Journal of Quality Engineering and Production Optimization

Vol. 2, No. 1, PP.27-46. 2017

New Formulation and Solution in PCB Assembly Systems with Parallel Batch processors

Iman Rastgar and Rashed Sahraeian*

Department of Industrial Engineering, College of Engineering, shahed University,

***Corresponding Author:** Rashed Sahraeian (E-mail: sahraeian@shahed.ac.ir)

Abstract- *This paper considers the scheduling problem of parallel batch processing machines with non-identical job size and processing time. In this paper, a new mathematical model with ready time and batch size constraints is presented to formulate the problem mathematically, in which simultaneous reduction of the makespan and earliness-tardiness is the objective function. In recent years, the nature-inspired computational intelligent algorithms have been successfully employed to achieve the optimum design of different structures. Since the proposed model is NP-hard, a metaheuristic algorithm according to a harmony search algorithm is developed and analyzed for solving the batch processing machine scheduling problem addressed in the current paper. Various parameters and operators of the proposed harmony search algorithm are discussed and calibrated by means of the Taguchi statistical technique. In order to evaluate the proposed algorithm, instance problems in concordance with previous research are generated. The proposed algorithm and basic harmony search, improved harmony search and global best harmony search are solved and the results of all the algorithms are compared. The conclusion reveals that the proposed algorithm performs better than the other algorithms.*

Keywords- *batch processing, harmony search algorithm, scheduling, Taguchi design of experiments, parallel machine*

I. INTRODUCTION

Stations of parallel batch processors are observed in the semiconductor industries and printed circuit board (PCB) manufacturing. The electronics industry is one of the greatest manufacturing industries in the world. One of its important sections is integrated circuits (IC). The similarity between scheduling problems and the semiconductor industry can be found in PCB production, as the jobs are input into environmental stress screening (ESS) ovens and then processed. The ovens are the same batch machines as in the studied scheduling problem. A similar ESS was designed in parallel in a workstation. PCBs after batch assembly are input for the determination of defective jobs. The number of circuits in each batch should not exceed the dimensional capacity of the batch. PCBs output from different production lines, waiting for input to one of the EES ovens, form a queue and are categorized into batches for final testing. Each line contains several PCBs with different sizes and processes within different times. When a batch consists of a finite number of PCBs, the batch processing time is equal to the longest processing time of an individual PCB. Considering the different availability times of each batch of PCBs, the processing ready time of each batch is equal to the last PCB that arrives at that batch.

Semiconductor factories, facing massive demand, must retain their competitive advantage in the market, as well as utilizing their resources effectively. ESS ovens, which are similar to thermal ovens, are very valuable and costly. Therefore, in order to reduce the production costs, we must optimize the total jobs (C_{max}) processing period and other costs supporting the scheduling techniques. In the past, the reduction of costs included reducing the circuits size, increasing the wafer sizes and return improvement, whilst simultaneously improving the operational processes in the semiconductor production systems. Critical factors which must be considered by the semiconductor industries include their capability to adapt and apply modern and advanced technologies in their products, continuous improvement of production processes, and the ability to meet orders on their due dates to avoid customer dissatisfaction.

Manuscript Received: 29-May-2014 and revised: 02-Nov-2015 *ISSN: 2423-3781*
Accepted: 30-Oct-2016

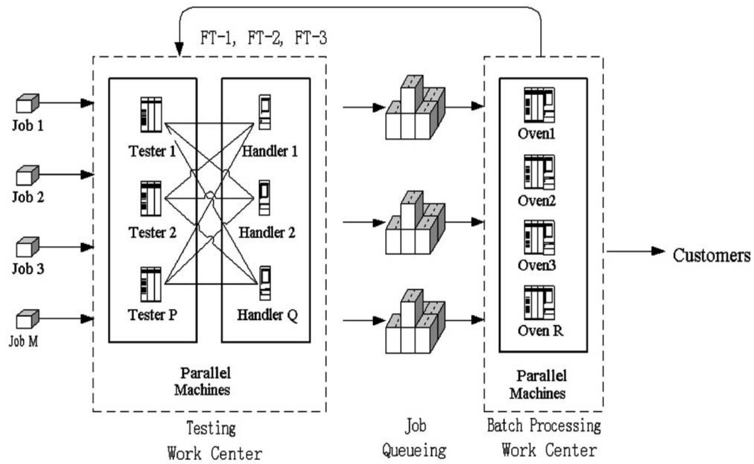

Fig. 1. Chip production processes in semiconductors (Velez Gallego & Adviser-Damodaran, 2009)

The chips production process in semiconductor factories is as shown below: firstly, thin circular silicon wafer plates are produced during several stages, then integrated circuits are assembled thereon. The chips production process consists of four main stages as follows:

1- Wafer production, 2- Wafer inspection, 3- Assembly, 4- Final test.

As per Figure 1, the studied problem is implemented in the fourth stage. Heat treatment at this stage is implemented by placing the chips in an oven, which is the same ESS for heating at high temperature where defective chips are revealed. The aforementioned stages are used in factories manufacturing chips in mass volume.

One of the important and applicable problems in engineering optimization is scheduling where, considering the assumed hypotheses therein and the large dimensions of the problem, obtaining appropriate answers within a reasonable time is complex. In scheduling problems, we deal with the allocation of a set of jobs to a set of machines, determining and prioritizing the processing time so as to achieve the maximum productivity of resources such as operators and machines (Pinedo, 2012). A new generation of processing jobs in connection with scheduling problems was required with the adoption of simultaneous batch processing in industrial environments, which reduces the setup time, transportation costs, and accelerates processing operations compared to single processing of jobs. In this type of processing, considering the limitation on each batch size for processing, finite jobs up to an extent not exceeding the batch capacity are placed in the batch and processed on the batch processor. The problem studied in this paper includes jobs scheduling and sequencing in a single-station environment containing M parallel machines, where the jobs are divided into batches and each batch is processed directly on one of the processors before exiting from the station. The major limitations of this problem include the job availability time and job capacity limitation in each batch.

In most scheduling systems, the attempt is made to reduce the total costs. Important costs in manufacturing companies are divided into two main categories: production costs and customer costs. The most important production costs include the maximum completion time and the cost arising from early production compared to the customers' delivery time, which imposes warehousing and maintenance costs. Meanwhile, the most important costs related to the customer include the cost of non-timely delivery of goods, meaning early delivery. In the recent decade, several researchers have carried out studies on minimizing these two costs in different scheduling environments (Bank & Werner, 2001; Kedad-Sidhoum, Solis, & Sourd, 2008; Mönch & Unbehaun, 2007; Moslehi, Mirzaee, Vasei, Modarres, & Azaron, 2009; Toksarı & Güner, 2009). The maximum completion time cost includes the costs of machinery, human resources, energy and other costs related to the production completion period. Tardiness costs are imposed when an order is delivered to the customer later than the determined date. This delivery cost includes customer dissatisfaction, contract penalties, sales lost, loss or damage to the reputation of the manufacturer and retailer. Also, if the allocated order's completion time is earlier than its delivery time, the company is involved in delivery earliness, meaning that it is obliged to hold the order in the company until delivery time. Hence, the delivery earliness cost is assumed as the storage costs or inventory cost of the order. In the presented model, the scheduling and optimal sequence of jobs operation is determined as distinct batches in a station containing M parallel batch machines with the purpose of simultaneous minimization of these three important costs. In order to solve the problem, considering the complexity of the studied problem, four metaheuristic methods are used independently based on the harmony search method.

The harmony search algorithm (HS) is one of the simplest and newest metaheuristic methods. The optimal responding search process in optimization problems has been inspired by the process of simultaneous playing of an orchestra. This solution method has been presented by (Geem, Kim, & Loganathan, 2001).

The paper is structured as follows: in chapter 2, the literature on the mentioned problem is reviewed. In chapter 3, the problem is defined. In chapter 4, the solutions procedures are proposed. In chapter 5, the design method of experiments applied in the solution algorithms is explained. In chapter 6, the results of the proposed model solution are shown, and the final chapter gives the conclusion of this paper and explains grounds for subsequent research.

II. LITERATURE REVIEW

The environment of parallel machines, where batch processors are designed in parallel form and the jobs are processed simultaneously in batches, is assumed as one of the most important environments in scheduling problems, but while single processors have attracted considerable attention in scheduling research, research on parallel machine and batch processor environments has rarely been undertaken. (Chang, Damodaran, & Melouk, 2004) offered a meta-heuristic simulated annealing (SA) algorithm for scheduling of batch parallel processors and compared its results to a commercial solver, concluding that the metaheuristic algorithm is more efficient in computation time indices and solution quality for problems with big dimensions in parallel machines and batching jobs. (Xu & Bean, 2007) made an integer planning model to minimize the total completion time and a genetic algorithm whose solution exhibiting mode is based on the random values for unequal parallel machines. (Shao et al., 2008) considered a solution approach based on neural networks assuming zero available times. They compared their results to the FFLPT and BFLPT heuristic methods. (Chung, Tai, & Pearn, 2009) offered a mathematical model and three heuristic methods for minimizing the total completion time under non-zero available time. In subsequent research, the same group presented a hybrid approach wherein batches are first formed and then scheduled. To form the batches, they used the delay heuristic method which had been proposed by (C.-Y. Lee, 1999) for a single machine model, and to schedule the batches on the parallel machines, they presented two non-delay schedulings by two simple scheduling rules.

Minimizing the maximum jobs completion time in the batch parallel machines has been studied in other research by (Kashan, Karimi, & Jenabi, 2008). They first offered a low limit for the maximum optimum completion time. Then, they suggested a heuristic method based on an extended hybrid genetic algorithm. The suggested algorithm was compared to a simulated annealing algorithm available in previous research that performed better than SA. (Damodaran, Hirani, & Velez-Gallego, 2009) proposed a genetic algorithm for minimizing the completion time of batch parallel machines. They compared their results to the SA metaheuristic approach suggested in the paper by (Chang et al., 2004) and the random key genetic algorithm (RKGA) (Xu & Bean, 2007) of this study and the CPLEX commercial solver. The genetic algorithm offered by Damodaran was very effective in finding good answers. (H.-M. Wang & Chou, 2010) presented a hybrid integer planning model for minimizing the maximum jobs completion time on the batch parallel machines, wherein the preparation time for jobs was applied. Then, in order to solve the model, they suggested a simulated annealing algorithm and a genetic algorithm. Also, a multi-stage dynamic planning algorithm categorizes the jobs for each machine.

(Damodaran, Vélez-Gallego, & Maya, 2011) offered a greedy randomized adaptive search for minimizing the maximum jobs completion time on batch machines in parallel form, applying the setup time for jobs. In this problem, the batch preparation time is equal to the longest time for preparation of jobs in the batch. Their suggested heuristic method performed better than a lower bound and several heuristic methods available in the former researches. In continuation, (Damodaran & Velez-Gallego, 2010) presented another heuristic method for minimizing the maximum jobs completion time on the batch parallel processing machines. The proposed heuristic method was compared, by means of some sample problems, to several heuristic methods proposed in former research, such as the same heuristic method as their previous research and performed better by comparison.

(B. Cheng, Wang, Yang, & Hu, 2013) proposed the new method of ant colony optimization (ACO) for the problem of identical parallel machines. This algorithm is the main criterion used for ants' routes selection to dominant undeveloped convergences. (Feng, Yuan, Liu, & He, 2013) studied two-agent scheduling on an unbounded parallel-batching machine problem. They considered two agents A and B, each having an objective function to be minimized. The objective function of agent A is the makespan of his jobs and the objective function of agent B is the maximum lateness of his jobs. They also present a polynomial-time algorithm for finding all the Pareto optimal solutions. In (Li, Huang, Tan, & Chen, 2013), several heuristics based on the best fit longest processing time (BFLPT) in two groups are proposed to solve the scheduling of unrelated parallel batch processing machines. In their paper, the objective function is minimizing the makespan. (Damodaran, Diyadawagamage, Ghrayeb, & Vélez-Gallego, 2012) presented a PSO algorithm to schedule jobs on non-identical parallel batch processing machines such that the makespan was minimized. The PSO algorithm was compared with the random key genetic algorithm and CPLEX solver. (Cakici, Mason, Fowler, & Geismar, 2013) studied the problem of minimizing weighted completion times on identical parallel batching machines with dynamic job arrivals and incompatible job families. They presented a mathematical model and heuristic algorithms based on different local search procedures.

(Yilmaz Eroglu, Ozmutlu, & Ozmutlu, 2014) proposed a random key genetic algorithm with a local search procedure for a parallel non-identical machine scheduling problem with the objective of minimizing the makespan. (Jia & Leung, 2015) gave simple heuristic and metaheuristic algorithms, and showed that both outperform the PSO algorithm given by (Damodaran et al., 2012) by a wide margin. (J.-Q. Wang & Leung, 2014) considered a set of equal-processing-time jobs. They showed that, unless P = N P, there is no polynomial-time algorithm with an absolute worst-case ratio less than 2. They then gave a polynomial-time algorithm with an absolute worst-case ratio of exactly 2. Finally, they gave a polynomial-time algorithm with an asymptotic worst-case ratio of 3/2. (Xu & Bean, 2015) studied the problem of minimizing the total weighted tardiness, a proxy for maximizing on-time delivery performance, on parallel non-identical batch processing machines. They used genetic algorithms, based on random keys and multiple choice encodings, to heuristically solve them.

(Chang et al., 2004) introduced a mathematical model for the studied problem for a state wherein a ready-time limitation was not assumed. Subsequently, (Chung et al., 2009) presented a mathematical model assuming an available time limitation like that presented by (Velez Gallego & Adviser-Damodaran, 2009), but changing the mathematical model, and assuming the batches position on each machine. Considering the related research in the field of parallel batch processing machines, no study has been applied where three objective functions including the maximum completion time, earliness and tardiness of jobs have been considered simultaneously in the problem. In this paper, the objective function of the proposed model is reduced to a linear aggregation of three respective objective functions and the studied problem will be solved according to the novel proposed metaheuristic solution based on the harmony search method. In other words, considering related surveys in the area of parallel batch processing machines, research in which the three functions of the makespan objective and earliness and tardiness of jobs are considered in the problem simultaneously, is not found. In this paper, continuing the research of (Chung et al., 2009), the proposed model objective function in the form of three target functions is checked out and considered as a linear integration. Also, it is noteworthy that this problem is taken from a real situation which is seen in producing products like semiconductors, so the model expresses the limitations of such a real case. All previous research on this problem used only the makespan objective function, while in applicable examples the objective function and, in practice, other functions are required.

III. PROBLEM DESCRIPTION

$$Z = MIN\left\{\begin{array}{l}\alpha \times (\max\{C_j\}) + \sum_{j=1}^{n} \beta_j \times \max\left\{0, d_j - \sum_{i=1}^{m}\sum_{k=1}^{b} X_{ijk} C_j\right\} \times S_j \\ + \sum_{j=1}^{n} \gamma_j \times \max\left\{0, \sum_{i=1}^{m}\sum_{k=1}^{b} X_{ijk} C_j - d_j\right\} \times S_j\end{array}\right\}$$

s.t:

$$\sum_{i=1}^{m}\sum_{k=1}^{b} x_{ijk} = 1 \qquad j=1,2 \ldots ,n \qquad (1)$$

$$\sum_{j=1}^{n}\sum_{j=1}^{m} s_j x_{ijk} \leq B \qquad k=1,2 \dots ,b \qquad (2)$$
$$T_{ik} \geq r_j x_{ijk} \qquad i=1,2\dots ,m \quad j=1,2 .. n \quad k=1,2 \dots ,b \qquad (3)$$
$$T_{ik} - T_{ik-1} \geq P_j x_{ijk} \qquad i=1,2\dots ,m \quad j=1,2 .. n \quad k=1,2 \dots ,b \qquad (4)$$
$$\left[\sum_{j=1}^{n} S_j\right] \Big/ B \leq b \leq n \qquad b \in Z^+ \qquad (5)$$
$$x_{ijk} \in \{0,1\} \qquad i=1,2\dots ,m \quad j=1,2 \dots n \quad k=1,2 \dots ,b \qquad (6)$$

- *Index*

I: machine index, i=1, 2 …, m
J: job index, j=1, 2 … n
K: batch index, k=1, 2 …, b

At the beginning of the scheduling time, there are n jobs $\{j_1, j_2, \dots, j_n\}$ with processing time p_j, ready time r_j and job size s_j. These jobs are entered into the workstation considering their availability time, where M similar machines are located in parallel form. The similar parallel machines all have equal processing capacity and process the batches with equal speed. Each machine has the maximum simultaneous processing capacity of B job units as a batch, while each batch is processed on the machine directly, and has a capacity equal to the machine's capacity size. Before processing on the machines, the jobs are placed in batches $(k: 1, 2, \dots, b)$ so that the total size of the jobs in the batch is equal to or less than the machine capacity. When the machines are idle, the batches are allocated to each one of them. After processing, each batch exits from the station and is replaced by the next batch. The size of each job means the number of products in the order demanded by the customer. The batch processing time is the maximum processing time of one of the jobs belonging to the batch and each batch's processing start time is equal to the maximum availability time among ready times of batch jobs.

The studied problems include two stages; in the first stage, the total available jobs are grouped into categories. So, using heuristic methods for making up the batches has a strong effect on the quality of the final solution. In the second stage, the established batches are allocated to the available parallel machines, and then the batches are sorted regularly on each machine in order to optimize the respective objective function.

In this section, the mixed non-linear mathematical modelling of the problem is provided according to the explained assumption, and the next section presents its solution procedure by metaheuristic methods to minimize the maximum completion time cost and the earliness and tardiness penalty costs.

- **parameter's**

B: size capacity of each batch
s_j: size of job *j* (i.e. the amount of items ordered by the customer for job *j*)
r_j: ready time of job *j*
P_j: processing time of job *j*
α: the cost factor of the maximum completion time
β_j: the penalty factor of any increment in completion earliness of job $_j$
γ_j: the penalty factor of any increment in completion tardiness of job *j*

- **Variables**

T_{ik}: beginning time of batch *k* processing on machine *i*;
X_{ijk}: decision variable that includes 0 and 1 modes. If job *j* is grouped in batch *k* and processed on machine *i*, X_{ijk} is equal to 1, otherwise 0.

The first objective function of the problem is minimizing the maximum completion time cost considering factor α. The second objective is minimizing the job's earliness cost considering the factor that is the penalty of any increment in job completion earliness and is explained for each unit. Therefore, the job number is assumed as s_j. The third objective function is minimizing the job's tardiness cost considering factor γ_j that a penalty is assumed for increase of the job's tardiness time per unit. This value is explained for one unit of each job that is calculated by its multiplication by the size of each job.

The first limitation guarantees that each job is placed only in one batch and that the batch is processed only on one machine. The second limitation is related to the machine capacity; the total jobs making up the batch should not exceed the machine capacity. The third limitation relates to the job's ready time, consequently the batch's processing start time should be at least equal to the maximum jobs available time. The fourth limitation shows that when two consecutive batches are processed on one machine, the minimum time difference between the two processing start times is equal to the processing time of the job in the second batch with the maximum value. The fifth limitation indicates the minimum and maximum batches. The sixth limitation indicates that the decision variable is binary.

A. Characteristics of respective problem

- Each job has an arbitrary size (s), processing time (p) and ready time (r) and all values are deterministic.
- The number of machines is at least 2 and at most M.
- The total size of the jobs constituting each batch do not exceed the batch capacity (B).
- It is assumed that there is no job whose individual size is bigger than the batch capacity.
- While processing a batch, jobs may not be added to or removed from the batch.
- The machines have equal processing capacity B.
- All machines are batch processors.
- Machine failure is not permitted.

Below is an example presented for a better description of this problem, which includes a workstation for batch processing of 10 jobs on 2 identical parallel machines.

The capacity of each batch is 7. Data related to the jobs is presented in Table 1.

An example of a feasible solution of the model if five batches are made is shown in the following figure:

Considering this type of job batching and assignment to parallel machines, the values of the objective functions of the proposed model are as given in Table 2.

TABLE I. Data related to jobs in simple example

job (j)	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Processing time (P_j)	7	6	10	7	5	3	9	4	9	4
Size (s_j)	1	3	1	3	3	2	3	1	3	2
Due date (d_j)	8	7	10	9	7	15	20	15	15	22

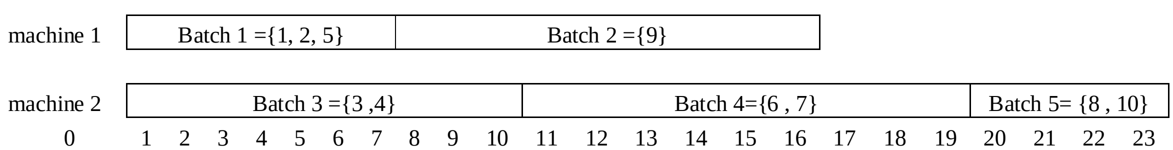

Fig. 2. A feasible solution of model in simple example

TABLE II. Value of objective functions of simple example

Job (j)	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Completion time (C_j)	7	7	10	10	7	9	9	23	16	23
Earliness (E_j)	1	0	0	0	0	6	12	0	0	0
Tardiness (T_j)	0	0	0	1	0	0	0	8	1	1

IV. SOLUTION PROCEDURES

A. soft computing procedures

Since the respective problem is NP-hard, to solve a problem with bigger dimensions by exact procedures such as simplex, dynamic programming, or branch and bound, we may not be able to reach the optimum answer within a reasonable time. Hence, due to the intrinsic complexity of discrete optimization problems and particularly production scheduling problems, using metaheuristic methods gives a better performance for solving the problem and providing an acceptable answer within an acceptable time. In this paper, in order to solve the problem, four metaheuristic algorithms including simple harmony search, improved harmony search, global best harmony search and hybrid Taguchi, and the novel global best harmony search (TNHS) are compared to each other. Metaheuristic algorithms, as powerful tools, have been used in different contexts for seeking the best solution in optimization problems. Despite the efficiency of metaheuristic algorithms for solving decision-making and optimization problems, these algorithms require a lot of accuracy and delicacy during the implementation of experiments. Inappropriate application and adjustment of their parameters reduce the efficiency and effectiveness of these methods. Parameter adjustment means selecting the best mode or value for the parameters so as to provide the optimum (the best possible) performance of the algorithm. These parameters may have a great effect on the algorithm’s efficiency (Ruiz & Maroto, 2006).

B. Review of harmony search algorithm

The harmony search algorithm has been increasingly taken into consideration within recent years and it has been used so far in practical optimization problems such as structural optimization, estimation of non-linear Muskingum model parameters, the optimal design of water distribution networks, routing, design of metal skeletons, energy transmission models, scheduling etc. In some researches, HS has been shown to obtain the appropriate answers earlier than genetic methods (K. S. Lee, Geem, Lee, & Bae, 2005; Mahdavi, Fesanghary, & Damangir, 2007). In this section, some methods based on the harmony search algorithm are reviewed in order to present an efficient method for solving the studied scheduling problem, according to these researches.

The improved harmony search algorithm (IHS) was presented in research by (Mahdavi et al., 2007). In this algorithm, a new method was offered for a novel solution vector whose accuracy and convergence improved on the basic harmony search algorithm. IHS is similar to HS but with a small difference based on the values of the Pitch Adjustment Rate (PAR) and bandwidth (BW), which are adjusted dynamically and separately in each iteration in compliance with the following relations. In this study, they show that in most of the reviewed problems, IHS performs better than HS. The global best harmony search algorithm (GBHS) was presented in research by (Omran & Mahdavi, 2008). In this method, the pitch adjustment has been designed so that the new vector imitates the best vector available in the memory. The idea of this method is similar to the Particle Swarm Optimization (PSO) method. The PAR parameter, similar to the previous method, is varied in each iteration dynamically, and the BW parameter was omitted from the solution improvement process. The results obtained from the proposed method in nine continuous optimization objective functions without any limitation were compared to the results of IHS and HS and, in most cases, GBHS acts better and a sensitivity analysis was provided for adjusting the PAR, HMS and HMCR parameters. A novel global harmony search (NGHS) algorithm was introduced by (Zou, Gao, Li, & Wu, 2011). In this algorithm, a change in the third pitch of the HS that includes improvising a novel solution was created, where the novel solution imitates the best solution available in the HS memory. Two new factors, the Trust Region and Adaptive Step, were designed in the NGHS. Based on these two factors, new position updating in NGHS is done for moving inappropriate solutions towards the best solution available in each iteration. This will obviate the disadvantage of the HS algorithm, namely the insufficient convergence, by incorporating the probability of genetic mutation p_m with a low probability value. This consequently increases the diversity of the candidate solutions and also solves the space search capacity in NGHS. A self-adaptive global harmony search (SGHS) algorithm has been presented in research by (Pan, Suganthan, Tasgetiren, & Liang, 2010). SGHS was inspired by a method presented in research by (Omran & Mahdavi, 2008). In this algorithm, a novel method was introduced for offering the novel solution and procedures for dynamic adjustment of the algorithm parameters. Similar to GBHS, it follows the characteristics of the best solution in the memory and the current iteration element is selected randomly from among the best memory solution vectors. The novel solution may have an inappropriate structure and may not be as good as the X_B vector, so in this method an element similar to the best solution element is used for making the novel solution. SGHS needs the exact adjustment of its parameters in the different problems, because it is provided with a learning mechanism for determination of the (PAR) HMCR parameters and adjustment of the BW parameter dynamically decreasing with the population number. A harmony search algorithm with a dynamic sub-population (DSHS) was presented in research by (Pan, Suganthan, Liang, & Tasgetiren, 2010). In this algorithm, the algorithm memory is varied dynamically and the whole algorithm memory is divided into subsets smaller than the HMS. Each memory subset runs its evolution process for finding the best solution and exchanges its data with the other subsets periodically by means of regrouping in order not to become trapped in a local optimum. In this algorithm, in a novel solution making section, a novel solution process by means of data of the best local solution is used also in each subset. The lower the number of vectors in each subset, the more the convergence velocity and diversity will be balanced. In each iteration R, the total solution vectors in memory are classified again randomly in subset groups and, continuing the same process, the previous generation is continued up to R iterations. The data obtained from each subset is exchanged between solutions in the previous search process and the diversity of each subset is increased. A highly reliable harmony search algorithm has been presented in research by (Taherinejad, 2009). This method is similar to HS, but the PAR computation procedure is descending linearly. According to this procedure, when the PAR is higher at the beginning of the algorithm iterations (initial generations), the diversity of searching the global space of the problem solution is greater and as we get closer to the terminal generations, the PAR becomes lower, the diversity decreases and the local search increases.

C. Hybrid Taguchi and novel global best harmony search algorithm (TNHS)

This algorithm has been proposed after extensive research on the methods reviewed in literature on the harmony search method. In this method, novel innovations have been offered in the parameters adjustment and changes in the HMCR and PAR parameters process during the algorithm iterations section, compared to GBHS. As the GBHS algorithm, compared to the results of several experiments, has worked better than HS and IHS, consequently this algorithm is assumed as the basic algorithm, and in changing its parameters and operators, we attempt to reach a more efficient method for solving optimization problems. Metaheuristic algorithms have been used as powerful tools in different contexts for seeking the best solution in optimization problems. Despite the efficiency of metaheuristic algorithms for solving decision-making and optimization problems, these algorithms require great accuracy and delicacy during the implementation of experiments. Inappropriate application and adjustment of its parameters reduces the efficiency and effectiveness of these methods. Parameters adjustment means selecting the best mode or value for the parameters to provide the optimum level (best possible) performance of the algorithm. These parameters may have a noticeable effect on the algorithm's efficiency and effectiveness.

In searching for the optimum solutions, a more efficient method is one that, at the beginning of the search, takes into consideration the diversity of the whole solution space (exploration), meaning that it identifies the whole solution space and moves mostly towards spaces in which the optimum will be located with a greater probability. Then, it may face two types of optimum, namely the local optimum and global optimum.

If the algorithm is situated in the local optimum, it may be separated from that zone by means of local search techniques. As shown in Figure 3, the solutions reach the objective value at a higher speed. According to the foregoing, the HS algorithm may identify the high functional areas in the solution space within an appropriate time, but it is not efficient in the implementation of local search in the hybrid optimization problems and in such cases is situated in the local optimum. One of the techniques used for controlling this problem is a restart phase. Using this technique, a shock is applied to the location in which the algorithm is presently situated in the problem solving space and increases its dispersal.

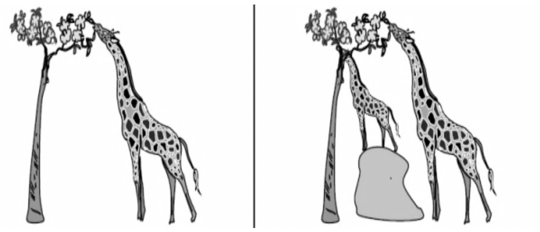

Fig. 3. The role of local search to find optimal solutions

In this method, when the best value is not improved after a number of iterations, it acts as follows:

First step: the harmony memory is sorted increasingly based on the best objective function value.

Second step: the first memory list solutions are sorted and stored equal to the restart phase consideration rate%, then for 1–Restart phase consideration rate of remaining memory, selects a half by a single dot mutation operator on regenerate restart phase rate, and makes the other half based randomly on the range of each solution's elements.

D. Adjustment of parameters in Hybrid Taguchi and novel global best harmony search (TNHS)

In the hybrid Taguchi and novel global best harmony search algorithm (TNHS), there are some parameters including the number of algorithm memory vectors (*hms*), minimum pitch adjustment rate (PAR_{min}), maximum pitch adjustment rate (PAR_{max}), minimum memory consideration rate ($hmcr_{min}$), and maximum memory consideration rate ($hmcr_{max}$) that must be adjusted by appropriate values. Parameter adjustment means selecting the best value for parameters so that the algorithm performance is at the optimum level. Thus, each of the algorithm's parameters has an intense effect on the algorithm's efficiency and effectiveness. Inappropriate adjustment of the parameters may cause inappropriate results to be obtained in the study problem. The appropriate parameters for the algorithm in one problem may not be deemed appropriate for solving another problem, so in each problem the parameters must be adjusted separately. In the proposed algorithm, there are three types of parameter adjustment: dynamic adjustment, adaptive adjustment and sequential adjustment based on the Taguchi method.

- Dynamic adjustment

The pitch adjustment rate parameters (PAR) and harmony memory consideration rate (HMCR) are dynamically varied linearly during the search process, which is presented by a mathematical formula in Figure 4, and in these parameters during the search process, different values of parameters are appropriate, so a constant value throughout the whole search process is not the best value. (T: algorithm stopping time, t: elapsed time of algorithm implementation)

- Adaptive adjustment

The value of the objective function of solution vectors in memory, generation by generation, upon increasing the algorithm iterations does not become worse until reaching the stopping criteria, because of having memory in the algorithm structure and its effect on the search process improvement.

- Sequential adjustment by Taguchi method

The appropriate combination of parameters is extremely effective for the final solution of the algorithm, so in order to specify the best parameters, the best hybrids of five parameters are computed. Considering the optimum levels of each parameter including three levels, altogether there are 3^5 parameter hybrids for implementation of each problem. Considering the sample number of problems (90) and five-times iteration of each experiment as explained in chapter 5, in total there are $90\times5\times3^5 = 109350$ examinations, and specifying the best hybrid from this number of experiments requires a lot of time and is not reasonable. Thus, the Taguchi design of experiments technique is used. Figure 5 exhibits the flowchart of TNHS.

The method proposed in this study is compared in Table 3 to the HS, IHS and GBHS methods and innovations and parameter variations have been considered.

$$HMCR(t) = HMCR_{min} + \frac{HMCR_{max} - HMCR_{min}}{T} \times t \qquad PAR(t) = PAR_{max} - \frac{PAR_{max} - PAR_{min}}{T} \times t$$

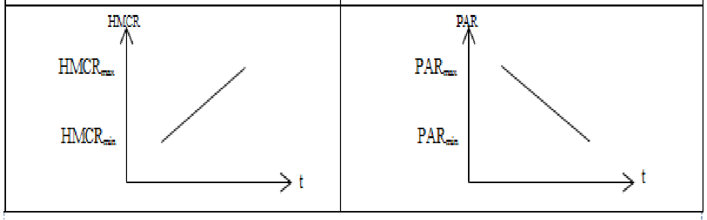

Fig. 4. Variations of PAR and HMCR with increasing generations

TABLE III. Comparing between proposed algorithm with other version of harmony search algorithm

	Parameters				**Innovation**
	HMS	**HMCR**	**PAR**	**BW**	
HS	Constant	Constant	Constant	Constant	Imitating the improvisation process of musicians
HIS	Constant	Constant	Dynamic-Ascending	Exponential	IHS employs a novel method for generating new solution vectors
GBHS	Constant	Constant	Dynamic-Ascending	Nothing	concepts from intelligence swarm are borrowed
TNHS (present research)	Constant	Dynamic-Ascending	Dynamic-descending	Nothing	1-hybrid restart phase as a local search and GBHS 2-all parameters are tuned with taguchi method

V. COMPUTATIONAL EXPERIMENTS

A. *Sample problems design*

Sample problems are designed according to the paper as per the following procedure that is used for evaluation of the performance of the presented algorithms. In this study, 30 scheduling problems of parallel batch machines have been made as per Table 4, with differences among the parameters including the processing time, ready time, job sizes, and numbers of jobs and machines. All these problems include 10 hybrids of the number of jobs (n) and number of batch processors (m). For greater reliability and to eliminate random factors, each problem was implemented three times independently. The independence of each iteration means that, after each implementation, the results are completely independent and have no association with each other.

Three types of problems have been designed for each hybrid and we have altogether 30 problems. In such problems, it is assumed that the size of each batch (B), which is the capacity of each machine, is equal to 7 in the first 15 problems, and 10 in the subsequent 15 problems. Also, for greater reliability, each problem will be implemented three times, so we have 90 implementations for each algorithm in the problem-solving process.

The due dates in these sample problems are created based on the equation proposed by Tavakkoli-Moqhaddam et al. (2007):

relation 1
$$\overline{P} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} P_{ij}}{m \times n}$$

In this proposed equation, after calculation of the average processing time ($\overline{P}$), the p equation is as follows:

relation 2
$$P = (m + n - 1) \times \overline{P}$$

In the above equation, n refers to the available jobs, whereas in the batch processing machine stations, machines are not faced with the number of jobs directly, but with the number of batches. For this reason, instead of the above equation, an equation is used in which the average number of batches that is usually formed ($\overline{B}$) is obtained. So the total number of jobs is calculated in order to determine the number of batches which can be formed in relation to the job size. Then the total sum of job sizes is divided into 0.8 of batch capacity, because, from the preliminary experiments, it is expected that on average, among the formed batches, 0.8 of capacity is occupied. Also, in relation to the above problem, from the fact that the problem in the Tavakkoli and colleagues (2007) article was a flowshop environment, by changing the presented relation and modifying it as in the third column of Table 5, we can match it in the parallel machine environment. After the above calculations, with the help of the changes in the equation, the following relations are considered for creating the due dates.

TABLE IV. Values of parameters for design of sample problem test

M*n	Type	Processing time	time Ready	Job Size
3*10	A	U~[1,20]	U~0.2*[1,Sum(processing time(A))]	U~[1,7]
	B	U~[1,30]	U~0.4*[1,Sum(processing time(B))]	
	C	U~[1,50]	U~0.49*[1,Sum(processing time(C))]	
3*20	A	U~[1,20]	U~0.2*[1,Sum(processing time(A))]	U~[1,7]
	B	U~[1,30]	U~0.4*[1,Sum(processing time(B))]	
	C	U~[1,50]	U~0.49*[1,Sum(processing time(C))]	
3*50	A	U~[1,20]	U~0.2*[1,Sum(processing time(A))]	U~[1,7]
	B	U~[1,30]	U~0.4*[1,Sum(processing time(B))]	
	C	U~[1,50]	U~0.49*[1,Sum(processing time(C))]	
3*100	A	U~[1,20]	U~0.2*[1,Sum(processing time(A))]	U~[1,7]
	B	U~[1,30]	U~0.4*[1,Sum(processing time(B))]	
	C	U~[1,50]	U~0.49*[1,Sum(processing time(C))]	
3*200	A	U~[1,20]	U~0.2*[1,Sum(processing time(A))]	U~[1,7]
	B	U~[1,30]	U~0.4*[1,Sum(processing time(B))]	
	C	U~[1,50]	U~0.49*[1,Sum(processing time(C))]	
5*40	A	U~[1,20]	U~0.2*[1,Sum(processing time(A))]	U~[1,10]
	B	U~[1,30]	U~0.4*[1,Sum(processing time(B))]	
	C	U~[1,50]	U~0.49*[1,Sum(processing time(C))]	
5*80	A	U~[1,20]	U~0.2*[1,Sum(processing time(A))]	U~[1,10]
	B	U~[1,30]	U~0.4*[1,Sum(processing time(B))]	
	C	U~[1,50]	U~0.49*[1,Sum(processing time(C))]	
5*120	A	U~[1,20]	U~0.2*[1,Sum(processing time(A))]	U~[1,10]
	B	U~[1,30]	U~0.4*[1,Sum(processing time(B))]	
	C	U~[1,50]	U~0.49*[1,Sum(processing time(C))]	
5*200	A	U~[1,20]	U~0.2*[1,Sum(processing time(A))]	U~[1,10]
	B	U~[1,30]	U~0.4*[1,Sum(processing time(B))]	
	C	U~[1,50]	U~0.49*[1,Sum(processing time(C))]	
5*500	A	U~[1,20]	U~0.2*[1,Sum(processing time(A))]	U~[1,10]
	B	U~[1,30]	U~0.4*[1,Sum(processing time(B))]	
	C	U~[1,50]	U~0.49*[1,Sum(processing time(C))]	

TABLE. V. Procedure of creating due date in each sample problem

Type	$\bar{B}$	$\bar{P}$	**P**	**date due**
A	$\frac{\Sigma sum(size(A))}{0.8 * S}$	$\frac{\Sigma sum(Processing(A))}{n}$	(m+$\bar{B}$(A)-1)*P(A)	U~[0,P(A)*(0.8+$\frac{o.3}{n}$]
B	$\frac{\Sigma sum(size(B))}{0.8 * S}$	$\frac{\Sigma sum(Processing(B))}{n}$	(m+$\bar{B}$(B)-1)*P(B)	U~[0,P(B)*(0.8+$\frac{o.3}{n}$]
C	$\frac{\Sigma sum(size(C))}{0.8 * S}$	$\frac{\Sigma sum(Processing(C))}{n}$	(m+$\bar{B}$(C)-1)*P(C)	U~[0,P(C)*(0.8+$\frac{o.3}{n}$]

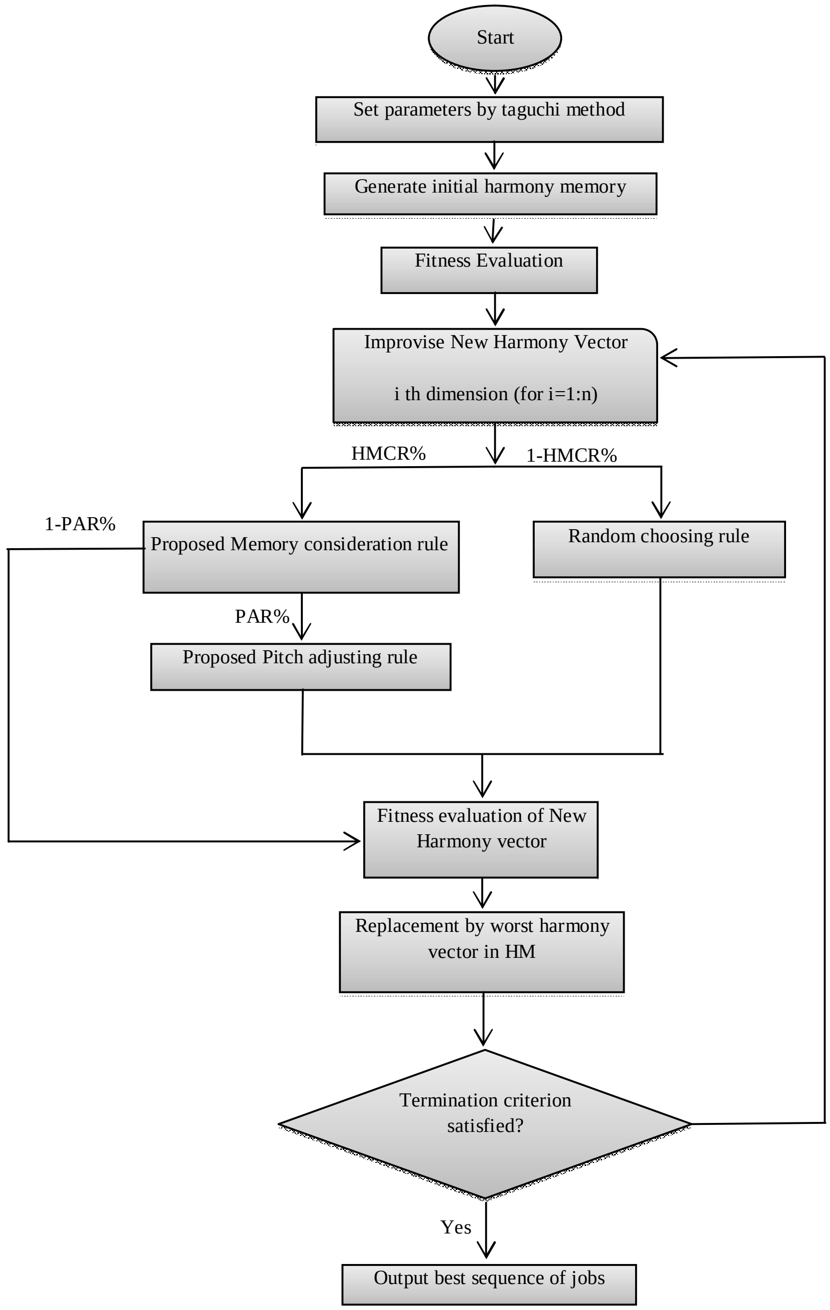

Fig 5. The flowchart of the proposed harmony search algorithm (TNHS)

B. Design of Experiments

The design of experiment methods were first devised by Fisher in England in the 1920s, in relation to agricultural systems. He was seeking to know how much sunlight, rain, chemical fertilizer and water was needed to produce the best product. He understood that experiments providing procedure may create errors in the analyses applied on the agricultural systems data. He defined three important principles in the field of experimental design: randomization, replication and blocking. Fisher introduced statistical thinking and principles such as using the concept of factorial designs and analysis of variance regularly in the design of experiments (Montgomery, 2012). An optimum experiment design is one that provides the data and information required for analysis and achievement of the optimal conditions, through the least experiments.

- *Adjustment of Taguchi parameter*

The Taguchi method is an experimental analysis technique that uses an orthogonal array for conducting experiments, based on which we can predict the effect of optimum factors and levels of experimental and laboratorial studies by a specified number of experiments. Dr. Genichi Taguchi, the head assessor of quality in a Japanese experimental design company, applied some research in his country in the 1940s, but his results were not taken up. After using his techniques for the reduction of cost and increasing the quality in different countries, he innovated an experimental design method in the USA in the mid-1980s that is today known as the robust parameter design method. Taguchi believed that the best way to improve and establish design quality lies in the product itself.

The design method of Taguchi experiments, requiring fewer experiments, leads to a clear saving of cost and time and provides the data and information required for performing the analysis and achieving the optimum conditions. This advantage has caused it to be considered by many researchers, within recent years, for the adjustment of parameters required for their proposed algorithms (Chan, Bhagwat, & Wadhwa, 2007; B.-W. Cheng & Chang, 2007; Naderi, Ghomi, & Aminnayeri, 2010; Naderi, Zandieh, & Roshanaei, 2009). In this paper, similar to the above researches, the best parameters and operators are obtained by means of this method for the implementation of algorithms.

- *Selecting the appropriate orthogonal array*

In this paper, we describe the Taguchi method's procedure and selection of the appropriate orthogonal array. In addition, the effect of the Taguchi method on the cost and time saving in comparison with full factorial design of experiment is explained. In the studied problem, 10 sample problems with different numbers of jobs and machines have been designed and in each one three different problems were devised considering the ready time, size of job and job processing times, and each experiment was replicated three times independently for greater reliability.

TABLE VI. Factor and levels of each factor of model solving algorithm

TNHS		**GBHS**		**HIS**		**HS**	
HMS	HMS(1) : 5 HMS(2) : 10 HMS(3) : 15	*HMS*	HMS(1) : 5 HMS(2) : 10 HMS(3) : 15	*HMS*	HMS(1) : 5 HMS(2) : 10 HMS(3) : 15	*HMS*	HMS(1) : 5 HMS(2) : 10 HMS(3) : 15
$HMCR_{max}$	$HMCR_{max}$(1) : 80% $HMCR_{max}$(2) : 99%	PAR_{max}	PAR_{max}(1) :50% PAR_{max} (2) :70% PAR_{max} (3) :90%	BW_{max}	BW_{max}(1) : 0.5 BW_{max}(2) : 0.99	*BW*	BW(1) : 0.2 BW(2) : 0.5 BW(3) : 0.99
PAR_{max}	PAR_{max}(1) :50% PAR_{max} (2) :90%	*HMCR*	HMCR(1) : 30% HMCR(2) : 60% HMCR(3) : 98%	PAR_{max}	PAR_{max}(1) :50% PAR_{max} (2) :70% PAR_{max} (3) :90%	*PAR*	PAR(1) : 10% PAR(2) :50% PAR(3) : 90%
				PAR_{min}	PAR_{min}(1) :20% PAR_{min}(2) :40%		
$HMCR_{min}$	$HMCR_{min}$(1) : 20% $HMCR_{min}$(2) : 50%	PAR_{min}	PAR_{min}(1) : 0 PAR_{min}(2) : 20% PAR_{min}(3) : 49%	*HMCR*	HMCR(1) : 50% HMCR(2) : 80% HMCR(3) : 99%	*HMCR*	HMCR(1) : 50% HMCR(2) : 80% HMCR(3) : 99%
PAR_{min}	PAR_{min}(1) : 20% PAR_{min}(2) : 40%			BW_{min}	BW_{min}(1) : 0.2 BW_{min}(2) : 0.4		

- *Adjustment of factor and levels of each factor of model solving algorithm*

In order to solve the problem, four metaheuristic algorithms including basic harmony search, improved harmony search, global best harmony algorithm and hybrid Taguchi and novel global best harmony search were compared. Appropriate parameters were selected by means of preliminary tests for each of these algorithms, and the best composition of parameters and operators was selected by means of the Taguchi method.

In the hybrid Taguchi and novel global harmony search algorithm, there are four two-level factors (minimum pitch adjustment rate, maximum pitch adjustment rate, minimum algorithm memory consideration rate and maximum algorithm consideration rate) and a three-level factor (harmony memory size). The factors and their levels are shown in Table 6. To implement the proposed algorithm for solving the problem, altogether $3\times2^4\times90\times5$ (i.e. 21600) experiments by the full factorial method are required. Considering the significance of cost and time reduction in the implementation of the algorithms, and particularly in scheduling problems, the implementation of this approach of experiments design is not economically viable. But the Taguchi fractional factorial designs effectively bring cost and time savings. In practice, the standardized and very simple Taguchi orthogonal array designs, with fewer experiments, have had the maximum application in estimating the optimum point and factors effects. In the first step, to implement the Taguchi method and adopt the appropriate orthogonal array, the required degrees of freedom must be computed. In this problem, one degree of freedom for the total mean, two degrees of freedom for three-level factors and one degree of freedom for each two-level factor are required. Therefore, the total required degrees of freedom are equal to: $1 + (1\times4) + 2 = 7$.

Thus an array must be selected that includes at least seven lines. Considering the standard orthogonal Taguchi arrays, it is concluded that in orthogonal array L12, these conditions are applicable and, considering the factors level, array L12 is selected. In Table 7, the structure of array L12 has been provided for the TNHS algorithm. The total number of problem implementations by means of the Taguchi method will be 12×90=1080 times, whilst it was equal to 21600 implementations when using complete factorial design of experiment. This means that 21600–1080=20520 experiments have been saved in time and cost. The implementation procedure of the experiments begins firstly by means of preliminary tests, appropriate parameters and candidate levels for the proposed algorithm. Then, according to the method explained above, the appropriate orthogonal array must be selected in consideration of these factors (parameters and operators) for implementation. Finally, upon implementing each algorithm, the best factors can be determined.

In most of the previous research applied, the parameters and operators of the algorithms to which the algorithm proposed in this study is compared are defined by the user or derived from previous research and only the parameters and operators of the proposed algorithm are adjusted (Naderi et al., 2010; Naderi et al., 2009), whilst the quality of an algorithm’s answer and its optimum parameters depend strongly on the objective function and the problem used therein (Ruiz & Maroto, 2006). Accordingly, in this study, to equalize the conditions for the TNHS algorithm and the other algorithms, the Taguchi method and parameters adjustment are implemented on all algorithms. Table 7 shows the appropriate orthogonal arrays for the other solution algorithms.

VI. IMPLEMENTATION OF EXPERIMENTS

To implement the experiments, all the algorithms used in this study were programmed by MATLAB software on a PC with a microprocessor core i5 GHz 2.27 and 4.00 GB RAM.

A. stopping criteria

In order to provide equal conditions for all algorithms, a stopping criterion has been considered and the algorithms’ stopping time is m×n×0.1. This factor is dependent on the problem size, meaning the number of jobs (n) and number of batch processors (m); upon increasing n and m, the stopping time is also increased.

TABLE VII. Orthogonal array for all algorithms

IHS algorithm

Trial	A	B	C	D	E	F
1	0	0	0	0	0	0
2	0	0	1	1	1	0
3	0	1	0	2	1	0
4	0	1	0	0	1	1
5	0	2	1	2	1	0
6	0	2	1	2	1	0
7	1	0	0	2	1	0
8	1	0	0	0	1	1
9	1	1	1	1	1	1
10	1	1	1	2	0	0
11	1	2	0	1	0	0
12	1	2	1	0	0	1
13	2	0	1	2	0	1
14	2	0	0	1	1	0
15	2	1	0	1	0	0
16	2	1	1	0	0	0
17	2	2	0	0	1	1
18	2	2	1	2	1	1

Trial	A	B	C	D	E
1	0	0	0	0	0
2	0	0	1	1	1
3	0	0	1	1	2
4	0	1	0	0	2
5	0	1	0	1	0
6	0	1	1	0	1
7	1	0	0	0	1
8	1	0	0	1	2
9	1	0	1	0	0
10	1	1	0	1	1
11	1	1	1	0	2
12	1	1	1	1	0

TNHS algorithm

HS and GBHS algorithm

Trial	A	B	C	D
1	0	0	0	0
2	0	1	2	1
3	0	2	1	2
4	1	0	2	2
5	1	1	1	0
6	1	2	0	1
7	2	0	1	1
8	2	1	0	2
9	2	2	2	0

A. Selection of optimum factors of model solving algorithms

In each experiment implementation, the obtained objective function must be converted according to the Taguchi method and proportional to the signal to noise ratio, and analysis is provided according to its variations, whereas the objective of each implementation is to minimize the objective function, so the signal to noise type is selected as per relation (3).

relation 3
$$S/N_s = -10\ \log\left(\frac{1}{n}\sum_{i=1}^{n} y_i^2\right)$$

In the Taguchi method, the S/N ratio indicates the ratio variable, and the objective function in each implementation is converted to this ratio in order that the decision is made based on it. In this study, considering the selected S/N ratios, proportional to the nature of this study's problems, the lowest S/N ratio for each factor in each algorithm is selected as the optimum factor. All the results obtained from the total model solution algorithm implementations using the Taguchi design of experiments for the adjustment of parameters are shown as S/N ratios in Figures 6–9.

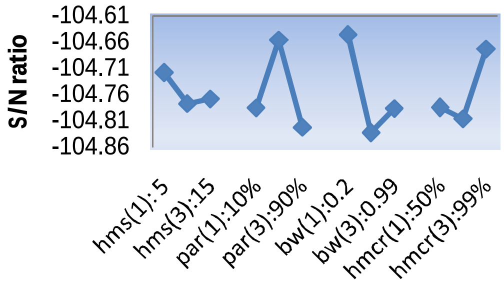

Fig. 6. Mean S/N ratio plot for HS

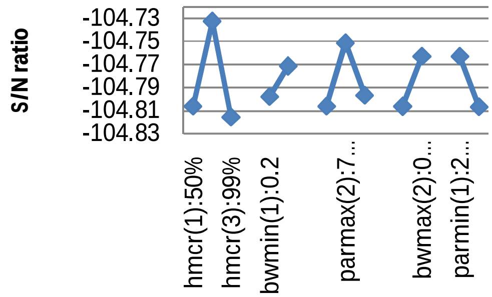

Fig. 7. Mean S/N ratio plot for IHS

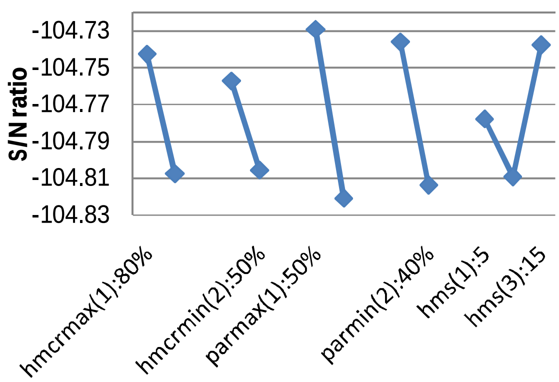

Fig. 8. Mean S/N ratio plot for TNHS

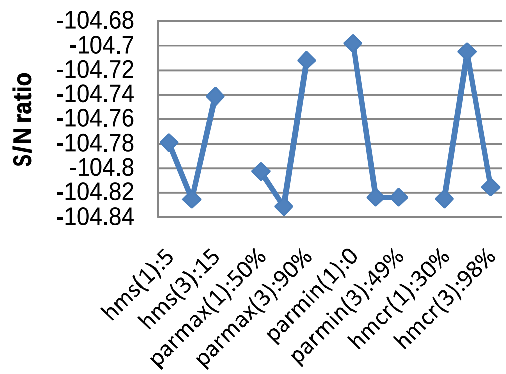

Fig. 9. Mean S/N ratio plot for GBHS

TABLE VIII. The optimum levels of parameters and operators

HS		**HIS**		**GBHS**		**TNHS**	
HMS	5	*HMS*	10	*HMS*	15	*HMS*	15
BW	0.2	BW_{max}	0.99	PAR_{max}	0.9	$HMCR_{max}$	0.8
PAR	0.5	PAR_{max}	0.7	*HMCR*	0.6	PAR_{max}	0.5
HMCR	0.99	*HMCR*	0.8	PAR_{min}	0	$HMCR_{min}$	0.2
		BW_{min}	0.4			PAR_{min}	0.2

As per the above figures, Table 8 shows the best factors for final implementation of the algorithms for model solving.

B. Computational results

The optimum parameters and operators of each algorithm which have been obtained by the Taguchi method are defined on the solution algorithms and the results of the above algorithm implementations are presented for the defined sample problems. In the sample problems devised for implementing the experiments, through changing problem dimensions such as the number of jobs and machines, due to the different and unequal scales of the objective function values, the relative percentage deviation (RPD) rate has been used to compare the algorithms (Sahraeian, Samaei, & Rastgar, 2014).

relation 4
$$RPD = \frac{ALG_{sol} - Min_{sol}}{Min_{sol}} \times 100$$

ALG_{sol} is the algorithm answer and Min_{sol} the minimum value of answers. In this ratio, when the RPD is lower, the answer quality and algorithm performance are better. In the results section, the performance of the algorithms is compared considering the problem size, which is varied based on the increased number of jobs and machines. After summarizing the RPD results, the performance of the algorithms is shown in diagram and table form.

C. Results of algorithms implementation

In the previous sections, we explained the proposed algorithms for solving the model. The RPD results obtained from the algorithm implementations are provided in Table 9. Table 9 shows the quality of the TNHS algorithm and basic harmony search (HS), with relative deviation rate means of 0.05036 and 0.04228 respectively, compared to the other algorithms. In the comparison of these two algorithms, it is observed that TNHS performed better. A diagram of the results summarized in Table 9 is shown in Figure 10. Furthermore, in order to evaluate the robustness of the algorithms in different situations, the means plots for the interaction between the different algorithms in numbers of processors and numbers of jobs are shown in Figures 11 and 12 respectively.

These cases show the high quality of the performance of the proposed harmony search algorithm in most indexes, and we can state that this algorithm, compared with the other algorithms, performs significantly better in solving such problems.

TABLE IX. Average relative percentage deviation for the solution methods

m×n	**HS**	**HIS**	**GBHS**	**TNHS**
3*10	0.048853586	0.042523912	**0.035508417**	0.037716399
3*20	0.071072375	0.07472343	0.067715736	**0.061150932**
3*50	0.095728311	0.084826867	**0.075068571**	0.080038965
3*100	0.137143133	0.095244469	**0.090260471**	0.094673713
3*200	**0.056884084**	0.059123798	0.059672232	0.064972284
5*40	0.017396947	0.040558163	0.03974089	**0.016239187**
5*80	0.020697891	0.039963149	0.041322198	**0.020113054**
5*120	**0.023549391**	0.043335106	0.044857978	0.023978817
5*200	0.021437507	0.046582531	0.054318456	**0.013772349**
5*500	0.010847835	0.023605891	0.023457069	**0.010218361**
Average	**0.050361106**	**0.055048732**	**0.053192202**	**0.042287406**

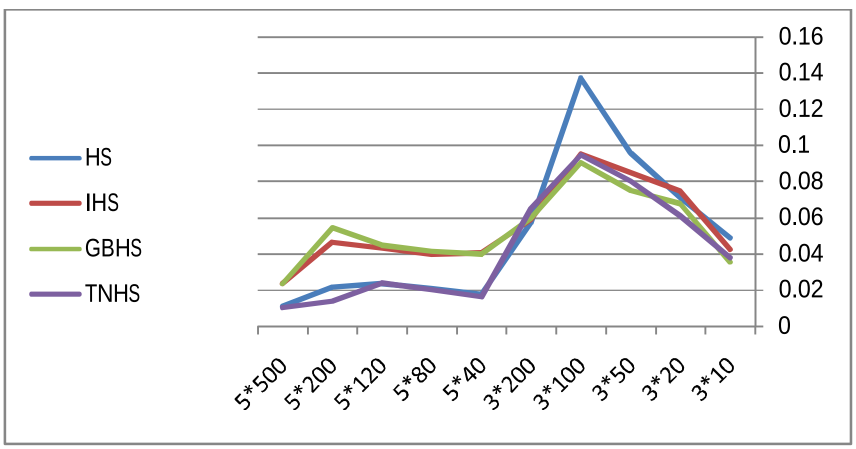

Fig. 10. Means plot diagram for the interaction between algorithms and size of problems

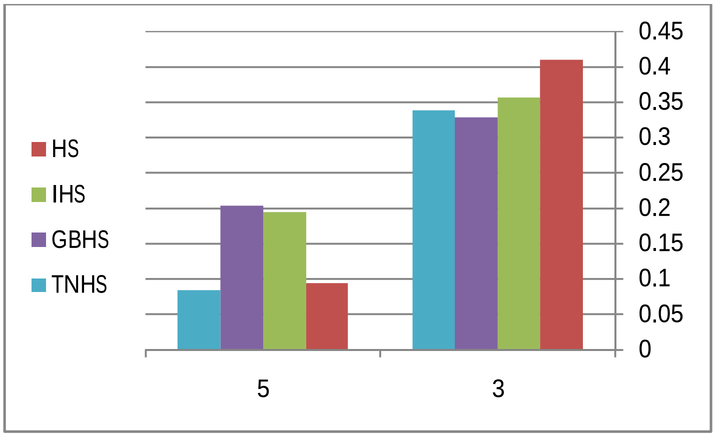

Fig. 11. Means plot diagram for the interaction between algorithms and number of batch processors

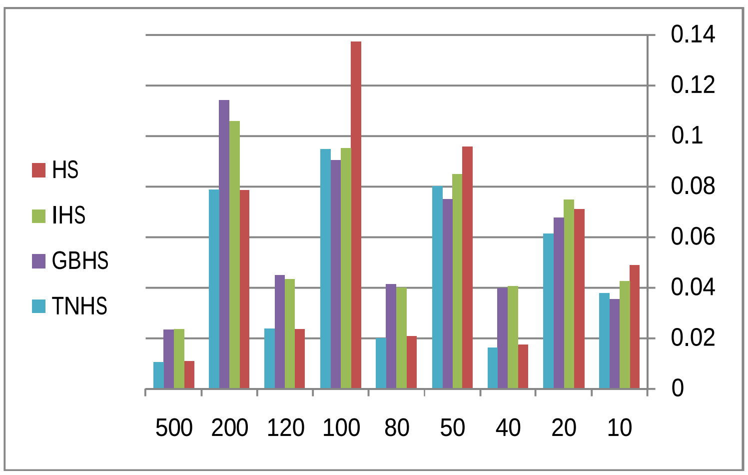

Fig. 12. Means plot diagram for the interaction between algorithms type and number of jobs

As can be seen in Figure 10, the TNHS and HS exhibit robust performance even when the size of the problem increases. They outperform the other algorithms on large size problems.

As per Figures 11 and 12, there is a clear trend that shows that when the number of batch processors and number of jobs increase, better performance of the TNHS is obtained. Furthermore, the behaviour of the HS is very close to that of TNHS in problems with large numbers of jobs and machines.

VII. CONCLUSION

This paper provided a new model for minimizing two objectives simultaneously, that is, the time function including makespan and the cost function including earliness and tardiness penalties of jobs, which is applicable in most industries. To solve the problem, although the problem is NP-hard, to obtain appropriate answers within a reasonable time using metaheuristic methods is economical. In this paper, the harmony search algorithm was used based on the structure of the global best harmony search method. In addition, a novel dynamic adjustment for variations of the PAR and HMCR parameters was presented upon increasing the algorithm iterations, and, according to exploration and exploitation concepts, it is implemented from the beginning of algorithm implementation until its stopping criterion. One of the disadvantages of the harmony search method that was specified in different research is that the algorithm has a high probability of becoming trapped in the local optimum, and has no appropriate local search. Consequently the dispersal must be increased at any time that it is situated in the local trap using the restart phase technique. Also, since the final results of each algorithm are dependent on its initial parameters, the Taguchi design of experiment was used, and the best composition of parameters in each algorithm was specified. The results obtained in the previous section indicate that the TNHS algorithm in most samples provides better results. Extension of the novel metaheuristic algorithms and the model in a multi-station mode such as a hybrid flowshop may be taken into consideration by researchers interested in this field.

REFERENCES

Bank, Jan, & Werner, Frank. (2001). Heuristic algorithms for unrelated parallel machine scheduling with a common due date, release dates, and linear earliness and tardiness penalties. *Mathematical and Computer Modelling, 33*(4), 363-383.

Cakici, Eray, Mason, Scott J, Fowler, John W, & Geismar, H Neil. (2013). Batch scheduling on parallel machines with dynamic job arrivals and incompatible job families. *International Journal of Production Research, 51*(8), 2462-2477.

Chan, Felix TS, Bhagwat, Rajat, & Wadhwa, S. (2007). Flexibility performance: Taguchi's method study of physical system and operating control parameters of FMS. *Robotics and Computer-Integrated Manufacturing, 23*(1), 25-37.

Chang, P-Y, Damodaran*, P, & Melouk, S. (2004). Minimizing makespan on parallel batch processing machines. *International Journal of Production Research, 42*(19), 4211-4220.

Cheng, Bayi, Wang, Qi, Yang, Shanlin, & Hu, Xiaoxuan. (2013). An improved ant colony optimization for scheduling identical parallel batching machines with arbitrary job sizes. *Applied Soft Computing, 13*(2), 765-772.

Cheng, Bor-Wen, & Chang, Chun-Lang. (2007). A study on flowshop scheduling problem combining Taguchi experimental design and genetic algorithm. *Expert Systems with Applications, 32*(2), 415-421.

Chung, SH, Tai, YT, & Pearn, WL. (2009). Minimising makespan on parallel batch processing machines with non-identical ready time and arbitrary job sizes. *International Journal of Production Research, 47*(18), 5109-5128.

Damodaran, Purushothaman, Diyadawagamage, Don Asanka, Ghrayeb, Omar, & Vélez-Gallego, Mario C. (2012). A particle swarm optimization algorithm for minimizing makespan of nonidentical parallel batch processing machines. *The International Journal of Advanced Manufacturing Technology, 58*(9-12), 1131-1140.

Damodaran, Purushothaman, Hirani, Neal S, & Velez-Gallego, Mario C. (2009). Scheduling identical parallel batch processing machines to minimise makespan using genetic algorithms. *European Journal of Industrial Engineering, 3*(2), 187-206.

Damodaran, Purushothaman, & Velez-Gallego, Mario C. (2010). Heuristics for makespan minimization on parallel batch processing machines with unequal job ready times. *The International Journal of Advanced Manufacturing Technology, 49*(9-12), 1119-1128.

Damodaran, Purushothaman, Vélez-Gallego, Mario C, & Maya, Jairo. (2011). A GRASP approach for makespan minimization on parallel batch processing machines. *Journal of Intelligent Manufacturing, 22*(5), 767-777.

Feng, Qi, Yuan, Jinjiang, Liu, Hailing, & He, Cheng. (2013). A note on two-agent scheduling on an unbounded parallel-batching machine with makespan and maximum lateness objectives. *Applied Mathematical Modelling, 37*(10), 7071-7076.

Geem, Zong Woo, Kim, Joong Hoon, & Loganathan, GV. (2001). A new heuristic optimization algorithm: harmony search. *Simulation, 76*(2), 60-68.

Jia, Zhao-hong, & Leung, Joseph Y-T. (2015). A meta-heuristic to minimize makespan for parallel batch machines with arbitrary job sizes. *European Journal of Operational Research, 240*(3), 649-665.

Kashan, Ali Husseinzadeh, Karimi, Behrooz, & Jenabi, Masoud. (2008). A hybrid genetic heuristic for scheduling parallel batch processing machines with arbitrary job sizes. *Computers & Operations Research, 35*(4), 1084-1098.

Kedad-Sidhoum, Safia, Solis, Yasmin Rios, & Sourd, Francis. (2008). Lower bounds for the earliness–tardiness scheduling problem on parallel machines with distinct due dates. *European Journal of Operational Research, 189*(3), 1305-1316.

Lee, C-Y. (1999). Minimizing makespan on a single batch processing machine with dynamic job arrivals. *International Journal of Production Research, 37*(1), 219-236.

Lee, Kang Seok, Geem, Zong Woo, Lee, Sang-ho, & Bae, Kyu-woong. (2005). The harmony search heuristic algorithm for discrete structural optimization. *Engineering Optimization, 37*(7), 663-684.

Li, XiaoLin, Huang, YanLi, Tan, Qi, & Chen, HuaPing. (2013). Scheduling unrelated parallel batch processing machines with non-identical job sizes. *Computers & Operations Research, 40*(12), 2983-2990.

Mahdavi, M, Fesanghary, Mohammad, & Damangir, E. (2007). An improved harmony search algorithm for solving optimization problems. *Applied mathematics and computation, 188*(2), 1567-1579.

Mönch, Lars, & Unbehaun, Robert. (2007). Decomposition heuristics for minimizing earliness–tardiness on parallel burn-in ovens with a common due date. *Computers & operations research, 34*(11), 3380-3396.

Montgomery, Douglas C. (2012). *Design and analysis of experiments* (Vol. 7).

Moslehi, G, Mirzaee, M, Vasei, M, Modarres, M, & Azaron, A. (2009). Two-machine flow shop scheduling to minimize the sum of maximum earliness and tardiness. *International Journal of Production Economics, 122*(2), 763-773.

Naderi, B, Ghomi, SMT, & Aminnayeri, M. (2010). A high performing metaheuristic for job shop scheduling with sequence-dependent setup times. *Applied Soft Computing, 10*(3), 703-710.

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.

Omran, Mahamed GH, & Mahdavi, Mehrdad. (2008). Global-best harmony search. *Applied Mathematics and Computation, 198*(2), 643-656.

Pan, Quan-Ke, Suganthan, PN, Liang, JJ, & Tasgetiren, M Fatih. (2010). A local-best harmony search algorithm with dynamic subpopulations. *Engineering Optimization, 42*(2), 101-117.

Pan, Quan-Ke, Suganthan, Ponnuthurai N, Tasgetiren, M Fatih, & Liang, Jing J. (2010). A self-adaptive global best harmony search algorithm for continuous optimization problems. *Applied Mathematics and Computation, 216*(3), 830-848.

Pinedo, Michael. (2012). *Scheduling: theory, algorithms, and systems*: Springer.

Ruiz, Rubén, & Maroto, Concepción. (2006). A genetic algorithm for hybrid flowshops with sequence dependent setup times and machine eligibility. *European Journal of Operational Research, 169*(3), 781-800.

Sahraeian, R, Samaei, F, & Rastgar, I. (2014). Minimizing the makespan on parallel batch scheduling with stochastic times *Annals of Industrial Engineering 2012* (pp. 209-216): Springer.

Shao, Hao, Chen, Hua-Ping, Huang, George Q, Xu, Rui, Cheng, Ba-yi, Wang, Shuan-shi, & Liu, Bo-wen. (2008). *Minimizing makespan for parallel batch processing machines with non-identical job sizes using neural nets approach.* Paper presented at the Industrial Electronics and Applications, 2008. ICIEA 2008. 3rd IEEE Conference on.

Taherinejad, Nima. (2009). *Highly reliable harmony search algorithm*. Paper presented at the Circuit Theory and Design, 2009. ECCTD 2009. European Conference on.

Toksarı, M Duran, & Güner, Ertan. (2009). Parallel machine earliness/tardiness scheduling problem under the effects of position based learning and linear/nonlinear deterioration. *Computers & Operations Research, 36*(8), 2394-2417.

Velez Gallego, Mario Cesar, & Adviser-Damodaran, Purushothaman. (2009). Algorithms for scheduling parallel batch processing machines with non-identical job ready times.

Wang, Hui-Mei, & Chou, Fuh-Der. (2010). Solving the parallel batch-processing machines with different release times, job sizes, and capacity limits by metaheuristics. *Expert Systems with Applications, 37*(2), 1510-1521.

Wang, Jun-Qiang, & Leung, Joseph Y-T. (2014). Scheduling jobs with equal-processing-time on parallel machines with non-identical capacities to minimize makespan. *International Journal of Production Economics, 156*, 325-331.

Xu, Shubin, & Bean, James C. (2007). *A genetic algorithm for scheduling parallel non-identical batch processing machines.* Paper presented at the Computational Intelligence in Scheduling, 2007. SCIS'07. IEEE Symposium on.

Xu, Shubin, & Bean, James C. (2015). Scheduling parallel-machine batch operations to maximize on-time delivery performance. *Journal of Scheduling*, 1-18.

Yilmaz Eroglu, Duygu, Ozmutlu, H Cenk, & Ozmutlu, Seda. (2014). Genetic algorithm with local search for the unrelated parallel machine scheduling problem with sequence-dependent set-up times. *International Journal of Production Research, 52*(19), 5841-5856.

Zou, Dexuan, Gao, Liqun, Li, Steven, & Wu, Jianhua. (2011). Solving 0–1 knapsack problem by a novel global harmony search algorithm. *Applied Soft Computing, 11*(2), 1556-1564.