An Effective Hybrid Particle Swarm Optimization for Flexible Job Shop Scheduling Problem

Guohui Zhang[1,*], Lingjie Zhang[2], Yongcheng Wang[1] and Lihui Wu[3]

[1]School of Management Science and Engineering, Zhengzhou Institute of Aeronautical Industry Management, Zhengzhou, 450015, China

[2]Department of Electrical and Information Engineering, Henan Vocational College of Water Conservancy and Environment, Zhengzhou, 450008, China

[3]School of Mechanic & Electrical Engineering, Henan University of Technology, Zhengzhou, 450052, China

Abstract: The job shop scheduling problem is one of the most arduous combinatorial optimization problems. Flexible job shop scheduling problem (FJSP) is an important extension of the classical job shop scheduling problem, where the same operation could be processed on more than one machine. This paper proposed a new effective approach based on the hybridization of the particle swarm optimization (PSO) and local search algorithm of variable neighborhood search (VNS) to solve the FJSP for minimizing the makespan, the maximal machine workload, and the total workload of machines. PSO is a highly efficient evolutionary computation technique inspired by bird's flight and communication behaviors. PSO integrating the local search and global search has highly search ability. VNS has the strong local search ability. Benchmark problems are used to evaluate and study the performance of the proposed algorithm. Computational results show that the proposed hybrid algorithm is an efficient and effective approach.

Keywords: Particle swarm optimization, Variable neighborhood search, Local search, Multi objective, Flexible job shop scheduling problem.

1. INTRODUCTION

Flexible job shop scheduling problem (FJSP) is very important in both fields of production management and combinatorial optimization. Mati and Xie proved that FJSP with two machines is NP-hard problem [1].

In 1990, Bruker and Schile [2] were the first to describe this problem. Bruker and Schile [2] developed an exact algorithm for a two jobs problem. However exact algorithms are not effective for solving the large scale combinational problem and FJSP, especially for large instances. Then, many researchers come to focus on developing heuristic procedures in recent years for FJSP, such as ant colony optimization (ACO) [3], genetic algorithm (GA) [4].

In practical industry, multi-objective scheduling problem is more suitable for the practical needs. Kacem *et al*. [5] solved the multi-objective FJSP considering minimization of total workload and maximum workload. Zhang [6] applied a hybrid algorithm of particle swarm optimization (PSO) and tabu search (TS) as a local search algorithm to solve the multi objective FJSP. Amiri *et al*. [7] used variable neighborhood search algorithm to solve the FJSP. Wang *et al*. [8] proposed a modified algorithm based on immune and entropy principle for multi objective FJSP. Rahmati *et al*. [9] developed two evolutionary algorithms to solve the multi-objective FJSP. Bagheri and Zandieh [10] proposed an integrated approach based on the variable neighborhood search algorithm to solve bi-criteria FJSP.

In this paper, an improved particle swarm optimization combined with variable neighborhood search as a local search algorithm is proposed to solve the multi objective FJSP. Particle swarm optimization allows an extensive search of solution space while the local search algorithm is employed to reschedule the results obtained from the PSO to find the better solutions. The proposed hybrid algorithm could counterbalance the shortcomings of single algorithm, and keep the balance between the diversification and the intensification during the search process.

The remainder of this paper is organized as follows. Section 2 describes the multi objective problem and the flexible job shop scheduling problem. Section 3 presents the proposed hybrid algorithm including the basic PSO, the basic VNS, hybrid algorithm framework, representation of individual, neighborhood structure. In Section 4, benchmark problems were performed with our proposed approach, and the computational results were reported followed by the comparison to other heuristic methods. Some concluding remarks are made in Section 5.

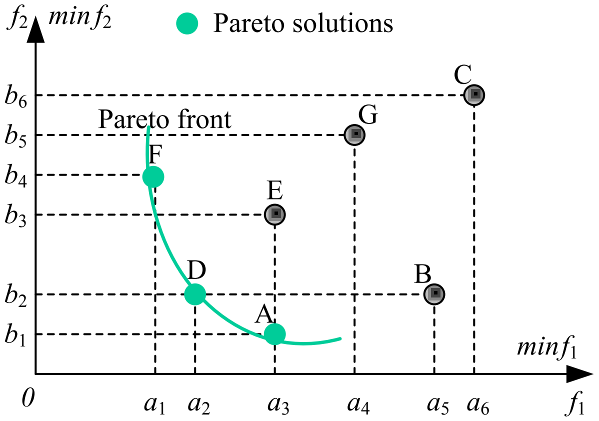

Fig. (1). Relationship among the Solutions in the Bi-objective Problem.

2. PROBLEM DESCRIPTION

2.1. Multi-objective Optimization

A general multi objective optimization problem consists of a number of objectives and is associated with a number of inequality and equality constraints. Non-dominated solutions are often used and important concept in multi objective optimization case.

Let's consider a bi-objective functions minimization problem, $f_1(x)$ and $f_2(x)$. From the Fig. (**1**), based on the above concept, it could be easily seen that the solution *A* dominates the solution *B*, *for* $f_1(a_3)<f_1(a_5)$, and $f_2(b_1)<f_2(b_2)$. The solution *D* do not dominate the solution *F*, *for* $f_1(a_1)<f_1(a_2)$, but $f_2(b_2)< f_2(b_4)$. Similarly, the solution *A* do not dominate the solution *F*. Then, among the solutions *A, B, C, D, E, F, and G,* the solutions *A*, *D* and *F* are the non-dominated solutions. In some literature, the solutions *A, D* and *F* were called the Pareto front solutions.

2.2. Flexible Job Shop Scheduling

The flexible job shop scheduling problem could be described as follows. There are a set of *n* jobs J={J_1, J_2,…, J_n} and a set of *m* machines M={M_1, M_2, …, M_m}. Each job J_i ($1≤i≤n$) consists of a pre-determined sequence of n_i operations. For each operation O_{ij} ($1≤j≤n_i$), there is a set of alternative machines set M_{ij} for performing it. The processing time p_{ijk} of an operation O_{ij} on each machine is predefined and fixed. The completion time of each job is equal to the last operation completion time C_{ini}. Each operation could not be interrupted during its performance. And each machine could perform at most one operation at any time.

Hypotheses considered in this paper are summarized as follows:

All machines are available at time 0;

All jobs are released at time 0;

Each machine can process only one operation at a time;

Each operation can be processed only by one machine at a time;

Each operation can be processed without interruption on one of a set of available machines;

Recirculation occurs when a job could visit a machine more than once;

The order of operations for each job is predefined and cannot be modified.

In this paper, the considered objective functions are to minimize the following criteria:

f_1: the maximal completion time of machines, i.e., the makespan;

f_2: the maximal machine workload, i.e., the maximum working time spent at any machine;

f_3: the total workload of machines, which represents the total working time over all the machines.

In the last decade, two approaches were mainly adopted to solve the multi-objective FJSP. One approach is the aggregation-based approach which deals with multi objective values to one objective value. The linear weighted summation is a common aggregation function. Another approach is Pareto approach. It could easily keep trade-off between objectives based on the Pareto dominance relation. In this study, the aggregation-based approach is adopted to solve the multi-objective FJSP. The two objective functions of maximal machine workload and total workload of machines are closely related with the makespan. Therefore, the weight of makespan is the biggest among them.

3. HYBRID PSO AND VNS FOR FJSP

3.1. Particle Swarm Optimization

Particle swarm optimization (PSO) is an evolutionary computation technique proposed by Kennedy and Eberhart in 1995 [11]. PSO algorithm is initialized with a population of random candidate solutions. Each single solution is like a “bird” in D-dimensional search space, conceptualized as “particle”. Each individual in the search space with a velocity which is dynamically adjusted according to its own flying experience and its companions’ flying experience. Each

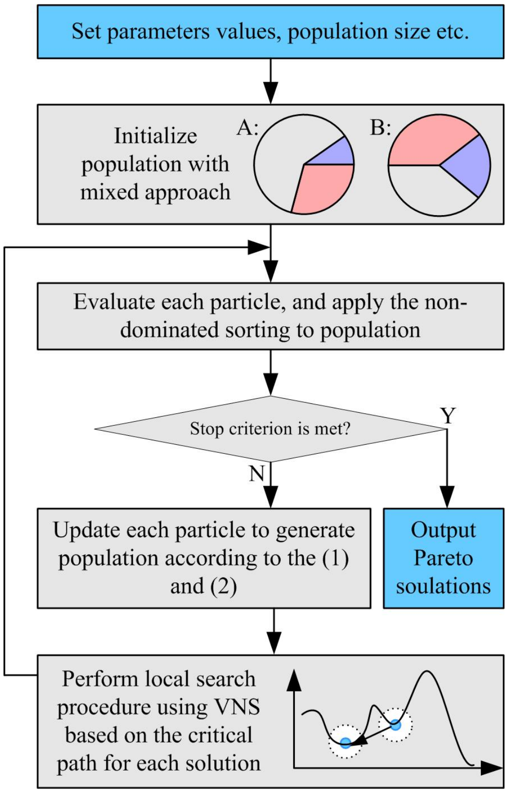
Set parameters values, population size etc.
Initialize population with mixed approach
A:
B:
Evaluate each particle, and apply the non-dominated sorting to population
Stop criterion is met?
Y
N
Update each particle to generate population according to the (1) and (2)
Output Pareto soulations
Perform local search procedure using VNS based on the critical path for each solution

Fig. (2). Framework of Proposed Hybrid Algorithm.

individual is treated as a volume-less particle (a point) in the D-dimensional search space. During each iteration, each particle is updated iteratively according to the following two equations:

$$v_{td} = w \times v_{td} + c_1 \times rand(\ ) \times (p_{ld} - x_{td}) + c_2 \times Rand(\ ) \times (p_{gd} - x_{td}) \quad (1)$$

$$x_{td} = x_{td} + v_{td} \quad (2)$$

where t is the t-th particle; v_{td} is the velocity of particle t in iteration d; x_{td} is the position of particle t in iteration d; w is the inertial weight. It is used to control the amount of the previous velocity between the global exploration and local exploitation abilities of the swarm. c_1 and c_2 are two positive constants; *rand*( ) and *Rand*( ) are two random functions in the range [0, 1]; p_{ld} is the best previous position of particle t so far, also called *pbest*, i.e., the local best solution; and p_{gd} is the best previous position among all the particles, also called *gbes*t, i.e., the global best solution.

3.2. Variable Neighborhood Search

VNS was first proposed by Mladenovic [12] in 1995. VNS is based on a simple principle: systematic change of neighborhood structure within a possibly local search, both in descent to local minima and in escape from the valleys which contain them.

During the initialization step, let us denote a finite set of pre-selected neighborhood structures with N_k, (k =1, ..., k_{max}), and with $N_k(s)$ the set of solutions in the k-th neighborhood of s. Most local search heuristics use one neighborhood structure, i.e., k_{max} =1. The neighborhood structure is systematically changed and the shake procedure works to switch to another region of the search space so as to carry out a new local search. The main step could possibly be iterated until some other stopping condition is met such as maximum number of iterations, or maximum number of successive iterations without improvement of best solution, or the maximal allowed CPU time.

3.3. Hybrid Algorithm Framework

The proposed hybrid VNS algorithm could be keep the balance between the diversification and the intensification during the search process. Fig. (**2**) gives the hybrid algorithm framework.

During the operation process, the initialization population is firstly generated by a mixed approach. Then, each individual in the population is evaluated, and apply the non-dominated sorting to population. If the stop criterion is met, the non-dominated solution is output. Otherwise, update the each particle according to the Eqs. (1) and (2). Then, perform local search procedure using the VNS based on the critical path to update the each member. The new particle swarm is generated. The stop criterion is that the max iteration generation is reached.

For the FJSP problem, a number of solutions with different maximal work-loads or total workloads may have the same makespan. From this point of view, when the particle is

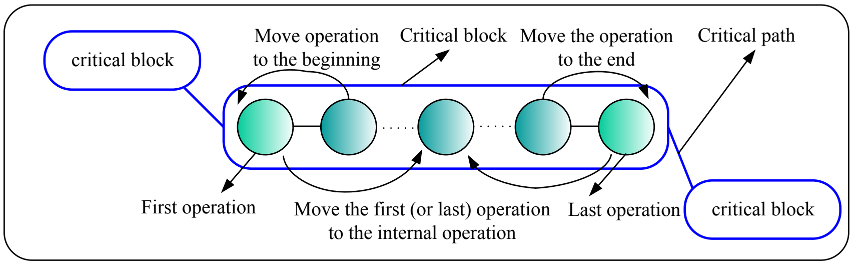
critical block
Move operation to the beginning
Critical block
Move the operation to the end
Critical path
First operation
Move the first (or last) operation to the internal operation
Last operation
critical block

Fig. (3). Neighborhood Structure I.

updated, we firstly find the solutions with the minimum makespan. Then minimize the maximal workload and the total workload in the presence of the minimum makespan.

3.4. Representation of Individuals

An efficient representation of the individuals which respects all constraints of FJSP is adopted. Two common encoding schemes for solving the FJSP are the three-tuple scheme which represents job, operation, machine in each tuple [13] and A-B string scheme which denotes routing and sequencing respectively [6]. They represent the same information, but in the implementation process is different. In this study, A-B string is also adopted.

Machine selection part: An array of integer values is used to present machine selection. The length is the sum of all operations of all jobs. Each integer value equals to the index of alternative machines set of each operation.

Operation sequence part: The operation-based representation is used. All operations for the same job are defined with the same symbol, then interpret them according to the order of occurrence in the sequence of individual. By this way, it can avoid creating an infeasible schedule through replacing each operation by the corresponding job index.

3.5. Initial Population

A good initial population locates promising areas in the search space and provides enough diversity to avoid premature convergence. Three initial rules are used to generate initial machine selection part [14]. (1) Random selection rule. For each operation, randomly select a machine from the alternative machines set, then place it at the position in the machine selection part. (2) Global selection rule. (3) Local selection rule. In our proposed algorithm, the initial machine selection part of 10% individuals in the population are generated by rule (1), 60% individuals by rule (2), and 30% individuals by rule (3).

Once the machines are assigned to all the operations, that is, the machine selection part is fixed. Then all the operations will be sequenced. We apply there rules to generate initial operation sequence part [13]. (1) Random dispatching rule. Randomly generate the sequence of the operations on each machine. (2) Most time remaining rule. Sequence the operations according to the remaining processing time of jobs, that is, the job with the most remaining time will be selected first. (3) Most number of operations remaining rule. The operations are sequenced according to the number of succeeding operations in the same job. That is, the most number of remaining operations, the high priority of the operation is selected. In our algorithm, the initial operation sequence part of 20% individuals in the population are generated by rule (1), 40% individuals by rule (2), and 40% individuals by rule (3).

3.6. Neighborhood Structure

The crucial of VNS is how to define the effective neighborhood structure. In this study, two neighborhood structures Neighborhood structure I and Neighborhood structure II were adopted based on the critical path. Moving different number critical operation on the critical path could get different neighborhoods.

Neighborhood structure I. Neighbor solutions could be generated by moving one critical operation on the critical path.

The procedure of generating neighbor solutions is described as following.

(1) In each critical block, the first (or last) operation is inserted into the internal operation within the critical block.

(2) In each critical block, the internal operation is moved to the beginning or the end of the critical block.

(3) If a critical block contains only one operation, then no swap is made.

It is also used for the FJSP problem by moving one operation in Fig. (**3**).

Neighborhood structure II. Neighbor solutions could be generated by moving two critical operations on the critical path. For FJSP, we have to make an extra check that the two operations to be swapped do not belong to the same job.

The rules of swapping two operations on the critical path are as follows.

(1) If the first (or last) critical block contains more than two operations, we only swap the last (or first) two operations in the critical block. Otherwise, if the first (or last) critical block contains only two operations, these operations are swapped.

(2) In each critical block, we only swap the last two and first two operations.

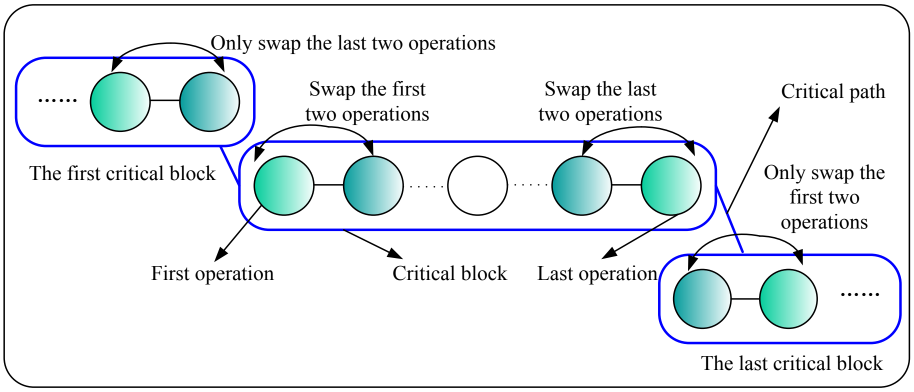
Only swap the last two operations
......
Swap the first two operations
Swap the last two operations
Critical path
The first critical block
Only swap the first two operations
First operation
Critical block
Last operation
......
The last critical block

Fig. (4). Neighborhood Structure II.

Table 1. The Computational Results.

Algorithm	S	4×5			8×8			10×10			15×10		
		f_1	f_2	f_3	f_1	f_2	f_3	f_1	f_2	f_3	f_1	f_2	f_3
AL+CGA	$S1$	16	9	35	15	13	79	7	5	45	23	11	95
	$S2$	16	10	34	16	13	75	8	5	42	24	11	91
	$S3$	18	7	33				8	7	41			
	$S4$	18	8	32									
PSO+SA	$S1$				15	12	75	7	6	44	12	11	91
	$S2$				16	13	73						
hGA	$S1$				14	12	77	7	5	43	11	11	91
Proposed algorithm	$S1$	11	9	34	14	12	77	7	5	43	11	11	91
	$S2$	11	10	32	15	11	83	7	6	42	11	10	93
	$S3$	12	8	32	15	12	75	8	5	42			
	$S4$	13	7	33	16	11	77	8	7	41			
	$S5$				16	13	73						

(3) If a critical block contains only one operation, then no swap is made.

The scheme of swapping between two operations is shown in Fig. (**4**).

The two type neighborhood structures were used only to improve the efficiency of local search. According to the VNS principle, the neighborhood structure I repeatedly works until the local optimum is met. Then, neighborhood II was executed. And, once an improved solution is found in the type II, the type I is used again for local search.

4. COMPUTATIONAL RESULTS

The proposed algorithm was implemented in C++ on a Core 2 Duo running at 2.0 GHz. And, data sets of problem instances are from Kacem et al. [15] (KaData). KaData includes 5 instances, 4×5, 8×8, 10×10, 15×10 (n×m, n denotes number of jobs, m denotes number of machines). In order to evaluate the efficacy, the test problems were run 5 times consecutively, and get their average value.

Using the above mentioned 5 KaData instances, we compare the proposed effective hybrid algorithm of PSO and VNS (PSOVNS) with the existing algorithm including the AL+CGA, PSO+SA, and hGA. AL+CGA is the algorithm by Kacem *et al*. [15]. PSO+SA is the algorithm by Xia and Wu [16]. hGA is the algorithm by Gao *et al*. [17]. Table **1** gives the computational results. The computational results of the compared algorithms are cited from the original literatures in which the algorithms were first proposed. From the computational results, it could prove that our proposed algorithm is effective.

In Table **1**, the first column lists the compared algorithm of AL+CGA, PSO+SA, hGA, and proposed algorithm. The second column presents the index of best solutions. The third

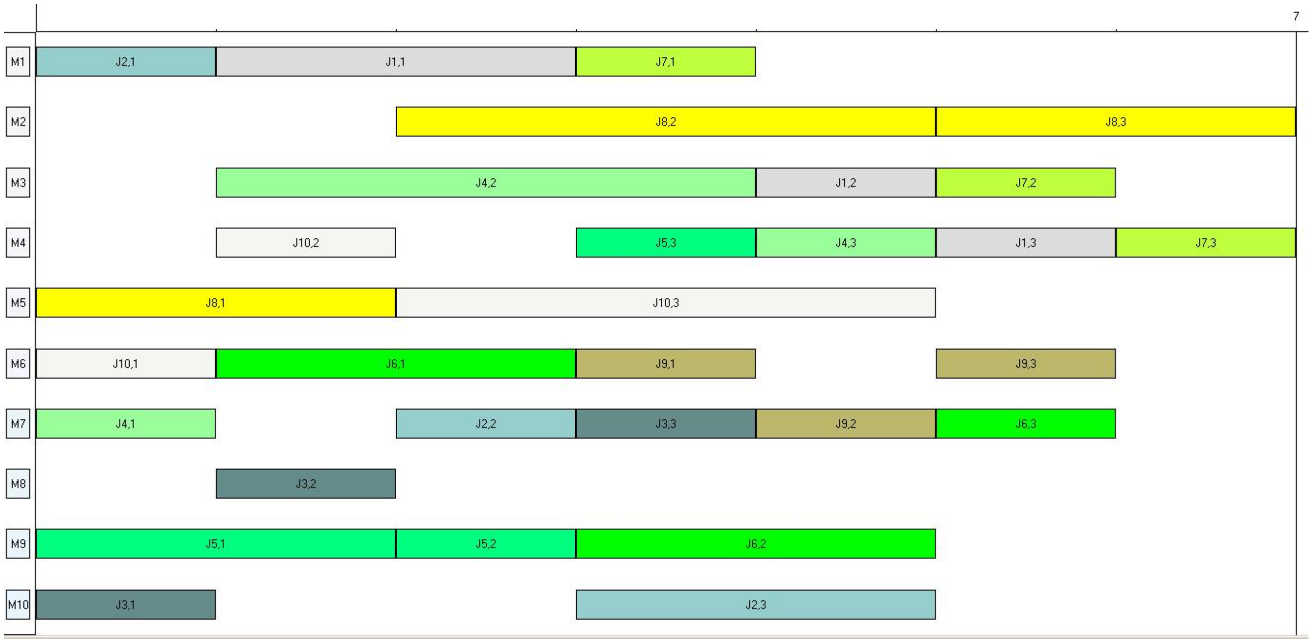

Fig. (5). The Gantt Chart of the $S1$ (f_1=7, f_2=5, f_3=43) for the 10×10 Problem.

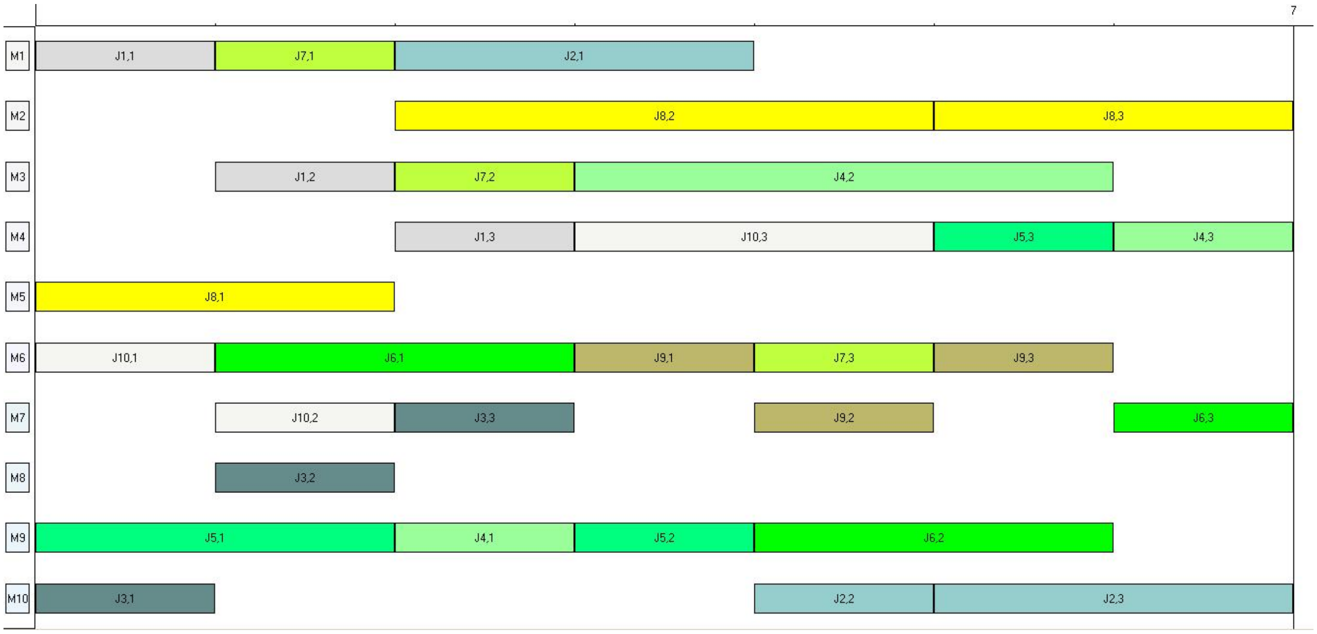

Fig. (6). The Gantt Chart of the $S2$ (f_1=7, f_2=6, f_3=42) for the 10×10 Problem.

column up to the fourteenth one represent the best results of the four problems, 4×5, 8×8, 10×10, and 15×10 respectively.

From Table **1**, in the 4×5 problem, the solutions of our proposed algorithm all dominate the solutions of the AL+CGA. In the 8×8 problem, the solutions of PSOVNS all dominate the solutions of AL+CGA. The Pareto solutions of PSO+SA have two solutions, and hGA have one solutions. However, PSOVNS have four solutions, and the solutions are never dominated. Thus, for small-sized problems, the proposed hybrid algorithm has no dominating solution but is not dominated by them. In the 10×10 problem, the solutions of our algorithm also all dominate the solutions of AL+CGA and PSO+SA, and there are only one solution of AL+CGA and PSO+SA. The proposed approach could get four non-dominated solutions. And, the solution of hGA is one of the four solutions. Then, in the 15×10 problem, the solutions of our approach dominate the solutions of AL+CGA and PSO+SA. Fig. (**5**, **6**, and **7**) show the Gantt chart of 10×10 problem.

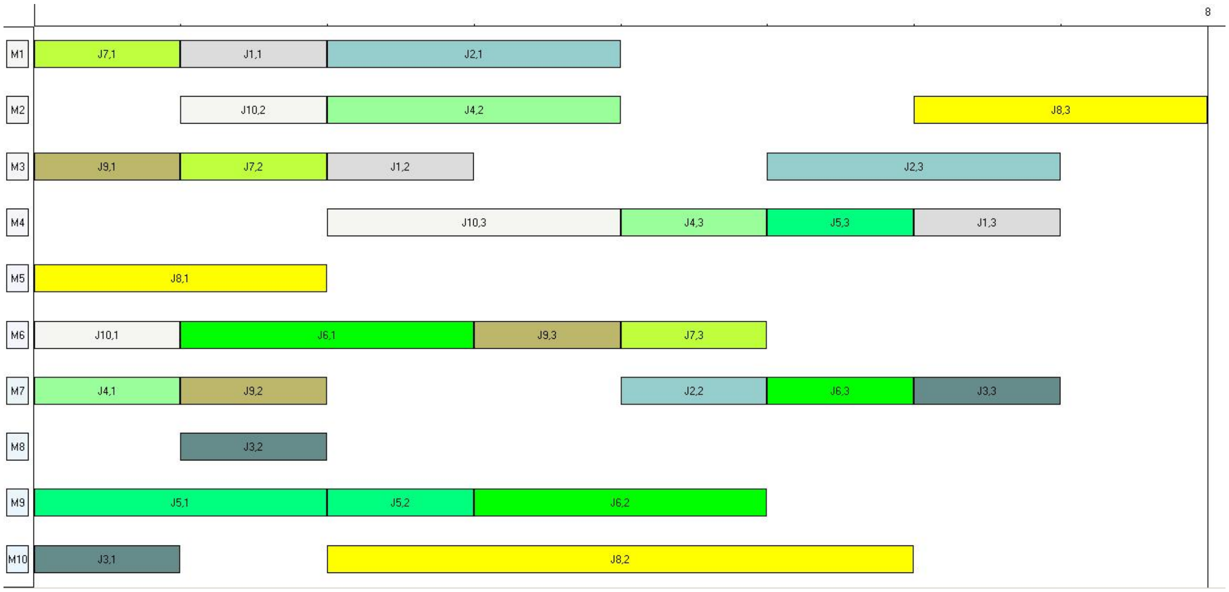

Fig. (7). The Gantt Chart of the $S3$ (f_1=8, f_2=5, f_3=42) for the 10×10 Problem.

CONCLUSION

In this paper, an effective particle swarm optimization and variable neighborhood search algorithm is proposed to solve the multi objective FJSP. Two neighborhood structures are adopted to enhance the ability of local search through systematic changing the neighborhood structure according to the critical operations of the critical path, and avoid the proposed hybrid algorithm into premature. To evaluate the performance of the proposed algorithm, four instances were optimized. Computational results show that the proposed algorithm is efficient and effective approach for solving the multi objective FJSP.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflicts of interest.

ACKNOWLEDGEMENTS

This paper presents work funded by the Program for Science & Technology Innovation Talents in Universities of Henan Province (No. 14HASTIT006), the National Natural Science Foundation of China (No. 61203179, No. 71201048), Foundation for University Key Teacher of Henan Province (No. 2014GGJS-105), and the Aviation Science Funds (No. 2014ZG55016).

REFERENCES

[1] Mati Y and Xie X. “The complexity of two-job shop problems with multi-purpose unrelated machines”. *European Journal of Operational Research* 2004; 153, 159-169.

[2] Brucker P and Schile R. “Job-shop scheduling with multi-purpose machines”. *Computing* 1990; 45(4): 369–375.

[3] Xing LN, Chen YW, Wang P, Zhao QS, and Xiong J. “A knowledge-based ant colony optimization for flexible job shop scheduling problems”. *Applied Soft Computing* 2010; 10(3): 888-896.

[4] Jia HZ, Nee AYC, Fuh JYH, Zhang YF. “A modified genetic algorithm for distributed scheduling problems”. *International Journal of Intelligent Manufacturing* 2003; 14(3-4): 351-362.

[5] Kacem I, Hammadi S, Borne P. “Approach by localization and multi-objective evolutionary optimization for flexible job-shop scheduling problem”. *IEEE Transactions on Systems, Man, and Cybernetics* Part C 2002; 32, 1-13.

[6] Zhang GH, Shao XY, Li PG and Gao L. “An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem”. *Computer & Industrial Engineering* 2009; 56(4): 1309-1318.

[7] Amiri M, Zandieh M, Yazdani M, and Bagheri A. “A variable neighborhood search algorithm for the flexible job-shop scheduling problem”. *International Journal of Production Research 2010*; 48(19): 5671-5689.

[8] Wang XJ, Gao L, Zhang CY, Shao XY. “A multi-objective genetic algorithm based on immune and entropy principle for flexible job-shop scheduling problem”. *The International Journal of Advanced Manufacturing Technology* 2010; 51(5-8): 757-767.

[9] Rahmati SHA, Zandieh M, Yazdani M. “Developing two multi-objective evolutionary algorithms for the multi-objective flexible job shop scheduling problem”. *The International Journal of Advanced Manufacturing Technology* 2012; 59: 683-697.

[10] Bagheri A, Zandieh M. “Bi-criteria flexible job shop scheduling with sequence-dependent setup times-variable neighborhood search approach”. *Journal of Manufacturing Systems* 2011; 30: 8-15.

[11] Kennedy J, Eberhart R. “Particle swarm optimization”. In: *Proceedings of the 1995 IEEE international conference on neural network*, IV, 1995; pp. 1942–1948.

[12] Mladenovic N. “A variable neighborhood algorithm a new metaheuristic for combinatorial optimization”. Abstracts of papers presented at Optimization Days: Montreal, 1995; pp. 112-112.

[13] Chiang TC, Lin HJ. “A simple and effective evolutionary algorithm for multiobjective flexible job shop scheduling”. *International Journal of Production Economics* 2012; 141(1): 87-98.

[14] Zhang GH, Gao L, Li PG, Zhang CY. “Improved genetic algorithm for the flexible job-shop scheduling problem”. *Journal of Mechanical Engineering* 2009; 45(7): 145-151.

[15] Kacem K, Hammadi S, and Borne P. "Pareto-optimality approach for flexible job-shop scheduling problems: Hybridization of evolutionary algorithms and fuzzy logic". *Mathematics and Computers in Simulation* 2002; 60: 245-276.

[16] Xia WJ and Wu ZM. "An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problem". *Computer & Industrial Engineering* 2005; 48(2): 409-425.

[17] Gao J, Sun LY and Gen M. "A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems". *Computer & Operations Research* 2008; 35(9): 2892-2907.

Received: September 16, 2014 Revised: December 23, 2014 Accepted: December 31, 2014