II	SPEA-II
HS-I	9	8.33	8.67	8.00
HS-II	10	8.91	9.33	8.67
HS-III	12	10.16	10.16	9.33
HS-IV	16	13.36	14.67	12.16
HS-V	19	15.26	15.26	14.33
HS-VI	24	19.53	20.53	17.50

Tab. 3. Comparison of the Error Ratio (ER) and Generational Distance (GD)

Problem	ER			GD		
	TRGA	NSGA-II	SPEA-II	TRGA	NSGA-II	SPEA-II
HS-I	0.00	0.00	0.00	0.00	0.00	0.00
HS-II	0.10	0.00	0.10	0.14	0.00	0.12
HS-III	0.08	0.08	0.16	0.11	0.11	0.15
HS-IV	0.13	0.06	0.13	0.21	0.16	0.26
HS-V	0.16	0.11	0.21	0.24	0.18	0.28
HS-VI	0.17	0.17	0.25	0.27	0.22	0.32

4-2-3-1. Overall Non-Dominated Vector Generation (ONVG). The Overall Non-dominated Vector Generation (ONVG) measures the total number of non-dominated vectors found during algorithm execution. This Pareto non-compliant metric is defined as equation (25).

$$\mathbf{ONVG} = |\mathrm{PF}_{\mathrm{known}}| \tag{25}$$

4-2-3-2. Spacing (S). The spacing (S) metric numerically describes the spread of the vectors in $\mathrm{PF}_{\mathrm{known}}$. In other word, this indicator measures the distance variance of neighboring vectors in $\mathrm{PF}_{\mathrm{known}}$ as equation (26).

$$S = \sqrt{\frac{1}{|PF_{known}|-1} \times \sum_{i=1}^{|PF_{known}|} (d_i - \bar{d})^2} \quad for\ i,j = 1,2,3,\dots,n \tag{26}$$

Where d_i indicates distances between the i^{th} solution from the nearest solution to it and is calculated as equation (27).

$$d_i = min_j\left(\left|f_1^i(x) - f_1^j(x)\right| + \left|f_2^i(x) - f_2^j(x)\right|\right) \quad for\ i,j = 1,2,3,\dots,n \tag{27}$$

In equation (29), $f_1(x)$ and $f_2(x)$ can be supposed as the makespan and the sum of earliness and tardiness in the considered problem. Also $\bar{d}$ is the mean of all d_i and n is is the number of vectors in $\mathrm{PF}_{\mathrm{known}}$. Table 6 represents the average values of the two above mentioned metrics in the medium and large problems. As illustrated in this table the proposed N-WBGA shows better performance in both medium and large problem sets and the result is near to NSGA-II.

4-2-3-3. Diversification Metric. Generally, multi-objective GA differs based on their fitness assignment procedure, and elitism or diversification approaches. The diversification mechanism in the algorithm is based on niching that results in a wide spread of solutions in the parametric space. It is defined as (28):

$$\mathbf{D} = \sqrt{\sum_{i=1}^{n} \max(\|X_i - X_i'\|)} \tag{28}$$

Where $n = |\mathrm{PF}_{\mathrm{known}}|$ and $\|X_i - X_i'\|$ is the Euclidean distance between of the two non-dominated solutions X_i and X_i' .Tables 4 and 5 represent the best values of the three above mentioned metrics during ten times running of algorithms.

Tab. 4. Comparison of the number of Non-dominated solutions that algorithms found

Problem name	Algorithm		
	TRGA	NSGA-II	SPEA-II
HM-I	35	38	34
HM-II	37	41	36
HM-III	40	43	38
HM-IV	63	68	61
HM-V	66	71	65
HM-VI	68	72	66
HL-I	108	114	99
HL-II	111	118	105
HL-III	116	124	112
HL-IV	175	192	168
HL-V	181	195	177
HL-VI	184	201	179

Tab. 5. Comparison of the number of Non-dominated solutions that algorithms found

Problem name	S			D		
	TRGA	NSGA-II	SPEA-II	TRGA	NSGA-II	SPEA-II
HM-I	7.26	7.04	7.95	12.26	13.72	12.04
HM-II	7.53	7.11	8.16	13.24	14.62	12.96
HM-III	8.13	7.48	9.78	13.53	14.89	12.98
HM-IV	8.64	7.62	9.81	14.05	15.21	14.01
HM-V	9.02	6.96	7.02	14.34	15.43	14.91
HM-VI	9.82	8.02	7.92	14.91	15.32	14.10
HL-I	11.16	9.25	8.85	17.12	17.25	16.81
HL-II	11.21	9.15	9.20	17.49	17.63	16.21
HL-III	11.52	8.95	9.05	18.50	19.05	17.45
HL-IV	11.91	10.01	9.51	21.31	22.01	20.51
HL-V	11.72	10.65	10.11	22.72	23.25	21.19
HL-VI	12.67	10.17	10.12	23.16	24.67	22.08

Table 4 indicates that the proposed algorithm TRGA and algorithm NSGA-II have obtained higher number Non-dominated solutions than algorithm SPEA-II. As illustrated in the table 5, the proposed TRGA shows good performance in both mediocre and large problem sets. In other words, TRGA provides a higher number of diverse local non-dominated solutions. Table 6 presents the average of computational times spent by algorithms after 10 generations executed in each test problem. As illustrated in this table, the proposed algorithm TRGA consumes less computational time than the others in all categories of problems. Because of the implemented structure of the calculations, the higher value of computational time of the NSGA-II is reasonable especially for the small-sized problems.

Tab. 6. Run time of the algorithms

Problem	Algorithm		
	TRGA	NSGA-II	SPEA-II
Small	21	34	30
Medium	198	241	229
Large	1165	1187	1235

5. Conclusion and Future Studies

In this paper a multi-objective scheduling problem was studied for a three-stage production system including a two-stage fabrication phase and an assembly stage. In this production system it is assumed that several products of different kinds are ordered to be produced. The parts are manufactured in the fabrication phase and products are assembled in the assembly stage after providing the parts. Two objective functions are considered simultaneously that are: (1) to minimizing the completion time of all products (makespan), and (2) minimizing the sum of earliness and tardiness of all products ($\sum_i (E_i / T_i)$. Since this problem is NP-hard, a new multi-objective algorithm based on GA was designed for searching local Pareto-optimal frontier for the problem. Various test problems were designed and the reliability of the proposed algorithm was presented in comparison two algorithms NSGA-II, and SPEA-II. The computational results show that the performance of the proposed algorithms is good in three-sized problems. Future research may be conducted this problem in condition that some of the parts are the same for the different products. Also to study this problem with a number of products of the same kind may be interesting as a future study. One can study this problem and proposed some other algorithms using Tabu Search, Simulated Annealing, and Particle Swarm Optimization and comparison their performance. This problem can also be studied with different factors of earliness and tardiness and some other objective functions.

References

[1] Arroyo, J.A.C., Armentano, V.A., (2005) *Genetic local search for multi-objective flowshop scheduling problems*, European Journal of Operational Research, 2005, 167: 717–738.

[2] Loukil, T., Teghem, J., Tuyttens, D., *Solving Multi-Objective Production Scheduling Problems Using Metaheuristics*, European Journal of Operational Research, 2005 161: 42–61.

[3] Lee, C.Y., Cheng, T.C.E., Lin, B.M.T., *Minimizing the Makespan in the 3-Machine Assembly-Type Flowshop Scheduling Problem*, Management Science, 1993 39: 616-625.

[4] Allahverdi, A., Al-Anzi, F.S., *The Two-Stage Assembly Scheduling Problem to Minimize Total Completion Time with Setup Times*, Computers & Operations Research, 2009, 36: 2740-2747.

[5] Sup Sung, Ch., Juhn, J., *Makespan Minimization for a 2-Stage Assembly Scheduling Problem Subject to Component Available Time Constraint*, International Journal of Production Economics, 2009, 119: 392–401.

[6] Lin, R., Liao, Ch.J., *A Case Study of Batch Scheduling for an Assembly Shop*, International Journal of Production Economics, 2012, 139: 473–483.

[7] Potts, C.N., Sevast'Janov, S.V., Strusevich, V.A., Van, Wassenhove, L.N., Zwaneveld, C.M., *The Two-Stage Assembly Scheduling Problem: Complexity and Approximation*, Operations Research, 1995, 43: 346-355.

[8] Hariri, A.M.A., Potts, C.N., *A Branch and Bound Algorithm for the Two-Stage Assembly Scheduling Problem*, European Journal of Operational Research, 1997, 103: 547-556.

[9] Cheng, T.C.E., Wang, G., *Scheduling the Fabrication and Assembly of Components in a Two-Machine Flow Shop*, IIE Transactions, 1999, 31: 135-143.

[10] Koulamas, C.h., Kyparisis, G.J., *A Note on the Two-Stage Assembly Flow Shop Scheduling Problem with Uniform Parallel Machines*, European Journal of Operational Research, 2007, 182: 945–951.

[11] F Sung, C.S., Kim, H.Ah., *A Two-Stage Multiple-Machine Assembly Scheduling Problem for Minimizing Sum of Completion Times*, International Journal of Production Economics, 2008, 113: 1038-1048.

[12] Yokoyama, M., *Hybrid Flow-Shop Scheduling with Assembly Operations*, International Journal of Production Economics, 2001, 73: 103-116.

[13] Yokoyama, M., *Flow-Shop Scheduling with Setup and Assembly Operations*, European Journal of Operational Research, 2008, 187: 1184–1195.

[14] Yokoyama, M., Santos, D.L., *Three-Stage Flow-Shop Scheduling with Assembly Operations to Minimize the Weighted Sum of Product Completion Times*, European Journal of Operational Research, 2005, 161: 754-770.

[15] Fattahi, P., Hosseini, S.M.H., Jolai, F., *A Mathematical Model and Extension Algorithm for Assembly Flexible Flow Shop Scheduling Problem*, International Journal of Advance Manufacture Technology, 2012, DOI 10.1007/s00170-012-4217-x.

[16] Konak, A., Coit, D.W., Smith, A.E., *Multi-Objective Optimization Using Genetic Algorithms: A Tutorial*, Reliability Engineering and System Safety, 2006, 91: 992–1007.

[17] Sun, Y., Zhang, Ch., Gao, L., Wang, X., *Multi-Objective Optimization Algorithms for Flow Shop Scheduling Problem: a Review and Prospects*, Int J Adv Manuf Technol, 2010, DOI 10.1007/s00170-010-3094-4.

[18] Coello, C.A., Lamont, G.B., Veldhuizen, D.A.V., *Evolutionary Algorithms for Solving Multi-Objective Problems*, Second edition, 2007, Springer.

[19] Karimi, N., Zandieh, M., Karamooz, H.R., *Bi-Objective Group Scheduling in Hybrid Flexible Flowshop: A Multi-Phase Approach*, Expert Systems with Applications, 2010, 37: 4024–4032.

[20] Moslehi, G., Mirzaee, M., Vasei, M., Modarres, M., Azaron, A., *Two-Machine Flow Shop Scheduling to Minimize the Sum of Maximum Earliness and Tardiness*, Int. J. Production Economics, 2009, 122: 763–773.

[21] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., *A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II*, IEEE Transactions on Evolutionary Computation, 2002, 6 (2) 182–197.

[22] Zitzler, E., Laumanns, M., Thiele, L., *SPEA2: Improving the Strength Pareto Evolutionary Algorithm*, Computer Engineering and Networks Laboratory (TIK) – Report, 2001, 103, September

[23] Rahimi-Vahed, A.R., Rabbani, M., Tavakkoli-Moghaddam, R., Torabi, S.A., Jolai, F., *A Multi-Objective Scatter Search for a Mixed-Model Assembly Line Sequencing Problem*, Advanced Engineering Informatics, 2007, 21: 85–99.

[24] RAHIMI-VAHED, A.R., MIRGHORBANI, S.M., RABBANI, M., *A Hybrid Multi-Objective Particle Swarm Algorithm for a Mixed-Model Assembly Line Sequencing Problem*, Engineering Optimization, 2007, 39 (8) 877–898.