Available online at www.ijpe-online.com

International Journal of Performability Engineering

vol. 13, no. 7, November 2017, pp. 1020-1029
DOI: 10.23940/ijpe.17.07.p3.10201029

A Variable Neighborhood Migrating Birds Optimization Algorithm for Flexible Job Shop Scheduling

Hongchan Li[a], Bangqin Cao[b], Haodong Zhu[a,*]

[a]*School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou, Henan, 450002, China*
[b]*Xinyang Vocational and Technical College, Xinyang, Henan, 464000, China*

Abstract

A hybrid meta-heuristic named variable neighborhood migrating birds optimization (VNMBO), which is a combination of variable neighborhood search (VNS) and migrating birds optimization (MBO). The main aim of this paper is to provide a new way for MBO to solve the flexible job shop scheduling problem (FJSP). A two-stage population initialization scheme was first adopted to improve the quality of the initial solutions. An individual leaping mechanism was introduced to the algorithm in order to avoid the premature convergence. To search the solution space effectively, three neighborhood structures were designed and a VNS was developed to enhance the local searching ability. Finally, to assess the performance of the proposed VNMBO, some published algorithms were compared by using two famous benchmark data sets. The comparison results show that the proposed VNMBO is effective for solving the FJSP with the objective of minimizing the makespan.

Keywords: variable neighborhood search; migrating birds optimization algorithm; flexible job shop; makespan

(Submitted on July 24, 2017; Revised on October 13, 2017; Accepted on October 21, 2017)

1. Introduction

Workshop scheduling plays a more and more important role in the production management system due to the growing consumer demands, reduced product life cycles, and constantly changing market conditions [19]. An effective scheduling scheme can help the manufacturers reduce the production cost and delivering product time to customers. The scheduling process is a decision-making process that deals with the limited resources allocation to tasks, and has a big impact on achieving various production goals.

Since the first scheduling research appeared in 1954 by Johnson [5], studies on this topic have taken serious attention in the manufacturing field. The classic job shop scheduling problem (JSP) is one of the most popular problems in the manufacturing area, which has been proved to be NP-hard. In JSP, operations can be only processed on one machine. However, in the real-life production, the processing of an operation can be conducted on different machines, so that an extension of the JSP called the flexible job shop scheduling problem (FJSP) is further considered. It breaks the limitation of unique resources and allows each operation to be processed by alternative machines. In the past decades, the FJSP has captured the interests of many researchers. Kacem et al. [6,7] dealt with the multi-objective FJSP and proposed some approaches to minimize the makespan, the total workload of machines and the workload of the critical machine. Brandimarte [1] proposed a hierarchical algorithm for solving the FJSP based on the tabu search. Ziaee [19] considered the FJSP with the objective of minimizing the makespan and developed an efficient heuristic algorithm. Li et al. [9] combined the particle swarm optimization with tabu search and presented a PSO+TS to minimize the makespan in the FJSP. Xing et al. [18] integrated the ant colony optimization (ACO) model and knowledge model to propose a KBACO algorithm. Nouiri et al. [11] developed an efficient and distributed particle swarm optimization algorithm (PSO) to minimize the makespan.

* Corresponding author. Tel.: +86-13592697657; fax: +86-0371-86609559.
E-mail address: zhdjsj016@163.com.

With the rapid development of intelligent algorithms, a new population-based algorithm called migrating birds optimization (MBO) is first proposed by Duman et al. [3] for solving quadratic assignment problems. The basic idea of the algorithm comes from the V-shaped flight formation of migrating birds. Duman et al. [3] demonstrated that MBO performs better than other metaheuristic-based optimization algorithms. At present, some researchers have applied the MBO to solve the scheduling problem in the area of production and operation management. Pan and Dong [12] proposed an improved MBO to solve the hybrid flow shop scheduling problem with the objective of minimizing the total flow time. Xie et al. [17] studied the scheduling problem in a blocking flow shop and proposed an effective MBO to minimize the flow time. Gao and Pan [4] proposed a shuffled multi-swarm micro-migrating birds optimization algorithm to solve a multi-resource constrained flexible job shop scheduling problem. Tao et al. [10] considered an integrated lot-streaming flow shop scheduling problem and presented an improved migrating birds optimization (IMMBO) to minimize the makespan. Following the successful applications of MBO, the main contribution of this paper is to derive a variable neighborhood migrating birds optimization (VNMBO) for solving the FJSP with the criterion to minimize the makespan. In the proposed algorithm, some effective technologies were introduced into the basic MBO, which are the initial population generation approach, the individual leaping mechanism and the variable neighborhood search strategy. Extensive experimental data demonstrates the effectiveness of our proposed algorithm.

2. Problem description and formulation

For a FJSP, n jobs need to be processed on m machines. Each job consists of several operations with a predetermined order in sequence. Processing time of each operation is determined by the assigned machine. Two sub-problems are involved in the FJSP, i.e., the machine assignment and the operation permutation. Machine assignment attempts to assign each operation to an appropriate machine. Operation permutation aims to acquire the processing sequence of operations assigned to machines.

For such a complex system, some assumptions are listed as below.

(1)Machines and jobs are simultaneously available at time 0.
(2)Each machine can perform only one operation at a time.
(3)No job preemption is allowed.
(4)Precedence constraints exit between operations of the same job. That is, each operation must be processed after its predecessor operation is completed.
(5)Jobs are independent with each other and have the same priorities.
(6)Setup time between two successive operations on the same machine can be negligible.
To facilitate the establishment of the mathematical model, some symbols and variables are defined.
n : number of jobs;
m : number of machines;
J_i : number of operations of job i ;
O_{ij} : the j th operation of job i ;
p_{ijk} : processing time of O_{ij} on machine k ;
S_{ij} : start time of O_{ij} ;
C_{ij} : complete time of O_{ij} ;
$C_{\max}$: makespan;
η : a large positive number;
x_{ijk} : 0-1 variable, if O_{ij} is processed on machine k , x_{ijk} =1; otherwise, x_{ijk} =0;
$y_{iji'j'k}$: 0-1 variable, if O_{ij} is processed on machine k prior to O_{ij} , $y_{iji'j'k}$ =1; otherwise, $y_{iji'j'k}$ =0.
The general mathematical model of the FJSP with the criterion to minimize the makespan is built as below:

$$\min C_{\max} = \min(\max(C_{ij})) \tag{1}$$

$$\text{s.t. } C_{ij} = S_{ij} + \sum_{k=1}^{m} x_{ijk} p_{ijk},\ i = 1, 2, \cdots, n;\ j = 1, 2, \cdots, J_i \tag{2}$$

$$S_{i(j+1)} - C_{ij} \geq 0,\ i = 1, 2, \cdots, n;\ j = 1, 2, \cdots, J_i - 1 \tag{3}$$

$$S_{i'j'} + \eta(1 - y_{iji'j'k}) \geq C_{ij},\ \ i, i' = 1, 2, \cdots, n;\ j, j' = 1, 2, \cdots, J_i; k = 1, 2, \cdots, m \tag{4}$$

$$S_{ij} + \eta y_{iji'j'k} \geq C_{i'j'},\ i, i' = 1, 2, \cdots, n;\ j, j' = 1, 2, \cdots, J_i; k = 1, 2, \cdots, m \tag{5}$$

$$\sum_{k=1}^{m} x_{ijk} = 1,\ i = 1, 2, \cdots, n;\ j = 1, 2, \cdots, J_i \tag{6}$$

$$x_{ijk} \in \{0,1\},\ i = 1, 2, \cdots, n;\ j = 1, 2, \cdots, J_i;\ k = 1, 2, \cdots, m \tag{7}$$

$$y_{iji'j'k} \in \{0,1\},\ i, i' = 1, 2, \cdots, n;\ j, j' = 1, 2, \cdots, J_i;\ k = 1, 2, \cdots, m \tag{8}$$

Equation (1) means that the optimization objective is to minimize the maximum completion time; constraint (2) ensures that no preemption is allowed for each operation; constraint (3) guarantees the precedence constraints between operations; constraint (4) and (5) present that every machine can only perform one operation at a time; constraint (6) shows that each operation cannot be assigned to another machine if it starts; equations (7) and (8) gives 0-1 variables.

3. The Proposed VNMBO

3.1. Encoding Scheme

Considering the machine assignment and the operation permutation in the FJSP, a two-phase encoding scheme is adopted in the proposed VNMBO (see Figure 1). Each solution can be divided into two components, whose length equals the total number of operations. In the first section, job-based encoding is employed, and the value of each element is the code of a job (see Figure 1(a)). Operations of the same job have the same value, e.g., the third '2' represents the third operation of job 2. In the second section, machine-based encoding is used, and the value of each element is the code of the machine assigned to an operation (see Figure 1(b)).

O_{11}	O_{21}	O_{12}	O_{13}	O_{22}	O_{31}	O_{32}	O_{23}	O_{33}
1	2	1	1	2	3	3	2	3

(a) job-based encoding

O_{11}	O_{12}	O_{13}	O_{21}	O_{22}	O_{23}	O_{31}	O_{32}	O_{33}
2	1	2	1	3	4	4	3	1

(b) machine-based encoding

Figure 1. Encoding scheme

3.2. Initial Population

Like other population-based optimization algorithms, population initialization is also the crucial factor in the MBO, which can affect the convergence speed and the quality of the final solution. In this section, the machine assignment and the operation permutation are separately generated for a solution in the initial population.

(1) For the machine assignment section, the approaches proposed by Pezzella et al. [13] are adopted, which are named AssignmentRule1 and AssignmentRule2. In addition, the random selection rule is used to maintain the diversity of solutions. The adoption of a mix of these three approaches acquires the initial assignments. In this paper, 20% of initial population could be generated by AssignmentRule1, 70% by AssignmentRule2 and 10% by random selection.

(2) For the operation sequencing section, random permutation and dispatching rules are always adopted in literatures. Here, the initial operation sequencing schemes are obtained by a searching method, which was proposed by Demir and İşleyen [2]. For each generated machine assignment scheme, a fixed number of operation sequencing schemes are first generated by random permutation and four dispatching rules (SPT, LPT, MWR, MOR). The optimization objective is evaluated for each combination of the machine assignment and the operation permutation. The best one will be selected as the initial population. The four dispatching rules can be explained as follows:

SPT: Select the job with the shortest processing time.
LPT: Select the job with the largest processing time.
MWR: Select the job with the most amount of work remaining.
MOR: Select the job with the most number of operations remaining.

3.3. Neighborhood Structure

In the MBO, the improvement process of solutions is implemented by searching neighbors. Therefore, three types of neighborhood structures are employed in our algorithm to acquire the neighboring solutions. The first two neighborhood structures attempt to change the operation sequence, and the third tries to alter the machine assignment in the candidate solution. The construction processes of these neighborhood structures are presented as follows:

(1) Neighborhood structure N_1

Two positions are chosen at random in the first section of the current solution. Then, the back operation is inserted into the front of another one to acquire a neighboring solution (see Figure 2).

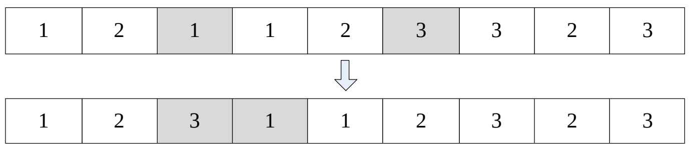

Figure 2. Neighborhood structure N_1

(2) Neighborhood structure N_2

First, two operations corresponding to different jobs are selected randomly from the first section of the current solution. Then, the positions of the selected operations are interchanged with each other (see Figure 3).

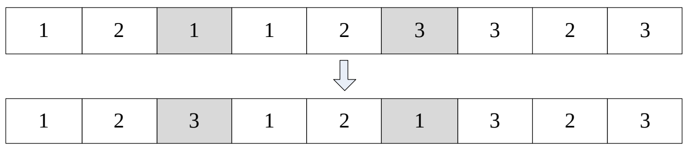

Figure 3. Neighborhood structure N_2

(3) Neighborhood structure N_3

Neighborhood structure N_3 is generated based on the mutation operator proposed by Demir and İşleyen [2]. An operation, which can be processed on more than one machine, is selected from the second section at random. The current machine is replaced by the one with the shortest processing time for the selected operation among all alternative machines (see Figure 4).

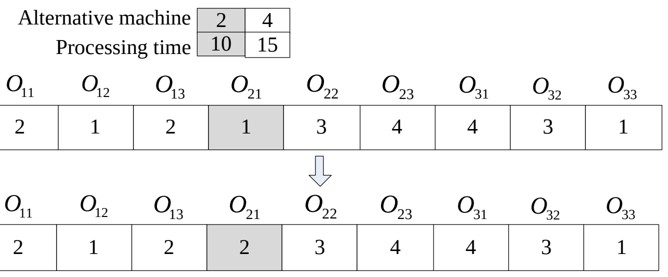
Alternative machine 2 4
Processing time 10 15
O_{11} O_{12} O_{13} O_{21} O_{22} O_{23} O_{31} O_{32} O_{33}
2 1 2 1 3 4 4 3 1
O_{11} O_{12} O_{13} O_{21} O_{22} O_{23} O_{31} O_{32} O_{33}
2 1 2 2 3 4 4 3 1

Figure 4. Neighborhood structure N_3

3.4 Individual Leaping Mechanism

To avoid the premature convergence, an individual leaping mechanism is adopted in the proposed algorithm. For the population, if the best solution is not replaced by an improved solution for lm iterations, all solutions will be reset to acquire a new initial population. In such a way, the algorithm might move away from the suboptimum and find a better solution in the long run.

3.5 Variable Neighborhood Search

Variable neighborhood search (VNS) has been successfully applied to solve various optimization problems [8,14,15,16]. It is capable of escaping from the local optima by systematically changing the neighborhood structures. In the VNMBO, on the basis of the above neighborhood structures, the proposed VNS is executed on the current best individual of the population to enhance the local search capacity.

The detailed steps of the VNS are shown as below:

Step 1: Initialization. Get the initial solution π, $\rho \leftarrow 1$, $l_{\max} \leftarrow 2$ and set the maximum iteration $\rho_{\max}$.
Step 2: Set $l \leftarrow 1$.
Step 3: Shaking. If $l = 1$, $N_1(\pi)$ and $N_3(\pi)$ are adopted simultaneously to obtain a new solution π'; if $l = 2$, $N_2(\pi)$ and $N_3(\pi)$ are applied.
Step 4: Local search. Perform a local search method to π', and get the local optima π''.
Step 5: If $C_{\max}(\pi'') < C_{\max}(\pi)$ is met, then $\pi \leftarrow \pi''$, $l \leftarrow 1$; otherwise, $l \leftarrow l+1$.
Step 6: Judge whether $l > l_{\max}$ is satisfied. If yes, then $\rho \leftarrow \rho+1$, and perform Step 7; otherwise, go to Step 3.
Step 7: Judge whether $\rho > \rho_{\max}$ holds. If yes, go to Step 8, otherwise, go to Step 2.
Step 8: Terminate the procedure.

For the local search in the VNS, the detailed steps are presented as follows:

Step 1: Initialization. Get the initial solution π' from the shaking part of the VNS, and set $\lambda \leftarrow 1$, $q_{\max} \leftarrow 3$ and the maximum iteration $\lambda_{\max}$.
Step 2: Set $q \leftarrow 1$.
Step 3: If $q = 1$, $N_1(\pi')$ is executed to obtain a new solution $\overline{\pi}$; if $q = 2$, $N_2(\pi')$ is employed; if $q = 3$, $N_3(\pi')$ is used.
Step 4: If $C_{\max}(\overline{\pi}) < C_{\max}(\pi')$ holds, then $\pi' \leftarrow \overline{\pi}$, $q \leftarrow q$; otherwise, $q \leftarrow q+1$.
Step 5: Judge whether $q > q_{\max}$ is satisfied. If yes, then $\lambda \leftarrow \lambda+1$, and go to Step 6; otherwise, go to Step 3.
Step 6: Judge whether $\lambda > \lambda_{\max}$ is met. If yes, $\pi'' \leftarrow \pi'$, go to Step 7, otherwise, go to Step 2.
Step 7: Terminate the local search.

3.6. Steps of the VNMBO

Having described the VNMBO algorithm, we give the detailed steps as follows:

Step 1: Fix the parameters of the VNMBO, including the population size $popsize$, the number of neighboring solutions k', the number of shared neighboring solutions x, the number of tours G, the age limit lm and the maximum iteration $K_{\max}$, $it_{\max}$ and $\lambda_{\max}$.
Step 2: Generate the initial population by using the method in Section 3.3.
Step 3: Set $K \leftarrow 1$, $g \leftarrow 1$, $fg \leftarrow 1$.
Step 4: Generate k' neighboring solutions randomly, improve the leader solution and fill the two shared neighboring sets S_{L} and S_{R}, which has x elements.
Step 5: For each solution π_{L} in the left line L of the V-shaped formation, randomly generate $k'-x$ neighbors according to N_1, N_2 and N_3. N_L represents the set of the newly generated $k'-x$ neighbors. The best solution in $N_{\mathrm{L}} \cup S_{\mathrm{L}}$ is used to replace the current solution π_{L}. The x best unused solutions in $N_{\mathrm{L}} \cup S_{\mathrm{L}}$ are selected to refill S_{L}.
Step 6: For each solution π_{R} in the right line R, randomly generate $k'-x$ neighbors. N_R represents the set of the newly generated $k'-x$ neighbors. The best solution in $N_{\mathrm{R}} \cup S_{\mathrm{R}}$ is used to replace the current solution π_{R}. The x best unused solutions in $N_{\mathrm{R}} \cup S_{\mathrm{R}}$ are selected to refill S_{R}.
Step 7: Perform the VNS to each solution in the current population and replace the original one if it is improved.
Step 8: Update the current best solution.
Step 9: Perform the reset mechanism if the best solution remains unchanged for lm iterations.
Step 10: Set $g \leftarrow g+1$. If $g > G$ is met, set $g \leftarrow 1$, go to Step 11; otherwise, go to Step4.
Step 11: Set a new leader bird. If $fg = 1$, move the leader to the tail of line L, and set the first solution of L as the new leader bird, and then set $fg = 0$; otherwise, move the leader to the tail of line R, and set the first solution of R as the new leader, and then set $fg = 1$.

Step 12: Set $K \leftarrow K+1$, if $K > K_{\max}$ is met, go to Step 13; otherwise, go to Step 4.
Step 13: Terminate the procedure.

4. Experiment Validations

In this section, two types of benchmark problems are taken from Kacem et al. [6,7] (Kacem data) and Brandimarte [1] (BRdata) are used to test the performance of the proposed VNMBO. For each benchmark instance, 10 independent replications were conducted. As recommended by Duman et al. [3], four parameters are fixed as follows: the population size $popsize$ =51; the number of neighboring solutions k' = 3; the number of the shared neighboring solutions x =1; and the number of tours G =10. In addition, the other parameters of the algorithms are selected based on the comparison of experimental data under different combinations of them. Thus, we set lm =10, $\rho_{\max}$ =30 and $\lambda_{\max}$ =10. The value of $K_{\max}$ is presented in table 1 for each different problem.

4.1 Effectiveness of the Individual Leaping Mechanism

The effectiveness of the individual leaping mechanism is first verified here. By excluding the mechanism from the VNMBO, we named the obtained algorithm as VNMBO-1. The average makespan $AVC_{\max}$ and the standard deviation SD for each instance are listed in table 1.

Table 1. Comparison Results of VNMBO and VNMBO-1

Instance	$n\times m$	K_{max}	VNMBO		VNMBO-1	
			AVC_{max}	SD	AVC_{max}	SD
Kacem01	4×5	*nm*	11.0	0.00	11.0	0.00
Kacem02	8×8	*nm*	14.0	0.00	14.4	0.52
Kacem03	10×7	*nm*	11.0	0.00	11.2	0.42
Kacem04	10×10	*nm*	7.0	0.00	7.6	0.52
Kacem05	15×10	*nm*	12.6	0.52	14.0	0.82
MK01	10×6	2*nm*	40.8	0.79	41.7	0.67
MK02	10×6	2*nm*	29.0	0.82	29.9	0.99
MK03	15×8	*nm*	204.0	0.00	204.3	0.95
MK04	15×8	2*nm*	67.2	0.63	68.3	1.34
MK05	15×4	3*nm*	176.9	1.60	179.1	1.91
MK06	10×15	4*nm*	71.0	2.45	74.6	3.34
MK07	20×5	4*nm*	148.7	1.57	151.6	2.55
MK08	20×10	*nm*	523.0	0.00	523.0	0.00
MK09	20×10	4*nm*	307.0	5.93	314.6	9.10
MK10	20×15	3*nm*	241.1	3.11	251.3	4.22

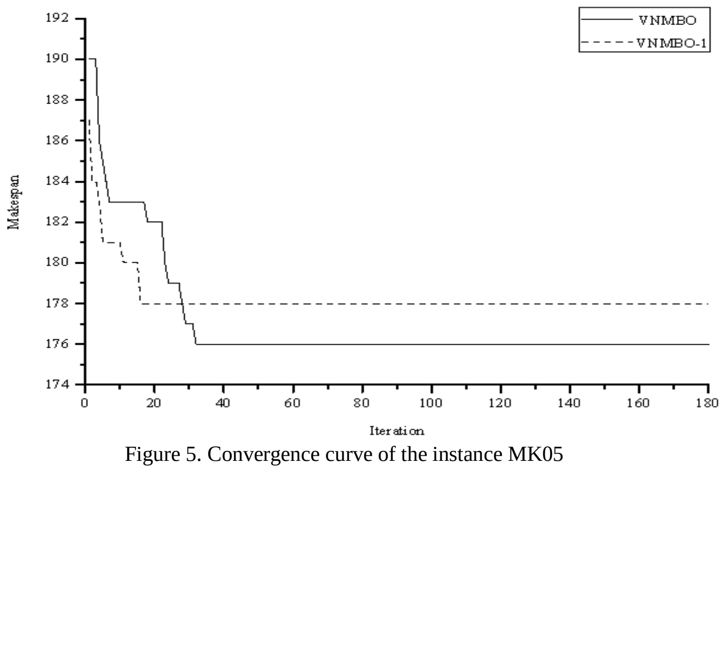

Figure 5. Convergence curve of the instance MK05

It can be seen from Table 1 that the leaping mechanism can improve the computational results $AVC_{\max}$ of VNMBO. It makes VNMBO yield two results equal to VNMBO-1 and thirteen better than VNMBO-1. In addition, VNMBO generates smaller *SD* values than VNMBO-1, which shows that the proposed leaping mechanism can enhance the robustness of the VNMBO to the initial populations. Figure 5 shows the convergence curve of the instance MK05 by using the two algorithms, which shows that the convergence feature of VNMBO is better than VNMBO-1. The leaping mechanism can avoid the premature convergence. As a whole, these findings demonstrate the effectiveness of the individual leaping mechanism.

4.2 Effectiveness of the Proposed VNMBO

The effectiveness of the proposed VNMBO for solving the FJSP is evaluated based on the Kacem data and BRdata. Table 2 and Table 3 show the comparison results of best makespan values ($BC_{\max}$) between the proposed VNMBO and other published algorithms: AL+CGA [6], tabu search (TS) [7], PSO+TS [9], KBACO [18], effective and distributed PSO (edPSO) [11], heuristic algorithm (HA) [19]. The computational results of compared algorithms have been reported in literatures. The symbol '-' means that the related value is not given. In addition, Figures 6 and 7 illustrate the Gantt charts of the instances MK01 and MK07.

Table 2. Comparison of Results on Kacem Instances

Instance	AL+CGA	PSO+TS	KBACO	edPSO	HA	VNMBO
Kacem01	-	12	11	11	11	11
Kacem02	15	-	14	17	15	14
Kacem03	-	11	11	-	13	11
Kacem04	7	7	7	8	7	7
Kacem05	23	-	11	-	12	12

Table 3. Comparison of Results on Brandimarte Instances

Instance	TS		KBACO		edPSO		HA		VNMBO	
	BC_{max}	*RPD*	BC_{max}	*RPD*	BC_{max}	*RPD*	BC_{max}	*RPD*	BC_{max}	*RPD*
MK01	42	7.7	39	0	41	5.1	42	7.7	40	2.6
MK02	32	23.1	29	11.5	26	0	28	7.7	28	7.7
MK03	204	0	204	0	207	1.4	204	0	204	0
MK04	81	24.6	65	0	65	0	75	15.4	66	1.5
MK05	186	8.8	173	1.2	171	0	179	4.7	173	1.2
MK06	86	41.0	67	9.8	61	0	69	13.1	66	8.2
MK07	157	9.0	144	0	173	20.1	149	3.5	144	0
MK08	523	0	523	0	523	0	555	6.1	523	0
MK09	369	24.2	311	4.7	307	3.4	342	15.2	297	0
MK10	269	17.5	229	0	312	36.2	242	5.7	237	3.5
AVRPD		15.6		2.7		6.6		7.9		2.5

According to table 2 and table 3, it can be seen that:

(1) For the five Kecam instances in table 2, our VNMBO achieves the optimal solutions in Kacem01~04. In the instance Kacem05, there is just one time unit difference between the best makespan obtained by our algorithm and the published KBACO.

(2) For the ten Brandimarte instances in table 3, $BC_{\max}$ and RPD are compared between our algorithm and others. RPD represents the relative percent deviation which is calculated as follows: $RPD = 100 \times (BC_{\max} - BC'_{\max}) / BC'_{\max}$ where $BC_{\max}$ is the best makespan obtained by each algorithm, $BC'_{\max}$ is the minimum of $BC_{\max}$ among all algorithms. As shown in table 3, the average relative percent deviation $AVRPD$ of the presented VNMBO outperforms other four algorithms. Figure 8 shows a graphical comparison of the RPD values of the methods for different instances.

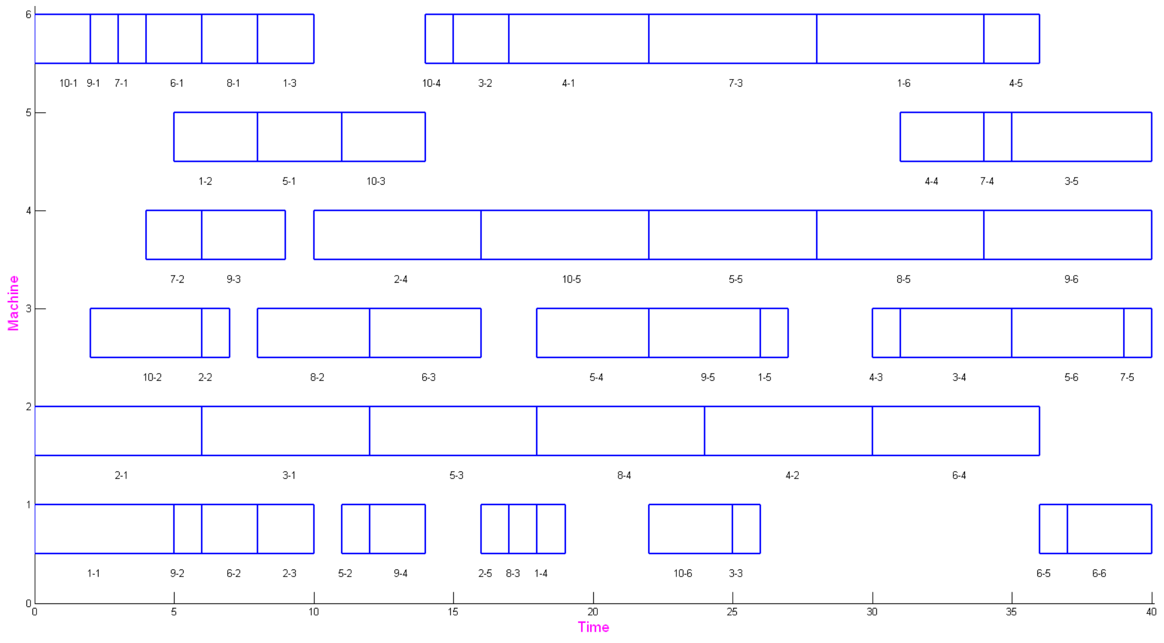
Figure 6. The Gantt chart of the MK01

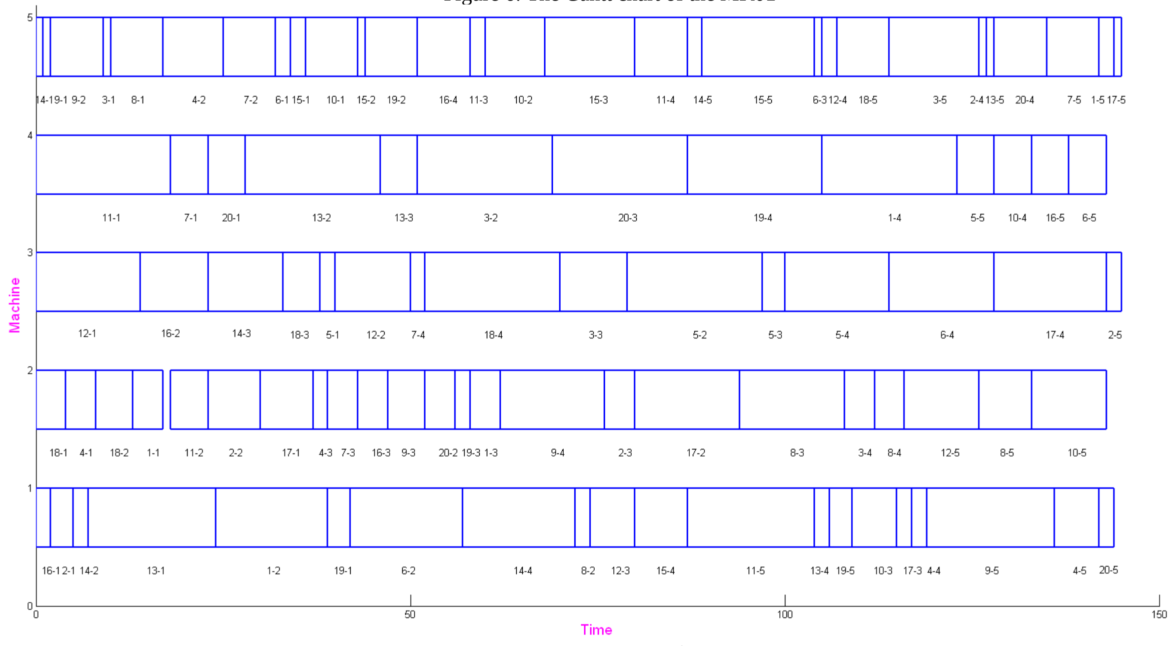
Figure 7. The Gantt chart of the MK07

5. Conclusions

For the characteristics of the flexible job shop, a hybridization algorithm based on MBO and VNS is proposed to solve the FJSP. To ensure the quality of the initial solutions, a two-stage population initialization scheme is designed. Then, an individual leaping mechanism is developed to avoid the premature convergence. To search the solution space, three neighborhood structures are developed to generate the neighborhood solutions, and base on which a variable neighborhood search algorithm is proposed to enhance the local search capacity. Finally, extensive computational data reveal that VNMBO is effective for solving the FJSP with the criterion to minimize the makespan. For further study, we give the following suggestions:

(1) Developing other hybrid algorithms such as MBO-SA and MBO-TS for FJSP.
(2) Consider other constraints in the model such as setup time of machines, random processing time of operations, and precedence constraints among operations of different jobs, etc.
(3) Applying the MBO algorithm to solve a multi-objective FJSP.

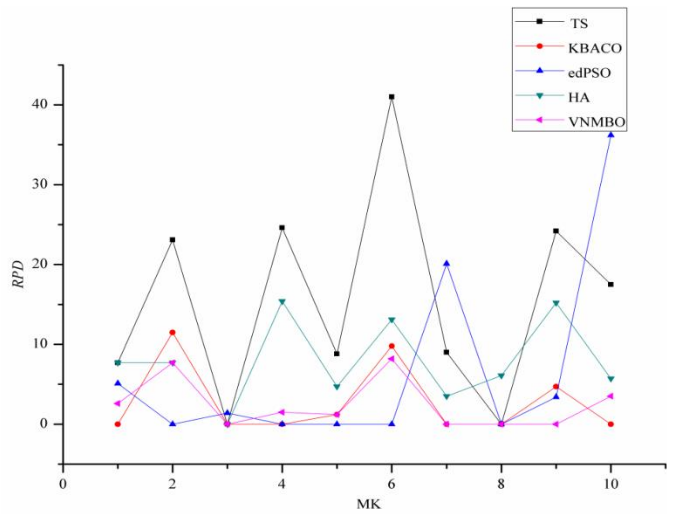

Figure 8. Comparison of the RPD values

Acknowledgements

The authors would like to thank the editors and the anonymous reviewers for their helpful comments and suggestions, which have improved the presentation. This work was supported in part by the Science and Technology Plan Projects of Henan Province of China under grant No.152102210357 and No.152102210149, the Youth Backbone Teachers Funding Planning Project of Colleges and Universities in Henan Province of China under grant No.2014GGJS-084, the Key Science Research Project of Colleges and Universities in Henan Province of China under grant No.16A520030, the Youth Backbone Teachers Training Targets Funded Project of Zhengzhou University of Light Industry of Henan Province of China under grant No.XGGJS02, the Ph.D.Research Funded Project of Zhengzhou University of Light Industry of Henan Province of China under grant No.2010BSJJ038 and No.2014BSJJ080, and the National Science Foundation of China under grant No.81501548.

References

1. P. Brandimarte, “Routing and scheduling in a flexible job shop by tabu search”, *Annals of Operations research*, vol.41, no.3, pp. 157-183, 1993.
2. Y. Demir, S. K. İşleyen, “An effective genetic algorithm for flexible job-shop scheduling with overlapping in operations”, *International Journal of Production Research*, vol.52, no.13, pp. 3905-3921, 2014.
3. E. Duman, M. Uysal, 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.
4. L. Gao, 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.
5. S. M. Johnson, “Optimal two- and three-stage production schedules with setup times included”, *Naval research logistics quarterly*, vol.1, no.1, pp.61-68, 1954.
6. I. Kacem, S. Hammadi, P. Borne, “Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems”, *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol.32, no.1, pp.1-13, 2002.
7. I. Kacem, S. Hammadi, P. Borne, “Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic”, *Mathematics and computers in simulation*, vol.60, no.3, pp.245-276, 2002.
8. D. Lei, X. Guo, “Variable neighbourhood search for dual-resource constrained flexible job shop scheduling”, *International Journal of Production Research*, vol.52, no.9, pp.2519-2529, 2014.
9. J. Li, Q. K. Pan, S. X. Xie, “A hybrid particle swarm optimization and tabu search algorithm for flexible job-shop scheduling problem”,*International Journal of Computer Theory and Engineering*, vol.2, no.2, pp.1793-1801, 2010.
10. T. Meng, Q. K. Pan, J. Q. Li,“An improved migrating birds optimization for an integrated lot-streaming flow shop scheduling problem”, *Swarm and Evolutionary Computation*, in press, 2017.
11. M. Nouiri, A. Bekrar, A. Jemai, “An effective and distributed particle swarm optimization algorithm for flexible job-shop scheduling problem”, *Journal of Intelligent Manufacturing*, 2015. doi: 10.1007/s10845-015-1039-3.
12. Q. K. Pan, Y. Dong, “An improved migrating birds optimisation for a hybrid flowshop scheduling with total flowtime minimisation”, *Information Sciences*, vol.277, pp. 643-655, 2014.
13. F. Pezzella, G. Morganti, G. Ciaschetti, “A genetic algorithm for the flexible job-shop scheduling problem”, *Computers & Operations Research*, vol.35, no.10, pp. 3202-3212, 2008.

14. M. Prandtstetter, G. R. Raidl,"An integer linear programming approach and a hybrid variable neighborhood search for the car sequencing problem", *European Journal of Operational Research*, vol.191, no.3, pp.1004-1022, 2008.
15. A. Sifaleras, I. Konstantaras, N. Mladenović, "Variable neighborhood search for the economic lot sizing problem with product returns and recovery", *International Journal of Production Economics*, vol.160, pp.133-143, 2015.
16. L. Wei, Z. Zhang, D. Zhang, "A variable neighborhood search for the capacitated vehicle routing problem with two-dimensional loading constraints", *European Journal of Operational Research*, vol.243, no.3, pp.798-814, 2015.
17. Z. P. Xie, Y. Jia, C. Y. Zhang, "Migrating birds optimization for blocking flow shop scheduling with total flowtime minimization", *Computer Integrated Manufacturing Systems*, vol.18, no.8, pp. 2099-2107, 2015.
18. L. N. Xing, Y. W. Chen, P. Wang, "A knowledge-based ant colony optimization for flexible job shop scheduling problems", *Applied Soft Computing*, vol.10, no.3, pp.888-896, 2010.
19. M. Ziaee, "A heuristic algorithm for solving flexible job shop scheduling problem", *The International Journal of Advanced Manufacturing Technology*, vol.71, no.4, pp.519-528, 2014.