

# Nature-inspired Metaheuristic Optimization Technique-Migrating bird's optimization in Industrial Scheduling Problem

<sup>1</sup>Ramanathan.L, <sup>2</sup>Ulaganathan.K

<sup>1</sup>Student, Dept. of Mechanical Engineering, Kumaraguru College of Technology  
Coimbatore, India

<sup>2</sup>SRG Prof, Dept. of Mechanical Engineering, Kumaraguru College of Technology  
Coimbatore, India

## Abstract

Migrating birds optimization (MBO) is a new nature-inspired metaheuristic for combinatorial optimization problems. This paper proposes application of MBO in a flow shop sequencing problem, which has important practical applications in modern industry. FSSP is a typical NP-Hard problem (non deterministic polynomial time) which is desired to be minimum make span. As the basic MBO algorithm is designed for discrete problems. The performance of basic MBO algorithm is tested via some FSSP data sets exist in literature. A mixed neighborhood is constructed for the leader and the following birds to easily find promising neighboring solutions. Extensive comparative evaluations are conducted with recently published algorithms in the literature.

**Keywords :** Scheduling, Flow shop, NP-Hard, Metaheuristic Methods, Make span.

## I. INTRODUCTION

Scheduling plays vital role in several industries. Effective scheduling techniques should be required for improving the efficiency of industries. Scheduling may be defined as a process of allocating resources over time to perform a collection of tasks. Different types of scheduling problems were addressed in the literature. This paper considers a flow shop scheduling problem. The flow shop scheduling problem is one of the most important scheduling problems. Many manufacturing systems and assembly lines resemble the flow shop scheduling environment. In the flow shop, a set of  $n$  jobs are to be processed in an identical order in a given set of machines. The flow shop scheduling model was first developed by Johnson. Johnson developed an exact algorithm to minimize the make span for 2-machines flow shop scheduling problems. The flow shop scheduling problem has been proved to be NP-hard. Due to the complexity of the problem, it is difficult to develop exact methods to solve this problem. Hence, researchers proposed different heuristics and metaheuristics to solve the flow shop scheduling problems.

Recently, researchers adapted different metaheuristics to solve the flow shop scheduling

problems. A genetic algorithm (GA) was applied to solve the flow shop scheduling problems by Reeves. Murata et al. solved the flow shop scheduling problems using the GA. Nowicki and Smutnicki applied the tabu search (TS) algorithm for solve flow shop scheduling problems with parallel machines. Chin et al. addressed a discrete version of particle swarm optimization (PSO) algorithm for solving the flow shop scheduling problems. Liu et al. presented an effective hybrid particle swarm optimization algorithm for solving the no-wait flow shop scheduling problem with make span criterion. Ying and Lin proposed an ant colony system heuristic for solving the non-permutation flow shop scheduling problems. Jarboui et al. proposed a hybrid GA to solve the flow shop scheduling problems. Migrating birds optimization (MBO) is a new metaheuristic that was presented by Duman et al. For solving quadratic assignment problems. MBO is inspired from the V flight formation of migrating birds, which is a very effective formation in terms of energy minimization. The authors showed that MBO could be an important player in metaheuristic-based optimizations. Following its successful applications, this paper presents an improved MBO for the HFS problem with a total flow-time criterion. We introduce some advanced and effective technologies, including a diversified initialization approach, a mixed neighborhood structure, and a leaping mechanism. We demonstrate the effectiveness of the proposed MBO with extensive comparisons to several high-performing algorithms in the literature.

The rest of the paper is organized as follows. The problem is defined in section 2. The proposed algorithm is presented in section 3. Finally, the conclusions and future research opportunities are discussed in section 4.

## II. HYBRID FLOWSHOP SCHEDULING PROBLEM

The hybrid flow shop is composed of a series of  $m$  production stages. Each stage  $k$  has  $s_k$  identical parallel machines, and  $s_k \geq 1$  for at least one stage. A set of jobs are required to be sequentially processed in the same production order, i.e., first on stage 1, then

on stage 2. . . And finally on stage m. At each stage k, job j can be processed on any machine  $i \in s_k$ . At any time, no job can be processed on more than one machine, and no machine can process more than one job simultaneously. All of the jobs are independent and available for processing at time 0. Job setup times and travel times between consecutive stages are included in the job processing times or can be negligible. The objective is then to find a schedule in which the total flow time is minimized.

### III. CONSTRAINTS

The Assumptions for this problem are as follows.

- Every job has to be processed at maximum once on machine. Every machine processes only one job at a time.
- Every job is processed at maximum on one machine at a time.
- The preparation times of the operations are included in the processing time and do not depend on the sequence. The operating sequences of the jobs are the same on every machine.

### IV. INTRODUCTION TO THE BASIC MBO ALGORITHM

In the MBO, the solutions are treated as birds aligned in a V formation. Each solution can derive benefit from the solution in front of it. MBO is formed with a number of initial solutions. Starting with the first solution, which corresponds to the leader bird in the flock, MBO attempts to improve the solution by exploring its neighborhood. Then, the following solution evaluates a number of its own neighbours and a number of the best unused neighbours from its previous solution (here, 'un-used' means a neighbour solution that is not used to replace the existing solution). The best solution replaces the current solution if it is better. The improvement process progresses toward the tails. Once all of the solutions in the flock are considered, this process is repeated again. After a number of tours, the leader solution is moved to the end of the line, and one of the solutions following it is forwarded to the leader position. Then, another loop starts. The above procedure is repeated until a termination condition is met. MBO is composed of four basic phases: initialization, improvement upon the leader, improvement upon the followers, and selecting a new leader. These are described below

#### A. Initialization

Initialization has two steps. The first step is to set the algorithmic parameters, including the population size (a) or the number of initial solutions, the number of neighbouring solutions to be considered (b), the number of neighbouring solutions to be shared with the next solution (v), and the number of tours (x). They correspond respectively to the number of birds in the flock, the speed of flight, the wing-tip spacing,

and the number of wing flaps before there is a change in the order of the birds (or the profiling energy spent). After extensive experiments, Duman et al. Suggested that  $a = 51$ ,  $x = 10$ ,  $b = 3$ , and  $v = 1$ .

The second step is to create the initial population. The initial population consists of solutions that were randomly Generated in the feasible solution space. One of them is selected as the leader. The remaining  $a-1$  solutions are divided equally into two groups. The two groups are then added to the left list (denoted as  $L_l = \{x_{l,1}; x_{l,2}; \dots; x_{l,a-1}\}$ ) and the right list (denoted as  $L_r = \{x_{r,1}; x_{r,2}; \dots; x_{r,a-1}\}$ ), respectively, where  $x_{l,k}/x_{r,k}$  is the kth solution in  $L_l/L_r$ . This arrangement is similar to the migrating birds' scenario, where a bird leads the flock and two lines of other birds follow it.

#### B. Improvement Upon the Leader

To improve the leader solution, b neighbouring solutions are randomly generated. If the best neighbouring solution is better than the leader, then the leader is replaced by this solution; otherwise, the leader stays unchanged. The remaining  $b - 1$  neighbouring solutions are sorted in ascending order of their objective values (for the minimization optimization problem). Then, two shared neighbour sets, specifically the left neighbour set  $K_l$  and the right neighbour set  $K_r$ , are formed as follows. The first neighbouring solution enters  $K_l$ , the second enters  $K_r$ , the third enters  $K_l$ , and the fourth enters  $K_r$ , and so on, until both  $K_l$  and  $K_r$  are filled with v solutions.

#### C. Equation Improvement Upon the Followers

The exploring process progresses along the lines toward the tails. For a solution  $x_{l,k} \in L_l$ , randomly generate  $b-v$  neighbouring solutions in its neighbourhood. Evaluate these  $b-v$  neighbouring solutions and v solutions from the left neighbour set  $K_l$ . The best solution is used to improve the current solution  $x_{l,k}$  if it has a better objective value. The best v solutions from the remaining  $b - 1$  solutions are used to fill  $K_l$  after  $K_l$  is reset to zero. The above procedure is repeated until all of the solutions in the left list  $L_l$  have been explored.

The same method is used to improve the solutions in the right list  $L_r$ . The benefit mechanism in which a solution shares the best unused neighbours with the subsequent solutions is totally unique to the MBO. With this benefit mechanism, a solution that fails to improve itself with its own neighbours is replaced by one of the neighbours from the previous solution if the previous solution is more promising. In this way, the region around the more promising solution will be explored in greater detail.

#### D. Selecting a New Leader

After the improvement procedure from the leader to the tails is performed for x replications, the order of the solutions is changed. The leader solution

is moved alternately to the ends of the left and the right lists, and the first solution in the corresponding list is forwarded as the new leader. This procedure is similar to what real birds do; the bird that spends the most energy and thus tires moves back to rest and another bird fills its position.

### E. Computational Procedure

The procedure of the MBO algorithm is as follows (see Fig. 1). As stated, the properties of MBO that distinguish it from other metaheuristic approaches are that a number of solutions run in parallel and the benefit mechanism between the solutions. Parallel processing can somehow be regarded as being inherent to genetic algorithms and scatter search. On the other hand, although MBO appears to have some similarities to swarm intelligence algorithms and to an artificial bee colony algorithm in particular in which better solutions are explored more, the benefit mechanism is totally unique to the MBO.

1. Procedure MBO
2. Initialize the algorithm
3. While not termination
4. Repeat
5. Improve the leader solution
6. Improve each solution in the left list
7. Improve each solution in the right list
8. Until a number  $\omega$  of replications
9. Select a new leader solution
10. End while
11. End

Fig. 1. The Procedure of the Basic MBO.

## V. THE PROPOSED MBO FOR THE HFS PROBLEM

This section proposes an MBO algorithm for the HFS problem with a total flow time criterion. A diversified method is presented to form an initial population that spreads out widely in solution space. A mixed neighbourhood is constructed for the leader and following birds to find promising neighbouring solutions easily. A leaping mechanism is developed to help the MBO escape from suboptimal solutions. With these advanced and effective technologies, the MBO is expected to generate high quality solutions with robustness for the HFS problem under consideration.

### A. Solution Representation

We adopt the permutation-based representation. In the encoding, a permutation of jobs

in an array represents the order in which the jobs are launched to the shop at the first stage. A job is allocated to the first machine that becomes available, i.e., the machine that is the first to finish the job (if any) that was previously assigned to it. For subsequent stages, the schedule is arranged as soon as the jobs are completed by the preceding stage.

### B. Mathematical Model

Flow shop sequencing problem minimizing the time between the beginning of perform of the first job on the first machine and the completion of perform of the last job on the last machine. This time is called make span. Assumptions for this problem are as follows:

- a. Every job has to be processed at maximum once on machine.
- b. Every machine processes only one job at a time.
- c. Every job is processed at maximum on one machine at a time.
- d. The preparation times of the operations are included in the processing time and do not depend on the sequence.
- e. The operating sequences of the jobs are the same on every machine.

For  $n$  jobs and  $m$  machine,

Job permutation  $\pi = \{\pi_1, \pi_2, \pi_3, \dots, \pi_n\}$  and  $p(i, j)$  processing time for job  $i$  on machine  $j$ .

$$C(\pi_1, 1) = p(\pi_1, 1)$$

$$C(\pi_i, 1) = C(\pi_{i-1}, 1) + p(\pi_i, 1) \text{ for } i = 2, \dots, n$$

$$C(\pi_1, j) = C(\pi_1, j-1) + p(\pi_1, j) \text{ for } j = 2, \dots, m$$

$$C(\pi_i, j) = \max \{C(\pi_{i-1}, j), C(\pi_i, j-1)\} + p(\pi_i, j)$$

$$\text{For } i = 2, n; \text{ for } j = 2, \dots, m$$

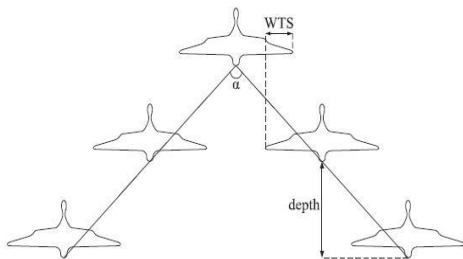
$$\text{Makespan } C_{\max} = C(\pi_n, m).$$

### C. Migrating Bird Algorithm

MBO is a neighbouring research technique including algorithm. Each bird in V formation represents a solution. Starts with first solution (leader bird) and goes on till tail. Each solution has solutions which try to advance it. These neighboring solutions provide the opportunity of local searches in the position of existing solution. If the neighbor solutions are better than the existing one, they are replaced. Apart from the other metaheuristic algorithms, there is another efficiency mechanism in MBO algorithm. This mechanism transfers the best unused neighbor solutions to following solution. In algorithm, this situation is called as sharing the neighbors. In this way, existing solution enhances itself by using not only its own produced solutions but also neighbor solutions which come from other solutions. This neighbor sharing is concluded with sharing of the solution by all birds (except for the last two birds in the tail). After that the leader bird is changed and new neighborhoods and neighbor sharing, which will be produced by new leader bird, is restarted. Algorithm shows similarities with real life of migrating birds.  $n$  value shows number of birds. First birds are in V

formation. An existing solution is produced for each bird.  $k$  value is thought as speed of a bird in real life. The bird should fly slowly for a more detailed discovery around itself. So  $k$  parameter is expected to have low values in order to be able to reach better solutions.  $x$  value is thought as wing distance of the bird (WTS length in Figure 2). In experimental studies of Lissaman and Schollenberger, optimum wing distance for the 1.5 meter wing length birds is determined as 16 cm. Which means that  $x$  parameter will be appointed very low values.  $m$  value is thought as number of flutter of the bird. It is the flutter number until leader bird gets tired and goes back for resting. According to algorithm, when leader bird gets tired it goes to the back of flock and its following bird becomes leader bird. As the birds are in V formation, first the bird on the left of the leader bird takes the position of leader bird. In this condition leader bird goes to the back line of the left order. In the next change, the bird on the right of the leader bird takes its position and leader goes to the right back of the flock. And the iteration number of algorithm should be as much as that each bird would become leader at least once.

1. Generate  $n$  initial solutions in a random manner and place them on a hypothetical V formation arbitrarily.
2.  $i = 0$
3. while( $i < K$ )
4.     for ( $j = 0; j < m; j++$ )
5.         Try to improve the leading solution by generating and evaluating  $k$  neighbors of it.
6.          $i = i + k$
7.         for each solution  $s_r$  in the flock (except leader)
8.             Try to improve  $s_r$  by evaluating  $(k-x)$  neighbors of it and  $x$  unused best neighbors from the



- solution in the front.
9.          $i = i + (k - x)$
  10.        endfor
  11.        endfor
  12.        Move the leader solution to the end and forward one of the solutions following it to the leader position

13.    endwhile
  14.    return the best solution in the flock
- $n$ : amount of beginning solutions (number of birds)  
 $k$ : number of related neighbour solutions  
 $x$ : number of neighbour solutions which are shared by final  
 $m$ : number of lap  
 $K$ : iteration limit

Fig. 2. The V- Formation.

## VI. SOLUTION OF FSSP WITH MBO ALGORITHM

As in most metaheuristic algorithms, MBO starts with a beginning solution. In the beginning, permutation of all jobs on all machines is created for each bird and make span is determined. For  $n$  job and  $m$  machine  $m \times n$  matrix is formed. Therefore all jobs and all machines are placed in the  $m \times n$  matrix. After that neighboring is produced for each bird in the flock in enough amounts and if the best neighbor of the existing solution is better than the existing solution, neighbor solution is replaced by existing solution. Neighbors are produced faithfully to original algorithm. The best unused solution is transferred to the bird in the back order. And the worst unused solution of the related bird's back order is dismissed.

## VII. EXPERIMENTAL RESULTS

In the application, MBO algorithm is tested Taillard's flow shop sequence problems. All tested problems are executed 10 times by keeping the starting point same and results are averaged. Iteration value of basic MBO algorithm is depended upon the produced neighbor amount and size of the problem. This value determined as  $n^4$  ( $n =$  job count). In both problems, static parameter values are given in Table 1.

Values for 10 times execution of all tested problems are saved. According to these values, maximum, minimum and average values of makespan are given in Tables 2-4 which are calculated by considering average of 10 studies for some iteration. Best known upper bounds of tested Taillard's instances are given in Table 5.

Table 1. Starting Parameter Values.

Mbo parameters			
n	k	x	m
51	3	1	10

Table 2. Maximum, Minimum and Average Values for 20 x 5 Instance 1.

Leader Replace Number	20 x 5 instance1		
	Min.	Max.	Avg.
1	1334.4	1426	1374.5
5	1297	1340.5	1311.6

10	1295.7	1305.9	1297.9
50	1291.2	1297	1295.4
100	1288.1	1296.8	1291.3
156	1284.5	1290.1	1287.9

**Table 3. Maximum, Minimum and Average Values for 20 × 10 Instance 1.**

Leader Replace Number	20 × 10 instance1		
	Min.	Max.	Avg.
1	1755.6	1890.4	1825
5	1658.8	1758.3	1711.1
10	1639.8	1712.6	1676.3
50	1601.9	1646.3	1624.7
100	1594.2	1635	1615.3
156	1591	1627.2	1610.6

**Table 4. Maximum, Minimum and Average Values for 20 × 20 Instance 1.**

Leader Replace Number	20 × 20 instance1		
	Min.	Max.	Avg.
1	2474.9	2629.6	2552.4
5	2395.4	2497.7	2448.7
10	2364.2	2455.1	2413.5
50	2325.6	2392.6	2355.4
100	2320.6	2375.3	2343
156	2318.9	2362.9	2337.7

**Table 5. Upper Bound Values of Tested Problems.**

Upper Bound Values		
20 × 5 ins.1	20 × 10 ins.1	20 × 20 ins.1
1278	1582	2297

**VIII. CONCLUSION**

In this article, solutions are sought for the well known optimization problem flow shop sequencing problem with migrating birds optimization (MBO) algorithm which is a new metaheuristic approach. Developed algorithm is experimentally tested by well known data set which are chosen from Taillard’s benchmark and compared according to optimal solutions. As a result of proposed study, for FSSP; it is seen that MBO algorithm follows a performance progress and has a more consistent approach to result.

**ACKNOWLEDGMENT**

The authors would like to thank the anonymous referees for their valuable comments and helpful suggestions, which have improved the quality and presentation of this paper. We also wish to express our thanks to Professor .Ekrem Duman, Ozyegin University, Department of Industrial Engineering, Alemdag, Cekmekoy, Istanbul, Turkey for providing the concept and algorithm.

**REFERENCES**

[1] Duman.E,Uysal.M and Alkaya.A.F,(2012),“Migrating bird optimization: new metaheuristic approach and its performance on quadratic assignment problems”. Information sciences, Vol.217,pp.65-77.

[2] Imma Ribas,Rainer Leisten, Jose.M Framinan,(2009),“Review and classification of hybrid flow shop scheduling problems from A production system and A solutions procedure perspective”. Computers and operations research, Vol.37,pp.1439-1454.

[3] Ravindra K. Ahuja ,Ozlem Ergun,james B. Orlin,(2002),“A survey of very large scale neighborhood search techniques”, Discrete Applied Mathematics, Vol.123,pp.75-102.

[4] Nilgun Fescioglu-unver,(2010),“Self controlling Tabu search algorithm for the quadratic assignment problem.” Computers And Industrial Engineering, Vol.60,pp. 310-319.

[5] Rui Zhang,Cheng Wu,(2009),“Bottleneck machine identification method based on constraint transformation for job shop scheduling with genetic algorithm”, Information Sciences, Vol.188,pp.236-252.

[6] Guan-chun Luh,chung-huei Chueh,(2007),“A multi-modal immune algorithm for the job shop scheduling problem”, Information Sciences, Vol.179,pp.1516-1532.

[7] Thomas Alerstam,Dept Of Biology,(2012),“Optimal bird migration revisited”. Peer-00697238,version 1.

[8] Lin-yu Tseng,ya-tai Lin,(2007),“A genetic local algorithm for minimizing total flow time in the permutation flow shop scheduling problem”. Int.J.Production Economics, Vol.127,pp.121-128.

[9] Herbert G.Campbell,Richard A.Dudek,(1970), "A heuristic algorithm for the N job,m machine sequencing problem". Management Science, Vol.160,pp.123-130.

[10] Orhan Engin,Alper Doyen,(2004),“A new approach to solve hybrid flow shop scheduling problem by artificial immune system ”,Future Generation Computer Systems, Vol.20,pp.1083-1095.

[11] Mohsen Ziaee,(2010),“General flow shop scheduling problem with sequence dependent setup times: A heuristic approach”, Information Sciences, Vol.251,pp.126-135.

[12] Ruben Ruiz,jose Antonio Vazquez ,(2009),“The hybrid flow shop scheduling problem”, European Journal Of Operation Research, Vol.205,pp.1-18.

[13] David J. Willis,Jaime Peraire,(1999),“A computational investigation of bio inspired formation flight and ground effect ”.American Institute Of Aeronautics And Astronautics, Vol.1,pp.1-35.

[14] J.Behnamian,M.Zandieh,(2011),“A discrete colonial competitive algorithm for hybrid flow shop scheduling to minimize earliness and quadratic tardiness penalties”, Expert Systems With Applications, Vol.38,pp.14490-14498.

[15] Rubenz Ruiz, Concepcion Maroto,(2005),“A genetic algorithm for hybrid flow shops with sequence dependent setup times and machine eligibility”, European Journal Of Operational Research, Vol.169,pp.781-800.

[16] B.Naderi,Ruben Ruiz ,(2009),“Algorithms for A realistic variant of flow shop scheduling”, Computers & Operations Research, Vol. 37,pp.236-246.

[17] Ching-jong Liao,Evi Tjandradjaja,(2011),“An approach using particle swarm optimization and bottleneck heuristic to solve hybrid flow shop scheduling problem”, Applied Soft Computing, Vol.12,pp.1755-1764.

[18] B.Naderi,m.Zandieh,(2009),“An improved simulated annealing for hybrid flow shops with sequence dependent setup and transportation times to minimize total completion time and total tardiness”, Expert Systems With Applications, Vol.36,pp.9625-9633.

[19] Zhihong Jin,zan Yang,(2006),“ Metaheuristic algorithms for the multistage hybrid flow shop scheduling problem”, Int.J.Production Economics, Vol.100,pp.322-334.

[20] Shih-chang Wang,ming-feng Yeh,(2010),“A modified particle swarm optimization for aggregate production planning”, Expert Systems With Applications , Vol.41,pp.3069-3077.

[21] Magdalene Marinaki,Yannis Mariachis,(2008),“Honey bees mating optimization algorithm for financial classification problems”, Applied Soft Computing, Vol.10,pp.806-812.

[22] Alireza Askarzadeh,(2012),“ Bird mating optimizer: an optimization algorithm inspired by bird mating

- strategies", Commun Nonlinear Sci Numer Simulat, Vol.19, pp.1213-1228.
- [23] Samir Sayah,abdellatif Hamouda,(2011),"A hybrid differential evolution algorithm based on particle swarm optimization for nonconvex economic dispatch problems", Applied Soft Computing, Vol.13, pp.1608-1619.
- [24] Christopher C.Yang, Jerome Yen,(2000),"Intelligent internet searching agent based on hybrid simulated annealing", Decision Support Systems, Vol.28, pp.269-277.
- [25] B.Naderi, Ruben Ruiz,(2009),"Algorithms for A realistic variant of flow shop scheduling", Computers & Operations Research, Vol.37, pp.236-246.
- [26] Huizhen Zhang, Cesar Beltran,(2010),"Effective Formulation Reductions For The Quadratic Assignment Problem", Computers & Operations Research, Vol. 37, pp.2007-2016.
- [27] Ching-jong Liao, Evi Tjandradjaja,(2011),"An approach using particle swarm optimization and bottleneck heuristic to solve hybrid flow shop scheduling problem", Applied Soft Computing, Vol.12, pp.1755-1764.
- [28] Rui Zhang, Cheng Wu,(2009), "Bottleneck machine identification method based on constraint transformation for job shop scheduling with genetic algorithm", Information Sciences, Vol.188, pp.236-252.
- [29] Wen-Hung Kuo, Chou-Jung Hsu,(2010), "Worst-case and numerical analysis of heuristic algorithms for flow shop scheduling problems with a time-dependent learning effect", Information Sciences, Vol.184, pp.282-297.
- [30] Salwani Abdullah, Hamza Turabieh,(2011), "On the use of multi neighborhood structures within A tabu-based memetic approach to university timetabling problem", Information Sciences, Vol.191, pp.146-168.