Research Article

Bi-objective Hybrid Flow Shop Scheduling with Family Setup Times Using Hybrid Genetic and Migrating Birds Optimization Algorithms

Wanida Laoraksakiat
Department of Industrial Engineering, Faculty of Engineering, King Mongkut's University of Technology North Bangkok, Bangkok, Thailand

Krisada Asawarungsaengkul*
Operations Research and Engineering Management Research Center, Department of Industrial Engineering, Faculty of Engineering, King Mongkut's University of Technology North Bangkok, Bangkok, Thailand

* Corresponding author. E-mail: krisada.a@eng.kmutnb.ac.th DOI: 10.14416/j.asep.2019.10.001
Received: 27 May 2019; Revised: 24 June 2019; Accepted: 28 June 2019; Published online: 15 October 2019

Abstract
This paper presents a hybrid metaheuristic algorithm to solve the hybrid flow shop scheduling problem (HFSP) with family setup times. Many conditions of HFSP have been extensively studied in recently years and metaheuristics and local search algorithms have also been developed to yield better solutions for multi-objective HFSP. HFSP in this work is based on a harddisk drive manufacturer. An effective NSGA-II integrated with migrating birds optimization (MBO) called MBNSGA-II is proposed to improve the quality of solutions for bi-objective HFSP. Makespan and total tardiness time are the objectives of this HFSP. MBO is added to mutation operation of genetic algorithm to improve the Pareto front. Next, various sizes of benchmark problem are utilized to evaluate the performance of NSGA-II and MBNSGA-II. The comparisons of two algorithms consisting of NSGA-II and MBNSGA-II are provided by using the numerical examples. It is obvious the Pareto fronts obtained from MBNSGA-II are adjacent to the approximated true Pareto front. In terms of inverted generational distance (IGD) which is the index of convergence and diversity of the solution set, the performance of proposed MBNSGA-II outperforms NSGA-II.

Keywords: Hybrid flow shop scheduling, Family setup times, NSGA-II, Migrating birds optimization, Hybrid algorithm

1 Introduction

This paper proposed a hybrid algorithm of migrating birds optimization and NSGA-II (MBNSGA-II) to handle the hybrid flow shop scheduling problem (HFSP) with family setup times. Normally, a job does not require the setup time if it follows a job belong to the same family [1]. However, there is a slightly difference for family setup time in the production line of head gimbal assembly (HGA). A magnetic disk requires two type HGAs including up and down type that belong to the same family. Thus, a group of order including order of up type and order of down type is typically issued to HGA production line. Change over between two jobs within the same family needs a quick setup time. Major setup time is applied when changing between two families.

Some production lines are comprised of several types of machine integrated into a flow shop assembly line. Due to unbalanced capacity of each machine, some processes or stages require more than one machine. The general definition for HFSP is that at

Please cite this article as: W. Laoraksakiat and K. Asawarungsaengkul, "Bi-objective hybrid flow shop scheduling with family setup times using hybrid genetic and migrating birds optimization algorithms," *Applied Science and Engineering Progress*, vol. 14, no. 1, pp. 19–30, Jan.–Mar. 2021, doi: 10.14416/j.asep.2019.10.001.

least one stage has identical machines. This HFSP has been applied in many manufacturing such as textile, semiconductors, assembly process of IC, Hard disk drive, assembly and electronic assembly process. Since these manufacturing require specific machine for each process, the standard machines are used as system integration. That is why machines having low capacity are required more than one machine. In production planning, the scheduling of job orders is our interest. In real world problems, the performances of scheduling are measured by at least two key performance indices. Typically, the makespan and total tardiness time are considered as the key objectives of scheduling.

HFSP is one of the NP-hard problems which is hard to solve [2]. The multi-objective of HFSP has been studied with various conditions. The group scheduling in HFSP with bi-objective was discussed by Bozorgirad *et al*. [2]. Behnamian and Ghomi [3] proposed a hybrid metaheuristic for HFSP considering machines with different speeds and sequence-dependent setup times. The minimization of makespan and total resource allocation costs were the objectives of this research. The Non-Dominated Sorting Genetic Algorithm II (NSGA-II) for multi-objective of reentrant HFSP was done by [4]. Others multi-objective of reentrant HFSP can be found in [5], [6]. The bi-objective HFSP having sequence-dependent setup times and limited buffer spaces was studied by [7]. The multi-objective HFSP with sequence-dependent setup times was presented in [2], [8]. The objectives for HFSP which were found in these papers [1]–[8] including minimization of makespan, total tardiness of jobs, total resource allocation cost, total weighted completion time, total weighted tardiness, and maximum tardiness.

Since HFSP and multi-objective HFSP can be classified as the NP-hard problem, the metaheuristic algorithms and hybrid metaheuristic algorithms were developed to determine the optimal or near optimal with reasonable computational time. The Non-Dominated Sorting Genetic Algorithm II (NSGA-II) was utilized to solve the multi-objective HFSP in [4], [7], [9]. The Pareto genetic algorithm adopting the Minkowski distance-based crossover operator and additional local search strategies was used to determine the solutions for multi-objective HFSP [4]. The results revealed that the proposed algorithm was superior to NSGA-II. Abyaneh and Zandieh [7] utilized the Sub-population genetic algorithm II (SPGA-II), NSGA-II and the local search algorithm to search for the best solutions for bi-objective HFSP under sequence-dependent setup times and limited buffers. The modified versions of both SPGA-II and NSGA-II could yield the better solutions. NSGA-II based memetic algorithm was also proposed by [9] in order to solve multi-objective parallel flow shop scheduling problem.

Other algorithms were also used in determining better quality of solution for multi-objective HFSP. Metaheuristic algorithms such as Tabu search [2], Lorenz non-dominated sorting genetic algorithm (L-NSGA) [5], [10], Strength pareto evolutionary algorithm 2 (SPEA2) [5], Iterated pareto greedy (IPG) algorithm [6], Novel neighborhood search (NSG) [11] etc. were tested with multi-objective HFSP. The HFSP was solved by discrete artificial bee colony algorithm [12] and tabu search heuristic algorithm [13]. A coevolutionary algorithm [14] utilizing global agents and local agents was applied to obtain the solutions for multi-objective HFSP. The performance metrics presented in [14] including: 1) dominating ratios (DR), 2) mean ideal distance (MID), 3) spread of non-dominated solution (SNS), and the best of each objective function (BEO). These metrics were used to compare the performance of proposed algorithms. Other metaheuristics which are ant colony optimization algorithm [15] and hybrid algorithm between artificial bee colony and tabu search were also developed to solve HFSP [16].

Recently, a new metaheuristic algorithm namely migrating bird optimization (MBO) algorithm was introduced by Duman *et al*. [17] which is able to solve the problems in a variety of areas effectively. This algorithm was inspired from the v-formation of the bird flocks to fly the long distances in V shape. The MBO has attracted the attention of researchers to deal with many different scheduling problems [18]–[23]. Pan and Dong [18] developed an improvement version of MBO to enhance the HFSP that tried to minimize the total flow time. Zhang *et al*. [19] applied the modified MBO to solve the single objective hybrid flow shop with lot-steaming to minimize the total flow time. The algorithm was also increased efficiency by using various neighborhood searches. Meng *et al*. [20] also studied the flow shop with lot-streaming to determine the makespan minimization. This research utilized MBO with the harmony search to create the effective solutions. A tabu list and neighborhood search and other

mechanisms were utilized in the permutation flow shop scheduling under the condition of sequence dependent setup times of which its objective is to minimize the makespan [21]. Gao and Pan [22] developed MBO with the shuffled multi-swarm micro-MBOs in order to solve the multi-resource-constrainted flexible job shop scheduling, effectively. In addition, MBO based on Pareto dominance relationship was applied for the multi-objective hybrid flow shop rescheduling problem [23]. The proposed algorithm was efficiently used to solve for three minimization objectives which were the makespan, the job start deviation, and the change in machine assignment after job cancellation.

It is obvious that the hybrid metaheuristics have been interested by many researchers due to their capability in solving the complex hybrid flow shop scheduling which is the NP-hard problem. Thus, this paper adopts a strong characteristic of Migrating birds optimization to the framework of Non-dominated sorting genetic algorithm II (NSGA-II) called MBNSGA-II. The remaining sessions are organized as following: 2) problem description, 3) NSGA-II and Migrating Birds Optimization Algorithm, 4) computational results and discussions, and 5) conclusions and future works.

2 Problem Description

This paper studies the process of head gimbal assembly (HGA) shown in Figure 1. The HGA is made of a slider and a suspension. HGA is utilized to read and write on both sides of the hard disk drive platter or magnetic disk. There are two types of HGA which are up and down type to read or write top and bottom of platter. Therefore, each customer usually orders both types, equally. The HGA process of case study consists of five stages, including the assembly process to attach the slider on the suspension, cleaning, cutting and bending process at the tail of HGA, the testing and the cleaning, and some stages have the identical parallel machines. The buffers, which are assumed the unlimited buffers are installed between stages. This production line can be classified to be the sequence-independent setup times with family setup times. A special characteristic for HFSP in this paper is that the setup time between two types is shorter than the setup time between families. This production line can be considered as the hybrid flow shop assembly line. The scheduling of production batches is a vital activity for production planning and

Figure 1: HGA assembly line with five processes or stages excluding buffers.

control. Therefore, the hybrid flow shop scheduling problem (HFSP) is our interest. Since HFSP is NP-hard problem [2], a development of effective hybrid metaheuristic algorithm is the main focus of this paper. Bi-objective for scheduling of hybrid flow shop are to minimize the makespan and the total tardiness time of jobs because the delay of jobs will make a big impact to the customer which is the head-stack operation.

The mathematical model from [24] is adopted and modified to represent the HFSP in this study. Bi-objective of this HFSP is to minimize the makespan and the total tardiness time of groups. Each group g comprises of two jobs including one job for up type and another job for down type. Two types of HGA are assigned to the same platter. Therefore, the tardiness time of each group is the maximum tardiness between two jobs within a group of order. The assumptions and constraints for this problem are shown as follows:

- All k jobs are available at the beginning of scheduling.
- All machines are not allowed to breakdown.
- The setup times is sequence-independent.
- Infinite buffers are installed between stages.
- Transportation time between stages is negligible.

The parameters and indices can be presented as below:

M Number of stages, ($i = 1,\ldots,M$)
J_i Number of parallel machines in stage i, ($j \in J_i$)
K Number of jobs, ($k = 1,\ldots,K$)
G Number of group orders, ($g = 1,\ldots,G$)
c_{ik} Completion time of job k in stage i
d_{ik} Departure time of job k from stage i
p_{ijk} Processing time of job k in stage i
L_k Due date of job k
S_{ikl} Setup time for changing from job k to job l at stage i (if job k and job l belong to the same family, it requires only quick setup time; otherwise, longer setup is required.)
Q Large value
TT Total tardiness time of jobs
C_{max} Maximize completion time
F_{gk} 1 if job k is in group g; 0 otherwise

The decision variables can be expressed as follows:

T_k Tardiness of each job k
x_{ijk} Binary decision 1 if job k is assigned to processor j in stage i; 0 otherwise
U_k Binary decision 1 if tardiness of job $k > 0$; 0 otherwise
R_{gk} Tardiness time of job k in group g
B_g Binary decision 1 if at least one of tardiness of job k in group $g > 0$; 0 otherwise
y_{ijkl} 1 if job k precedes job l on the processor j in stage i; 0 otherwise

The mathematical model is formulated as follows:

$$Minimize\ Z_1(x) = Minimize\ C_{MAX} = Minimize\ Max(c_{mk}) \quad (1)$$

$$Minimize\ Z_2(x) = Minimize\ TT = Minimize \sum_{g=1}^{G} B_g \quad (2)$$

Subject to:

$$c_{1k} \geq \sum_{j=1}^{J_i} p_{1k} x_{1jk} \qquad \forall k \quad (3)$$

$$c_{ik} - c_{(i-1)k} \geq \sum_{j=1}^{J_i} p_{ik} x_{ijk} \qquad \forall i,k; i > 1 \quad (4)$$

$$c_{ik} \leq d_{ik} \qquad \forall i,k; i < m \quad (5)$$

$$c_{mk} = d_{mk} \quad (6)$$

$$c_{ik} - \sum_{j=1}^{J_i} p_{ik} x_{ik} = d_{(i-1)k} \qquad \forall i,k; i > 1 \quad (7)$$

$$c_{ik} + Q(2 + y_{ijkl} - x_{ijk} - x_{ijl}) \geq d_{il} + (p_{ik} \times x_{ijk}) + S_{ilk} \qquad \forall i,j,k,l; k < l \quad (8)$$

$$c_{il} + Q(3 - y_{ijkl} - x_{ijk} - x_{ijl}) \geq d_{ik} + (p_{il} \times x_{ijl}) + S_{ikl} \qquad \forall i,j,k,l; k < l \quad (9)$$

$$c_{mk} \leq L_k + T_k \qquad \forall k \quad (10)$$

$$T_k \times F_{gk} \leq R_{gk} \qquad \forall g,k \quad (11)$$

$$c_{ik} \leq C_{max} \qquad \forall i,k \quad (12)$$

$$B_g = \underset{k=1}{\overset{K}{Max}} (R_{gk}) \qquad \forall g \quad (13)$$

The objective functions (1) and (2) are to minimize the maximum makespan and the total of maximum tardiness time. Constraints (3) and (4) show that job k is processed on the first machine and successively on all downstream machines. Constraints (5) and (6) ensure that each job k leaves the line after completing on the last machine. Constraint (7) indicates that each machine starts immediately after each job leaves from the precedence machine. Constraints (8) and (9) determine two parts allocated to the same machine cannot be processed simultaneously. Constraints (10) and (11) express the tardiness time. Constraint (12) indicates the maximize completion time from the latest completion time of jobs on the last machine. Constraints (13) defines the maximum tardiness time of group g.

3 NSGA-II and Migrating Birds Optimization Algorithm

This paper develops a hybrid algorithm to improve the performance of NSGA-II by integrating the migrating birds optimization into NSGA-II. NSGA-II and MBO procedures are described in the following subsections.

3.1 *Non-dominated sorting genetic algorithm-II (NSGA-II)*

Non-dominated sorting genetic algorithm II (NSGA-II) developed by [25] is an algorithm to deal with the multi-objective simultaneously. It is the metaheuristic based on the genetic algorithm to find the pareto optimal solutions. NSGA-II proposes three important procedures including the fast non-dominated sorting which is a procedure that sorts a population into different non-domination levels, the crowding distance that is a procedure to calculate the density of each solution, and the crowded-comparison that is the comparison operator between two solutions belonging to the same front.

3.2 *The migrating birds optimization algorithm (MBO)*

The original migrating birds optimization algorithm was developed by Duman *et al*. [7]. It was tested the performance in the quadratic assignment problem. MBO is a search algorithm inspired from the V- formation of the migrating birds. The method starts with the initial solutions or bird population. The best solution in the population becomes the leader bird and the other solutions become the follower birds. The leader bird

uses the neighborhood search to generate the neighbor solutions. The follower bird explores its neighbor solution by its solution and combines them with the best unused neighbor solutions from the predecessor bird as the V- formation. These steps are continued until a number of tours are reached. Then, the leader bird moves to the end of line and the next follower bird becomes the new leader. This algorithm is repeated until the stop criterion is met.

3.3 *The hybrid migrating bird NSGA-II (MBNSGA-II) for HFS scheduling problem*

The hybrid metaheuristic algorithm is proposed by combining migrating birds optimization and NSGA-II together. The proposed algorithm called hybrid migrating bird optimization NSGA-II (MBNSGA-II) tries to enhance the performance of NSGA-II and to obtain better quality of pareto optimal front for the bi-objective HFSP.

The procedure of MBNSGA-II for HFS problem shown in Figure 2 can be described as below:

Step 1. Chromosome representation and initial solutions: The scheduling is a technique to sequence the jobs within limited resources. Thus, the permutation representation is used for the sequence of jobs in each stage. For the machine assignment, the first unoccupied machine is assigned in the chromosome. The chromosome for this problem is depicted in Figure 3.

From Figure 3, there are four jobs for sequencing, two stages and two parallel machines in each stage. The sequence of jobs in the first stage and the second stage are 4, 3, 2, 1 and 3, 4, 2, 1, respectively. Because there is no job assigned to both machines at the beginning of scheduling, so job 4 and job 3 are performed by the machine 1 and machine 2 in stage 1, respectively. Job 2 is operated in the machine 1 that is the first available machine when comparing with the machine 2. Lastly, the job 1 is performed by machine 2.

Step 2. Fitness evaluation: The fitness of each chromosome is used in the selection operator to create the next population. The non-dominated level is assigned as the fitness of chromosome by the fast non-dominated sorting method. Thus, the best fitness is the lowest level that is the first Pareto front. Moreover, the crowding distance that is the density of solutions around the particular solution is also assigned to each solution. The large crowding distance shows better diversity in the Pareto front. The crowding distance of solutions

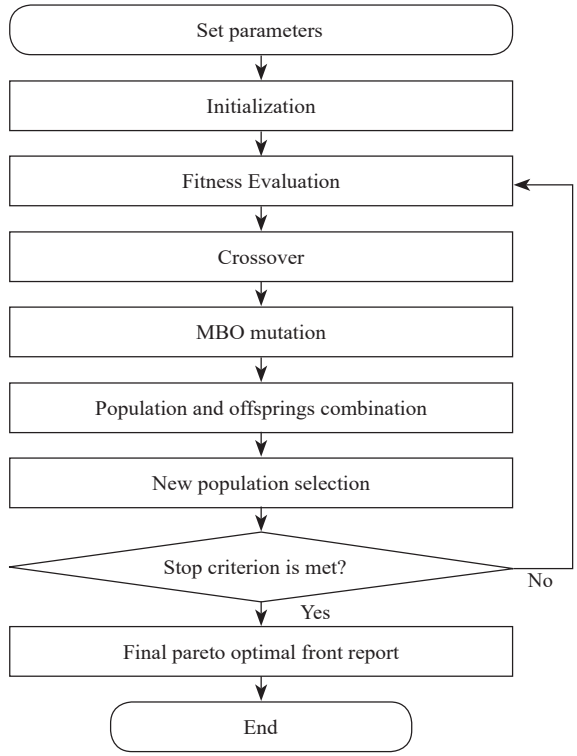

Figure 2: Flow of MBNSGA-II.

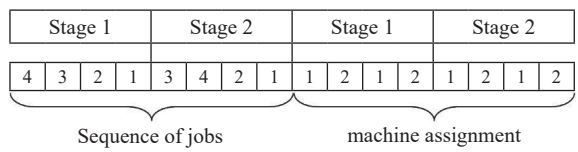

Figure 3: Chromosome representation.

on each Pareto front are calculated. The distance of extreme solutions is determined as infinity value. For the other solutions, their distance is calculated by the Equation (14) as follows:

$$I_i = I_i + (I_{i+1,m} - I_{i-1,m}) \tag{14}$$

where I_i is the distance of solution i and $I_{i,m}$ is the crowding distance of solution i in objective m.

Step 3. Selection operator: The tournament method is used to select the chromosomes for crossover and mutation operation. There are 4 chromosomes are selected randomly. The best chromosome in this group wins the tournament if its non-dominated level or its crowding distance is better than others.

Step 4. Crossover operator: Crossover operation is the method to generate a child by exchanging identical parts between two parents. The Partial-Mapped-Crossover (PMX) is employed for the sequence of jobs and the 2-point crossover is performed for the machine assignment vector.

Step 5. Mutation operator: After selecting the chromosomes to perform the mutation operation, MBO is applied for them to search the neighbor solutions. The swap or the insertion is performed to generate the neighbor solution at random between (0, 1). If the random number is not greater than 0.5, the swap method is used to generate the neighbor, otherwise, the neighbor is generated by the insertion. The step of MBO are repeated m tours to generate k–x neighbor solutions (except the leader bird) and shared x neighbor solutions with the next follower solution (bird) along the lines of V- formation. The current solution is replaced with the best neighbor solution if it cannot dominate the best neighbor solution. After the number of tours is reached, the leader bird is changed l times by the next follower bird. The steps of MBO are presented as Figure 4.

Step 6. Elitism and reproduction: The elitism is a procedure to keep the best fitness value and update it if the better fitness value is found in the next generation. However, the multi-objective cannot identify the best fitness as the single objective. Thus, the chromosomes with the first non-dominated level are copied to next generation after combining the parents and the offsprings.

After offsprings are created by crossover and mutation operation, the parents and the offsprings are combined. Then, the non-dominated levels are identified and the crowding distances are calculated. The new population is filled from the lowest non-dominated level to higher non-dominated level until equal or greater than the population size. If the number of chromosomes of new population are over the population size, the chromosomes having the lowest crowding distance in the last accepted non-dominated level will continuously be eliminated to preserve the population size.

Step 7. Stopping criteria: The algorithm will be terminated when the maximum number of generations is reached.

4 Results and Discussion

MBNSGA-II is evaluated by using the following data sets that are the combination of the number of jobs and number of stages (K, M). A number of jobs are 12, 18, 24, 32, 40, 48, 56, 64 and 90. A number of stages are 5 and 10 for 12 jobs to 56 jobs and 7 and 14 stages for 64 jobs and 90 jobs. The groups of order are 4, 6, 6, 8, 8, 6, 8, 8 and 10 and the number of families is 2. Refer to [26], the processing time of job in each stage is a random value at interval [20, 100]. The setup time for switching family is generated uniformly [5, 40] and the setup time for changing types is uniformly distributed of [1, 4]. The number of machines in each stage is a random integer at the interval [1, 4]. The due date of each job k can be generated from the approach of Karimi and Davoupour [27] in Equations (15) and (16) as follows:

Algorithm: MBO for mutation step	
Input:	n chromosomes for mutation, m, k, x, l
Output:	n offsprings
1.	If mod(n) = 0 // The number of chromosomes is even number.//
2.	Generate a new chromosome.
3.	Combine a new chromosome with n chromosomes.
4.	End if
5.	Set the best solution as the leader.
6.	Separate the others solutions equally to the left line and the right line as V-formation.
7.	for l = 1 to l do
8.	Improve the leader by local search to k neighbor solutions.
9.	Replace the leader with the best neighbor solutions if the leader cannot dominate the best neighbor
10.	solutions, otherwise, the leader is unchanged.
11.	for i = 1 to m do
12.	for j = 1 to p do
13.	Share x best unused neighbors from the precedence solution along the lines to the neighbor set of j.
14.	Generate k–x neighbors randomly.
15.	Combine x solutions with k–x neighbors in set j
16.	Evaluate the neighbor set j and replace the best solution of set j on the current
17.	solution cannot dominate the neighbor solution, otherwise, the current solution is unchanged.
18.	end for
19.	end for
20.	Replace the current leader with the next follew bird
21.	end for
22.	Update the non-dominated solutions

Figure 4: Pseudo Code of MBO for mutation step in MBNSGA-II.

$$P_k = \sum_{i=1}^{m} p_{ik} \tag{15}$$

$$d_k = P_k \times (1 + r \times 3) \tag{16}$$

where d_k is the due date of job k, r is the random number over interval [0, 1] and P_k is the sum of processing time on all stages.

4.1 *Performance metrics*

To evaluate the performance of each algorithm, the inverted generational distance (IGD), the spacing and the diversity are used in this paper. These metrics can be explained as follows.

4.1.1 The inverted generational distance (IGD)

IGD [28] that is the average distance between true Pareto front solutions and the non-dominated solutions is applied to evaluate the algorithm. It can measure the convergence and diversity of the solution set on a single scale. Assume that P^* is a set in the objective space along the true Pareto front. P is the set of solution points. The distance between P^* and P is defined by Equation (17) as follows:

$$IGD(P^*, P) = \frac{\sum_{a \in P} d(a, P)}{|P|} \tag{17}$$

where $d(a, P)$ is the minimum euclidean distance from the point a in set P to P^*.

Low value of IGD indicates the set P is very close to the true Pareto front. However, the actual true Pareto front of HFSP is typically unknown. Therefore, the set of non-dominated solutions obtained from all the runs of all algorithms that are the reference set P^* is used to be the representative of the approximated true Pareto front.

After obtaining the results, the normalization method is used to rescale the value of each result by Equation (18) as follows:

$$Z_y = \frac{x_y - x_{min}}{x_{max} - x_{min}} \tag{18}$$

where Z_y is the normalized value of data y of each objective, x_y is the original.

4.1.2 The spacing metric (SP)

The spacing metric [29] is used to measure the distribution of the Pareto solutions on the Pareto front. The lower *SP* is desirable because the distances between points are equidistant. *SP* can be calculated by using Equations (19) and (20) as belows:

$$SP = \sqrt{\frac{1}{N-1} \sum_{m=1}^{N} (\overline{d} - d_m)^2} \tag{19}$$

$$d_m = \min_n \left[\sum_{o=1}^{o=K} \left| f_o^m - f_o^n \right| \right] \tag{20}$$

where K is the number of objectives. m, n = 1,…,N and N is the number of points in the Pareto front. $\overline{d}$ is the average of all d_m. f_o^m, f_o^n are the value of objective.

4.2 *Design of experiment for parameters setting*

The optimal values of parameters are determined by Design of Experiment (DOE) using 2^k factorial design. The parameters are comprised of the population size, the crossover rate, the number of generations, the number of tours and the number of leader birds. There are 4 replications for each experiment plan. The response variable of this DOE is the IGD since the convergence and diversity metric are the major performances of this problem. The total runs of experiment plan for NSGA-II equals to 32 (2×2×2×4) while total runs of experiment for MBNSGA-II is 128 (2×2×2×2×2×4). Then, the levels of these parameters are shown Table 1. The optimal parameters for both algorithms obtained from the response optimizer are presented in Table 2.

Table 1: The levels of parameters for 2^k factorial design

Parameters	NSGA-II		MBNSGA-II	
	Low Level	High Level	Low Level	High Level
Population size	100	300	100	300
Crossover rate	0.4	0.8	0.4	0.8
Maximum generations	300	500	300	500
Number of tours (Small, Medium, Large)	-	-	8, 10, 1	10, 12, 12
Number of leaders birds	-	-	2	5

Table 2: The optimum level of parameters

Parameters	NSGA-II			MBNSGA-II		
	S	M	L	S	M	L
Population size	300	300	300	300	300	300
Crossover rate	0.4	0.4	0.4	0.4	0.8	0.8
Maximum generations	300	300	300	300	100	100
Number of tours	-	-	-	8	12	12
Number of leaders birds	-	-	-	5	5	5

4.3 *Experimental results*

The proposed MBNSGA-II is compared with NSGA-II for hybrid flow shop scheduling. These algorithms are coded in Matlab and run on PC with an Intel Core i7-8700 3.20 GHz with 8 GB of Ram. Each algorithm is run 5 times for each problem. For the mutation step of MBNSGA-II, the number of neighbor solutions and the number of sharing solutions with the next follower solution along the V-formation line are 5 and 2, respectively. The results are presented in Table 3.

There are three problem sizes classified by $k \times m$. Table 3 shows the average of IGD value and the standard deviation from 5 runs. From Table 3, it reveals that the average IGD value of MBNSGA-II in most problems except 40×10 and 90×14 are lower than NSGA-II. The percentage of instances that MBNSGA-II outperforms NSGA-II is 88.88%. It is clear that MBNSGA-II is closer the approximated front than NSGA-II.

Figure 5 to Figure 7 illustrate the box plots of IGD metric in each problem size including small, medium, and large size problems. The box plots show that the variance of IGD values obtained from MBNSAGA-II is significant less than those obtained from NSGA-II. These results indicate that MBNSGA-II performs better than NSAGA-II. The line plot that illustrates the average of IGD is presented in Figure 8. The line plot indicates that the average IGD values of each problem size from NSGA-II are higher than the average IGD values of each problem size from MBNSGA-II. Therefore, it is seen that MBNSGA-II outperforms NSGSA-II in terms of convergence metric.

The results from spacing metric are illustrated by the box plots in the Figures 9 and 10. These plots reveal that the average and variation of SP values obtained from NSGA-II and MBNSGA-II are not significant different. Therefore, the IGD metric is considered to be the performance index of the hybrid flow shop scheduling problem in this paper.

Table 3: IGD metric values and spacing metric values found by each algorithm

Problem	Problem Size ($k \times m$)	NSGA-II				MBNSGAI-II			
		IGD		SP		IGD	SP	IGD	SP
		Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
Small	12×5	0.226	0.119	60.88	37.69	***0.193***	0.182	***37.49***	24.82
	12×10	0.149	0.084	69.56	46.80	***0.104***	0.107	***60.74***	26.10
	18×5	0.746	0.273	***3.46***	N/A	***0.041***	0.034	67.66	20.30
	18×10	1.201	0.089	N/A	N/A	***0.012***	0.013	56.25	27.71
	24×5	0.144	0.130	308.93	134.66	***0.140***	0.167	***183.89***	167.79
	32×5	0.981	0.057	N/A	N/A	***0.082***	0.178	169.16	N/A
Medium	24×10	0.265	0.256	233.27	178.82	***0.167***	0.075	***168.81***	88.00
	32×10	1.268	0.064	***1.15***	N/A	***0.005***	0.005	348.60	358.06
	40×5	0.287	0.119	***107.68***	65.59	***0.166***	0.180	836.64	772.39
	40×10	***0.099***	0.079	***366.30***	314.03	0.128	0.113	743.96	379.24
	48×5	0.993	0.259	N/A	N/A	***0.106***	0.069	1,263.13	1,655.28
	56×5	1.131	0.059	***109.32***	68.77	***0.016***	0.013	536.79	268.35
Large	48×10	1.302	0.093	***300.80***	N/A	***0.013***	0.016	1,398.29	1,257.68
	56×10	1.341	0.038	***548.99***	330.58	***0.004***	0.004	1,000.55	908.23
	64×7	0.179	0.066	4,784.71	6621.61	***0.117***	0.105	***1,047.44***	849.62
	64×14	0.247	0.093	***719.14***	670.96	***0.194***	0.124	726.00	1,061.75
	90×7	0.211	0.082	***1,281.27***	1,483.47	***0.128***	0.185	2,278.42	1,352.66
	90×14	***0.167***	0.085	***1,629.37***	1,303.70	0.197	0.196	1,354.19	628.46

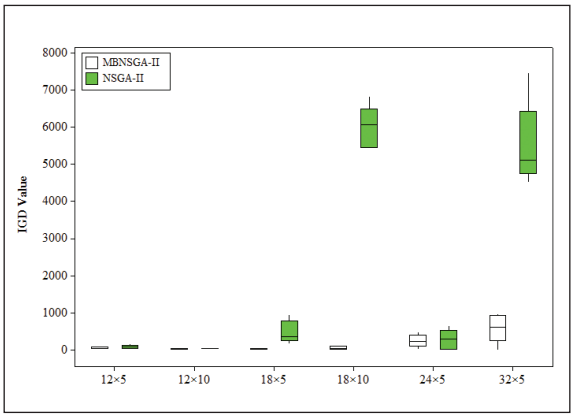

Figure 5: Boxplot of IGD for small size problems.

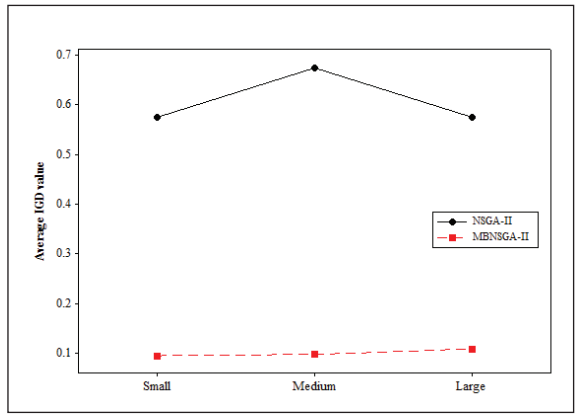

Figure 8: The average of IGD of all instances in each problem size.

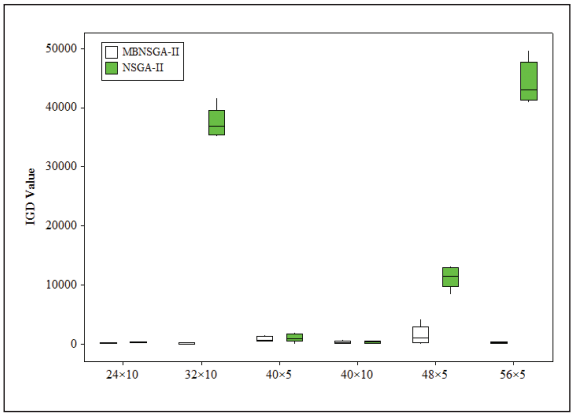

Figure 6: Boxplot of IGD for medium size problems.

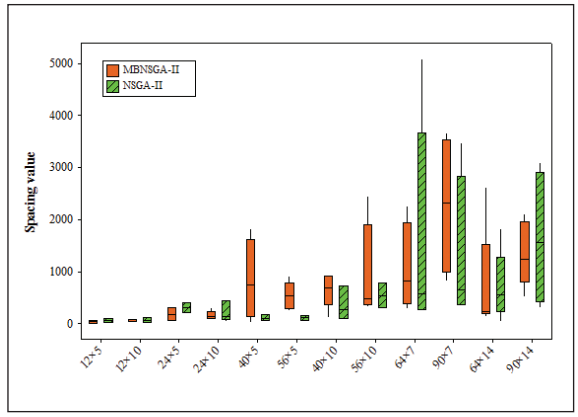

Figure 9: Boxplot of spacing value of all instances.

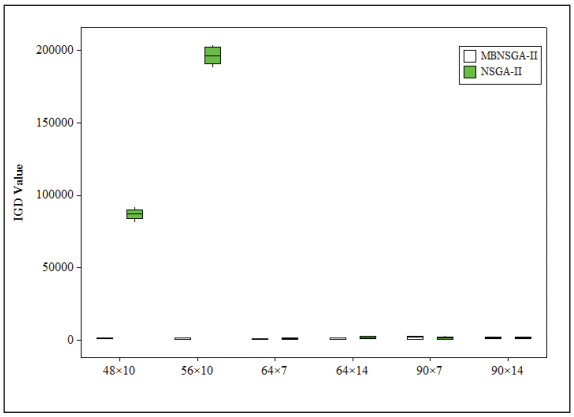

Figure 7: Boxplot of IGD for large size problems.

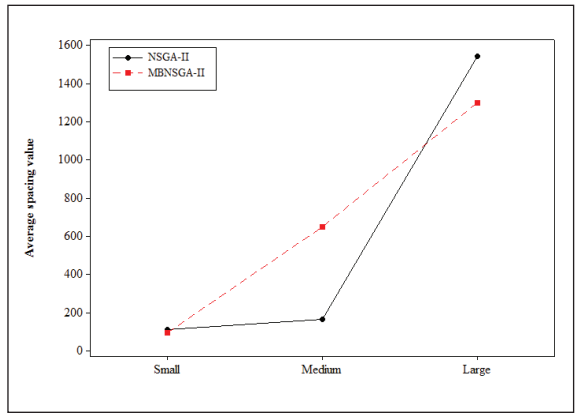

Figure 10: The average of spacing value of all instances in each problem size.

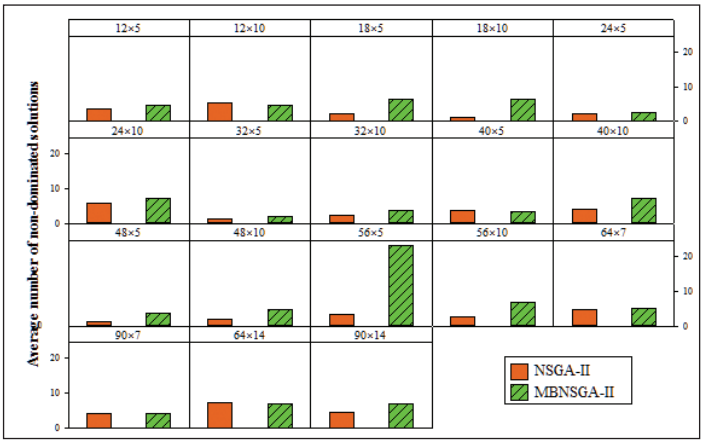

Figure 11: Average number of non-dominated solutions from 5 runs in each algorithm.

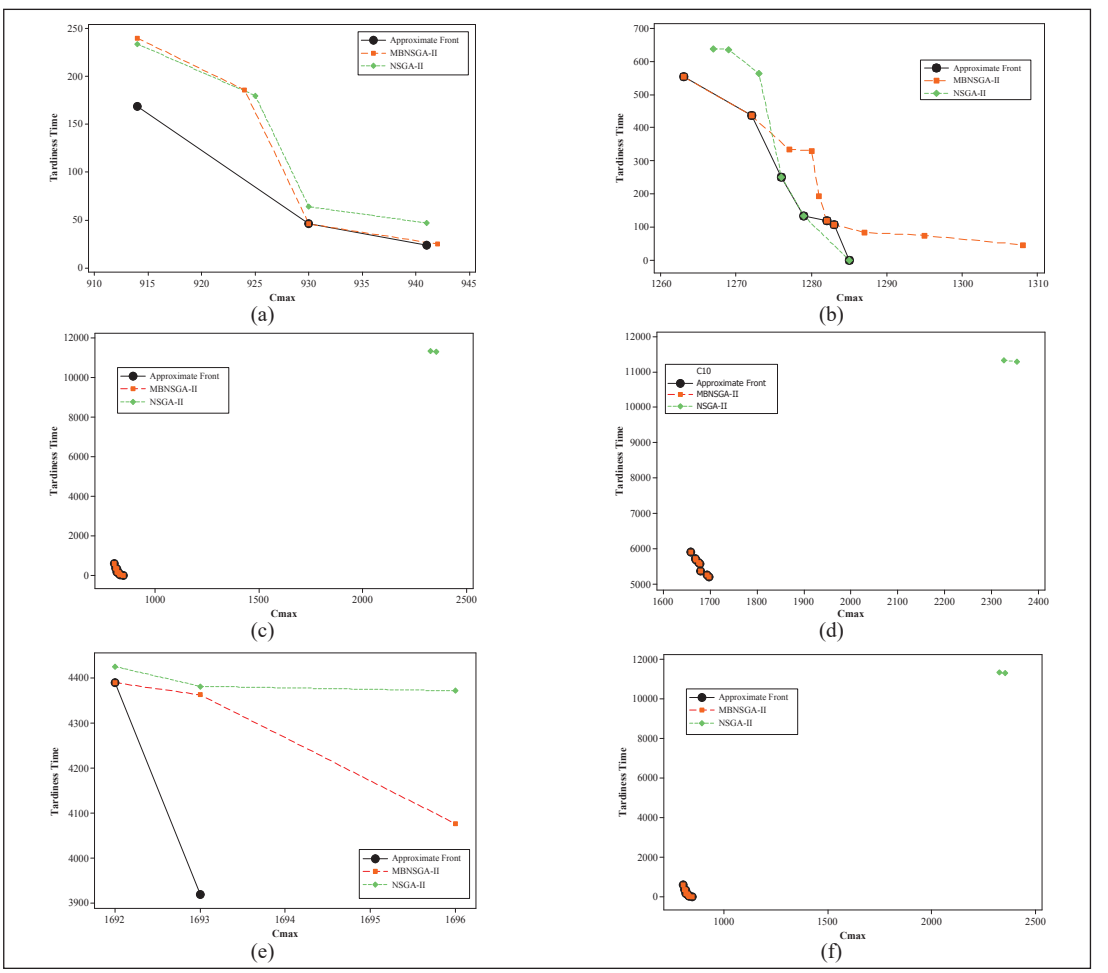

Figure 12: Pareto front of each algorithm for small size problems. (a) Pareto front of 12×5 (b) Pareto front of 12×10 (c) Pareto front of 18×5 (d) Pareto front of 18×10 (e) Pareto front of 24×5 (e) Pareto front of 32×5.

The average number of solutions including in each Pareto front from both algorithms are displayed in Figure 11. It is obvious that an average number of solutions in a Pareto front from MBNSGA-II are more than or equal to those from NSGA-II. This implies that MBNSGA-II is more efficient in finding more diversified solutions in a Pareto than NSGA-II. Figure 12 illustrates the Pareto fronts of the small size problems obtained from both algorithms.It is seen that the Pareto fronts of MBNSGA-II are close to the approximated true Pareto front. Thus, these graphical plots of Pareto fronts help us to confirm that MBNSGA-II can effectively determine the solutions that satisfies bi-objective which is minimization of makespan and total tardiness time of jobs.

5 Conclusions

This paper focuses on the development a novel hybrid metaheuristic algorithm which NSGA-II is enhanced by migrating birds optimization (MBO). The multi-objective hybrid flow shop scheduling with family like HGA production line has been discussed. There are two objectives which are to minimize the makespan (Cmax) and the total tardiness of jobs. The pareto front is the set of solution that each solution cannot dominate other solutions in that frontier. Migrating birds optimization is designed to be performed in the mutation operation using swap and insertion to generate new chromosome. IGD that is a metric for measuring the convergence and diversity of the Pareto front is employed to measure the quality of solutions or Pareto front. It is found that 16 out of 18 of the tested HFSP problems, the proposed hybrid metaheuristic algorithm (MBNSGA-II) successfully yields much better quality of Pareto fronts comparing to NSGA-II, since MBNSGA-II can yield Pareto front with the lowest IGD. The diversity of solutions of MBNSGA-II is also outperforms NSGA-II. From the performance metric results, it means that migrating birds optimization procedure can enhance the genetic algorithm to explore more diversified solutions, which are close to the approximated true Pareto front. MBNSGA-II is able to provide decision maker with the best quality of solutions.

Further research is to develop other hybrid metaheuristic algorithms to obtain better solutions for HFSP.

Acknowledgements

This research was supported by the Research and Researchers for Industries (RRi) under the Thailand Science Research and Innovation (TSRI). It offers a scholarship for Ph.D students at King Mongkut's University of Technology North Bangkok (KMUTNB) through grant PHD56I0042.

References

[1] J. E. Schaller, "Minimizing total tardiness for scheduling identical parallel machines with family setups," *Computers & Industrial Engineering*, vol. 72, pp. 274–281, 2014.

[2] M. A. Bozorgirad and R. Logendran, "Bi-criteria group scheduling in hybrid flowshops," *International Journal of Production Economics*, vol. 145, pp. 599–612, 2013.

[3] J. Behnamian and S. M. T. F. Ghomi, "Hybrid flowshop scheduling with machine and resource-dependent processing times," *Applied Mathematical Modelling*, vol. 35, pp. 1107–1123, 2011.

[4] H.-M. Cho, S. J. Bae, J. Kim, and I.-J. Jeong, "Bi-objective scheduling for reentrant hybrid flow shop using Pareto genetic algorithm," *Computers & Industrial Engineering*, vol. 61, pp. 529–541, 2011.

[5] F. Dugardin, L. Amodeo, and F. Yalaoui, "Multiobjective scheduling of a reentrant hybrid flowshop," in *Proceeding 2009 International Conference on Computers & Industrial Engineering*, 2009, pp.193–198.

[6] K.-C. Ying, S.-W. Lin, and S.-Y. Wan, "Bi-objective reentrant hybrid flowshop scheduling: An iterated Pareto greedy algorithm," *International Journal of Production Research*, vol. 52, no. 19, pp. 5735–5747, 2014.

[7] S. H. Abyaneh and M. Zandieh, "Bi-objective hybrid flow shop scheduling with sequence dependent setup times and limited buffers," *The International Journal of Advanced Manufacturing Technology*, vol. 58, pp. 309–325, 2012.

[8] S. M. Mousavi, M. Zandieh, and M. Amiri, "Comparisons of bi-objective genetic algorithms for hybrid flowshop scheduling with sequence-dependent setup times," *International Journal of Production Research*, vol. 50, no. 10, pp. 2570–2591, 2012.

[9] H. Wang, Y. Fu, M. Huang, G. Q. Quang, and J. Wang, "NSGA-II based memetic algorithm for multiobjective parallel flowshop scheduling

problem," *Computers & Industrial Engineering*, vol. 113, pp. 185–194, 2017.

[10] X. Li, H. Checade, F. Yalaoui, and L. Amodeo, "Lorenz dominance based metaheuristic to solve a hybrid flowshop scheduling problem with sequence dependent setup times," in *Proceeding of 2011 International Conference on Communications, Computing and Control Applications*, 2011, pp. 1–6.

[11] D. Lei and Y. Zheng, "Hybrid flow shop scheduling with assembly operations and key objectives: A novel neighborhood search," *Applied Soft Computing*, vol. 61, pp. 122–128, 2017.

[12] Q.-K. Pan, L. Wang, J.-Q. Li, and J.-H. Duan, "A novel discrete artificial bee colony algorithm for the hybrid flowshop scheduling problem with makespan minimization," *Omega*, vol. 45 pp. 42–56, 2014.

[13] X. Wang and L. Tang, "A tabu search heuristic for the hybrid flowshop scheduling with finite intermediate buffers," *Computers & Operations Research*, vol. 36, pp. 907–918, 2009.

[14] S. Su, H. Yu, Z. Wu, and W. Tian, "A distributed coevolutionary algorithm for multiobjective hybrid flowshop scheduling problems," *The International Journal of Advanced Manufacturing Technology*, vol. 70, pp. 477–494, 2014.

[15] E. L. Solano-Charris, J. R. Montoya-Torres, and C. D. Paternina-Arboleda, "Ant colony optimization algorithm for a Bi-criteria 2-stage hybrid flowshop scheduling problem," *Journal of Intelligent Manufacturing*, vol 22, pp. 815–822, 2011.

[16] J.-Q. Li and Q.-K. Pan, "Solving the large-scale hybrid flow shop scheduling problem with limited buffers by a hybrid artificial bee colony algorithm," *Information Sciences*, vol. 316, pp. 487–502, 2015.

[17] E. Duman, M. Uysal, and A. F. Alkaya, "Migrating birds optimization: A new metaheuristic approach and its performance on quadratic assignment problem," *Information Sciences*, vol. 217, pp. 65–77, 2012.

[18] Q.-K. Pan and Y. Dong, "An improved migrating birds optimisation for a hybrid flowshop scheduling with total flowtime minimization," *Information Sciences*, vol. 277, pp. 643–655, Sep. 2014.

[19] B. Zhang,Q.-K. Pan, L. Gao, X.-L. Zhang, H.-Y. Sang, and J.-Q. Li, "An effective modified migrating birds optimization for hybrid flowshop scheduling problem with lot streaming," *Applied Soft Computing*, vol. 52, pp. 14–27, 2017.

[20] T. Meng, Q.-K. Pan, J.-Q. Li, and H.-Y. Sang, "An improved migrating birds optimization for an integrated lot-streaming flow shop scheduling problem," *Swarm and Evolutionary Computation*, vol. 38, pp. 64–78, 2018.

[21] A. Sioud and C. Gagné, "Enhanced migrating birds optimization algorithm for the permutation flow shop problem with sequence dependent setup times," *European Journal of Operational Research*, vol. 264, pp. 66–73, 2018.

[22] L. Gao and Q.-K. Pan, "A shuffled multi-swarm micro-migrating birds optimizer for a multi-resource-constrained flexible job shop scheduling problem," *Information Sciences*, vol. 372, pp. 655–676, 2016.

[23] B. Zhang, Q.-K. Pan, L. Gao, X.-L. Zhang, and K.-K. Peng, "A multi-objective migrating birds optimization algorithm for the hybrid flowshop rescheduling problem," *Soft Computing*, pp. 1–29, 2018.

[24] T. Sawik, *Scheduling in Supply Chains Using Mixed Integer Programming*. New Jersey: John Wiley and Sons, 2011.

[25] K. Deb, A. Pratap, S. Agarwal, and T. A. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[26] M. Gholami, M. Zandieh, and A. Alem-Tabriz, "Scheduling hybrid flow shop with sequence-dependent setup times and machines with random breakdowns," *The International Journal of Advanced Manufacturing Technology*, vol. 42, pp. 189–201, 2009.

[27] N. Karimi and H. Davoupour, "Multi-objective colonial competitive algorithm for hybrid flowshop problem," *Applied Soft Computing*, vol. 49, pp. 725–733, 2016.

[28] C. A. C. Coello and N. C. Cort´es, "Solving multiobjective optimization problems using an artificial immune system," *Genetic Programming and Evolvable Machines*, vol. 6, no. 2, pp. 163–190, 2005.

[29] J. R. Schott, "Fault tolerant design using single and multicriteria genetic algorithm optimization," M.S. thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, MA, USA, 1995.