2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

A Memetic Algorithm based on Immune Multi-objective Optimization for Flexible Job-shop Scheduling Problems

Jingjing Ma, Yu Lei, Zhao Wang, Licheng Jiao, Ruochen Liu

***Abstract*—The flexible job-shop scheduling problem (FJSP) is an extension of the classical job scheduling which is concerned with the determination of a sequence of jobs, consisting of many operations, on different machines, satisfying parallel goals. This paper addresses the FJSP with two objectives: Minimize makespan, Minimize total operation cost. We introduce a memetic algorithm based on the Nondominated Neighbor Immune Algorithm (NNIA), to tackle this problem. The proposed algorithm adds, to NNIA, local search procedures including a rational combination of undirected simulated annealing (UDSA) operator, directed cost simulated annealing (DCSA) operator and directed makespan simulated annealing (DMSA) operator. We have validated its efficiency by evaluating the algorithm on multiple instances of the FJSPs. Experimental results show that the proposed algorithm is an efficient and effective algorithm for the FJSPs, and the combination of UDSA operator, DCSA operator and DMSA operator with NNIA is rational.**

Keywords-Flexible job-shop scheduling; memetic algorithm; multi-objective optimization; immune algorithm; simulated annealing

I. Introduction

In the job-shop scheduling problem (JSP), a group of m machines process n jobs. Each job i consists of a sequence of m operations ($o_{i1}, o_{i2}, \ldots, o_{im}$), where o_{ik} (the kth operation of job i) must be processed without interruption on a predefined machine m_{ik} for p_{ik} time units. The operations $o_{i1}, o_{i2}, \ldots, o_{im}$ must be processed one after another in the given order and each machine can process at most one operation at a time. In this paper we study a generalization of JSP called the flexible job-shop scheduling problem (FJSP), which provides a closer approximation to a wide range of problems encountered in real manufacturing systems. The FJSP extends JSP by allowing an operation o_{ik} to be executed by one machine out of a set A_{ik} of given machines. The FJSP problem is to choose for each operation o_{ik} a machine $M(o_{ik}) \in A_{ik}$.

The FJSP is NP-hard since it is an extension of the job-shop scheduling [1]. Bruker and Schlie [2] were the first to address the FJSP. They proposed a polynomial algorithm for solving the FJSP with two jobs, in which the machines capable of performing one operation have the same processing time. Jurisch [3] considered the routing and scheduling decisions simultaneously. Hurink, Jurisch and Thole [4] and Chambers [5] developed tabu search algorithms to solve the problem. Mastrolilli and Gambardella [6] proposed two neighborhood functions for this problem. Kacem and Borne [7] proposed a localization approach to solve the resource assignment problem, and an evolutionary approach controlled by the assignment model for the FJSP problem. Zhang and Gen [8] proposed a multistage operation-based GA to deal with the problem from a point view of dynamic programming. Xia and Wu [9] treated this problem with a hybrid of particle swarm optimization (PSO) and simulated annealing (SA) as a local search algorithm. Gao, Sun and Gen [10] proposed a hybrid genetic and variable neighborhood descent algorithm for flexible job-shop scheduling problems. Pezzella Morganti and Ciaschetti [11] proposed a genetic algorithm for the Flexible Job-shop Scheduling Problem. Frutos, Olivera and Tohme [12] introduced a memetic algorithm based on a NSGA-II scheme for the total flexible job-shop scheduling problem. Moslehi and Mahnam [13] proposed a Pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search. Rahmati and Zandieh [14] proposed a new biogeography-based optimization (BBO) algorithm for the flexible job-shop scheduling problem.

In this study, the considered objective is to minimize the makespan ($C_{\max}$) and the total operating cost (*TOC*). This optimization problem is known as multi-objective optimization problems (MOPs). In the past two decades, many efficient multi-objective optimization evolutionary algorithms (MOEAs) have been presented. The typical representatives of these algorithms were the Pareto Archived Evolution Strategy (PAES) [15], the Pareto Envelope based Selection Algorithm (PESA) [16], the Multi-Objective Messy Genetic Algorithm (MOMGA) [17], the Micro Genetic Algorithm (MicroGA) [18], the Strength Pareto Evolutionary Algorithm (SPEA) [19] and its improved version (SPEA2) [20], the Nondominated Sorting

This work was supported by the National Natural Science Foundation of China (Grant no. 61202176), the Specialized Research Fund for the Doctoral Program of Higher Education (Grant no. 20130203110011) and the Fundamental Research Fund for the Central Universities (Grant no. K5051202053).

Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, Xidian University, Xi'an, 710071, China

Genetic Algorithm (NSGA) [21] and its improved version (NSGA-II) [22], the Multi-objective Particle Swarm Optimization (MOPSO) [23], the Multi-objective Evolutionary Algorithm Based on Decomposition (MOEA/D) [24], Regularity Model Based Multi-Objective Estimation of Distribution Algorithm (RM-MEDA) [25], the Archived Multi-objective Simulated Annealing Algorithm (AMOSA) [26], and the Fast Hypervolume-Based Many-Objective Optimization (HypE) [27]. We also proposed a multi-objective optimization algorithm, named Nodominated Neighbor Immune Algorithm (NNIA) [28]. NNIA adopts a nondominated neighbor-based selection technique, an immune inspired operator, two heuristic search operators, and elitism. It turns out that NNIA is an effective method for solving MOPs by a mass of experiments [28–31].

In this paper, we seek solutions to the FJSP by means of a memetic algorithm that combines NNIA with three guided local search operators. We use two vectors to express the machine assignment and operation sequence information for FJSP solution candidates.

The remainder of this paper is organized as follows: Section II briefly describes related background. The memetic algorithm based on NNIA for the flexible job-shop scheduling problem and its contrastive algorithms are presented in Section III. In Section IV, experiments will be carried out to evaluate the efficiency of the memetic algorithm with several instances based on practical data. In Section V, concluding remarks are presented.

II. Related Background

A. Multi-objective optimaization and NNIA

Multi-objective optimization [32–33] seeks to optimize a vector of functions, where Ω is the decision space, $F:\Omega \to R^m$ is the map of decision space to m real valued objectives space.

$$F(\boldsymbol{x}) = (f_1(\boldsymbol{x}),...,f_m(\boldsymbol{x}))$$
$$\text{Subject to } \boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in \Omega \tag{1}$$

Take a minimization problem into consideration. It is said that a vector $\boldsymbol{x}_A \in \Omega$ dominates another vector $\boldsymbol{x}_B \in \Omega$ (written as $\boldsymbol{x}_A \prec \boldsymbol{x}_B$) if and only if

$$\forall i = 1,2,...,m$$
$$f_i(\mathbf{x}_A) \le f_i(\mathbf{x}_B) \wedge \exists j = 1,2,...,m \tag{2}$$
$$f_i(\mathbf{x}_A) < f_i(\mathbf{x}_B)$$

It is said that a vector of decision valuables $\boldsymbol{x}^* \in \Omega$ is a Pareto-optimal solution or nondonimated solution if there does not exist another $\boldsymbol{x} \in \Omega$ such that $\boldsymbol{x} \prec \boldsymbol{x}^*$.

Therefore, the Pareto-optimal set is defined as (3).

$$P^* = \{\boldsymbol{x}^* \in \Omega \mid \neg\exists \boldsymbol{x} \in \Omega, \boldsymbol{x} \prec \boldsymbol{x}^*\} \tag{3}$$

Then the Pareto-optimal set is the set of all Pareto-optimal solutions. The corresponding image of them under the objective function space is called the Pareto-optimal front, which can be described as follows:

$$PF^* = \{F(\boldsymbol{x}^*) = (f_1(\boldsymbol{x}^*), f_2(\boldsymbol{x}^*),...,f_m(\boldsymbol{x}^*)) \mid \boldsymbol{x}^* \in P^*\} \tag{4}$$

The purpose of an MOEA is to find a set of Pareto-optimal solutions approximate the true Pareto-optimal front.

As is presented in Section 1, many MOEAs have emerged since the early 2000s. The NNIA, which is based on Artificial Immune System, was presented by us in [28]. NNIA is proposed for multi-objective optimization by using an immune inspired operator, two heuristic search operators, and elitism. In this algorithm, we store nondominated individuals found so far in an external population, called the dominant population. Only partial less-crowded nondominated individuals, called active antibodies, are selected to do proportional cloning, recombination, and static hypermutation. Furthermore, the population storing clones is called the clone population. The dominant population, active population, and clone population at time t are presented by time-dependent variable matrices D_t, A_t and C_t, respectively. The details of NNIA are described as follows.

Input: $G_{\max}$ (maximum number of generations)
n_D (maximum size of dominant population)
n_A (maximum size of active population)
n_C (size of clone population)

Output: $D_{G_{\max}+1}$ (final approximate Pareto-optimal set)

Step1: Initialization: Generate an initial antibody population B_0 with size n_D . Create the initial $D_0 = \phi$, $A_0 = \phi$, and $C_0 = \phi$. Set $t = 0$.

Step2: Update Dominant Population: Identify dominant antibodies in B_t. Copy all the dominant antibodies to form the temporary dominant population (denoted by DT_{t+1}). If the size of DT_{t+1} is not greater than n_D , let $D_{t+1} = DT_{t+1}$. Otherwise, calculate the crowding distance values of all individuals in DT_{t+1} , sort them in descending order of crowding distance, and choose the first n_D individuals to form D_{t+1}.

Step3: Termination: If $t > G_{\max}$ is satisfied, export D_{t+1} as the output of the algorithm, Stop; Otherwise, $t = t+1$.

Step4: Nondominated Neighbor-Based Selection: If the size of D_t is not greater than n_A , let $A_t = D_t$. Otherwise, calculate the crowding distance values of all individuals in D_t, sort them in descending order of crowding distance, and choose the first n_A individuals to form A_t.

Step5: Proportional Cloning: Get the clone population C_t by applying proportional cloning to A_t.

Step6: Recombination and Hypermutation: Perform recombination and hypermutation on C_t and set C_t' to the resulting population.

Step7: Get the antibody population B_t by combining the C_t' and D_t; go to Step 2.

B. *Bi-objective flexible job-shop scheduling problem*

The FJSP is to organize the execution of *n* jobs, noted $J = \{J_1, J_2, ..., J_n\}$, where each job $J_i (1 \le i \le n)$ consists of a sequence of n_i operations, $O_{i,1}, O_{i,2}, ..., O_{i,n_i}$ on a given machine from a machine set named $M = \{M_1, M_2, ..., M_m\}$. In general, the details of FJSP definition are described as follows [34]. All machines are available at time 0. All jobs are released at time 0. The order of operations for each job is predefined and cannot be modified. The processing time of an operation $O_{i,j}$ on machine $k (O_{ij}^k)$ is predefined and denoted by p_{ij}^k and starting time for operation $O_{i,j}$ on machine k denoted by t_{ij}^k. Cost value for i th operation from job j on machine k denoted by v_{ik}^i. Each operation cannot be interrupted during its performance. Each machine can perform at most one operation at any time. The precedence constraints of the operations in job can be defined for any pair of operations.

In this paper the bi-objective FJSP is considered. One is to minimize the maximum completion time, denoted makespan $C_{\max}$ and the other to minimize the total operation cost *TOC*. Formulation of the problem studied in this research was established by Frutos, Olivera and Tohme. [12].

$$\min\ C_{\max} = \max(t_{ij}^k + p_{ij}^k) \tag{5}$$

$$\min\ TOC = \sum_{i \in J} \sum_{i \in S_j} \sum_{k \in M} x_{jk}^i \times v_{jk}^i \tag{6}$$

Subject to:

$$t_{ij}^k \ge 0, \qquad \forall j \in J, \forall k \in M$$

$$t_{ij}^k - t_{ij}^h \ge p_{sj}^h, \qquad \text{if } O_{sj}^h \text{ precedes } O_{ij}^k, \forall j \in J, \forall \{h,k\} \in M$$

$$t_{ij}^k - t_{sp}^k \ge p_{sp}^k, \qquad \text{if } O_{sp}^h \text{ precedes } O_{ij}^k, \forall \{j,p\} \in J, \forall k \in M$$

$$\sum_{k=1}^{m} x_{jk}^i = 1, \qquad \forall k \in M$$

Where:

$$x_{jk}^i = 1, \qquad \text{if } O_{jk}^i \in E_k, x_{jk}^i = 0, \text{ otherwise}$$

$$t_{jk}^i = \max(t_{jh}^{i-1} + \tau_{jh}^{(i-1)}, t_{pk}^s + \tau_{pk}^s, 0)$$

$$\forall \{p \to j\} \in E_k, \quad \forall \{h,k\} \in M, \quad \forall \{s,j\} \in S_j$$

III. The Proposed Memetic Algorithm for FJSP

A. *Algorithm framework*

The pseudo-code of the proposed memetic algorithm is given in Fig.1. In the local search procedure of the proposed memetic algorithm framework, we used the DCSA operator or the DMSA operator at equivalent probability after the UDSA operator firstly applied, and then the improved algorithm is called the Nondominated Neighbor Immune Algorithm with three diverse Simulated Annealing operators (NNIAT). Five contrastive algorithms based on the above framework are proposed as follows. The memetic algorithm framework without local search procedure is called the Original Nondominated Neighbor Immune Algorithm (ONNIA). The memetic algorithm framework with the UDSA operator in local search procedure is called the Nondominated Neighbor Immune Algorithm with UDSA operator (NNIAU). The memetic algorithm framework with the DMSA operator in local search procedure is called the Nondominated Neighbor Immune Algorithm with DMSA operator (NNIAM). In the local search procedure of memetic algorithm framework, we used the DMSA operator after the DCSA operator firstly applied, and then the algorithm is called Nondominated Neighbor Immune Algorithm with DCSA operator and DMSA operator (NNIACM). In the local search procedure of memetic algorithm framework, we used the DCSA operator after the DMSA operator firstly applied, and then the algorithm is called Nondominated Neighbor Immune Algorithm with DMSA operator and DCSA operator (NNIAMC).

Memetic Algorithm based on NNIA for FJSP:

```
1.  begin algorithm
2.    Continue = Yes;
3.    Generation = 0;
4.    Create Initial Population (P0);
5.    Decode Population (P0);
6.    Evaluate Population (P0);
7.    [ClonePop0, Mpop0]=Select(P0);
8.    while Generation ≤number of generations;
9.        P1 = Clone(ClonePop0);
10.        P2 = Crossover (P1);
11.        P3 = Mutation (P2);
12.        P4 = Local Search (P3);
13.        Decode Population (P4);
14.        Evaluate Population (P4);
15.        [ClonePop, Mpop] = Select (Mpop0,P4);
16.        Generation ++;
17.        until Continue = No;
18.  end while
19. end algorithm
```

Figure 1. Framework of the proposed memetic algorithm for FJSP

B. *Solution Represention*

The FJSP is a combination of machine assignment and operation scheduling decisions, so a solution can be expressed by the assignment of operations on machines and the processing sequence of operations on the machines. The chromosome is therefore composed of two parts: machine assignment vector and operation sequence vector. In the first part, for each operation a number between 1 and the number of available machines for the related operation is generated. The second part is a sequence vector which is generated discretely that each job i appears in the operation sequence vector exactly n_i times to represent its n_i ordered operations. The first part represents machines assigned and the second part represents the operations sequence to the operations in polar place similar to Gen Tsujimura and Kubota [35]. The main advantages of Gen et al.'s representation are that each possible chromosome always represents a feasible operation sequence, and that the coding space is smaller than that of permutation representation. In principle, a chromosome of the FJSP can be decoded into an infinite number of schedules because superfluous idle time can be inserted between operations. In this paper, we use priority-based decoding to translate chromosomes into active schedules. The priority-based decoding [10] allocates each operation on its assigned machine one by one in order represented by the operation sequence vector. The method allows an operation to search the earliest available time interval on the machine. The operation sequence in a chromosome is reordered according to the operations' starting time in the decoded schedule before the chromosome involves crossover operations and mutation operations.

C. Crossover and mutation operators

During the past decades, several crossover operators have been proposed for permutation representation, such as partial-mapped crossover, order crossover, cycle crossover, and so on [36]. In this paper, we apply the order crossover for the operation sequence vectors. The order crossover works as follows:

Step1: Select a subsection of operation sequence from one parent at random.

Step2: Produce a proto-child by copying the substring of operation sequence into the corresponding positions.

Step3: Delete the operations that are already in the substring from the second parent. The resulted sequence of operations contains operations that the proto-child needs.

Step4: Place the operations into the unfixed positions of the proto-child from left to right according to the order of the sequence in the second parent.

We use two crossover operators at equivalent probability for the machine assignment vectors: extended order crossover and uniform crossover [10]. The extended order crossover is related to crossover for operation sequence. It copies the machine assigned for an operation from the same parent where its operation sequence comes. Uniform crossover is accomplished by taking an allele from either parental machine assignment vector to form the corresponding allele of the child.

Two kinds of mutation operations are implemented here: allele-based mutation and immigration mutation [10]. For machine assignment vectors, allele-based mutation randomly decides whether an allele should be selected for mutation with a certain probability. Then, another available machine will be assigned for the operation indicated by the selected allele. For operation sequence vectors, allele-based mutation randomly decides whether to mutate an allele r. If allele r is to be mutated, then another allele is randomly selected to exchange with it. Immigration mutation randomly generates a number of new members of the population from the same distribution as the initial population.

D. Local search operators through simulated annealing

The local search complement to the genetic stage is provided by simulated annealing (SA). It is intended to allow a progression towards a better solution when an algorithm gets stuck in a local minimum [37]. We introduce three local search operators including undirected simulated annealing (UDSA) operator, directed cost simulated annealing (DCSA) operator and directed makespan simulated annealing (DMSA) operator. We use a rule $T_m = \alpha T_{m-1}$, where $\alpha(\alpha \in (0,1))$ is the cooling coefficient [37]. The number of iteration for each temperature T is $M(T) = 1/T$.

```
Algorithm: UDSA
begin algorithm
  Continue = Yes;
  while Stop Condition = No(T>T_f)
   Calculate M=1/T;
   for i=1 to M
     Alter machine assignment vector and operation sequence vector of
     current solution imultaneously like allele-based mutation, and then
     decode new solution and Evaluate new solution;
     if new solution p current solution
         current solution replaced by new solution;
     else
        if rand(0,1) < 0.01
           current solution replaced by new solution;
        end
     end
   end
  end
end algorithm
```

Figure 2. Undirected simulated annealing scheme

```
Algorithm: DCSA
begin algorithm
  Continue = Yes;
  while Stop Condition = No(T>Tf)
    Calculate M=1/T;
    for i=1 to M
      Only alter machine assignment vector of current solution like
      allele-based mutation;
      Decode new solution and Evaluate new solution;
      if new solution p current solution
          current solution replaced by new solution;
      else
         if rand(0,1) < 0.01
            current solution replaced by new solution;
         end
      end
    end
   end
end algorithm
```

Figure 3. Directed cost simulated annealing scheme

```
Algorithm: DMSA
begin algorithm
  Continue = Yes;
  while Stop Condition = No(T>Tf);
    Calculate M=1/T;
    for i=1 to M;
      Only alter operation sequence vector of current solution like
      allele-based mutation;
      Decode new solution and Evaluate new solution;
      If improve makespan
         current solution replaced by new solution;
      else
         if rand(0,1) < 0.01
            current solution replaced by new solution;
         end
      end
    end
   end
end algorithm
```

Figure 4. Directed makespan simulated annealing scheme

The UDSA operator is an undirected local search operator. The DCSA operator and DMSA operator are the directed ones. The UDSA is to minimize the makespan and the total operating cost by altering machine assignment vector and operation sequence vector simultaneously. The DCSA directionally minimizes the total operating cost by only altering machine assignment vector. The DMSA directionally minimizes the makespan by only altering operation sequence vector. The basic schemes for UDSA, DCSA and DMSA are presented in Fig.2, Fig.3 and Fig.4 respectively.

IV. EXPERIMENTAL STUDIES

In this section, we compare NNIAT with ONNIA, NNIAU, NNIAM, NNIACM and NNIAMC in solving four FJSPs to validate the effectiveness of the combination of UDSA operator, DCSA operator and DMSA operator in NNIAT.

TABLE I. ZJ01 /PROBLEM 3×4 WITH 8 OPERATIONS

J_j	O^i_{jk}	M1	M2	M3	M4
		τ^i_{j1}/v^i_{j1}	τ^i_{j2}/v^i_{j2}	τ^i_{j3}/v^i_{j3}	$\tau^i_{j4}/^i_{j4}$
J_1	O^1_{1k}	-/-	3/8	4/6	1/9
	O^2_{1k}	3/4	8/2	-/-	1/12
	O^3_{1k}	3/8	5/4	4/6	7/3
J_2	O^1_{2k}	4/7	-/-	1/14	4/6
	O^2_{2k}	2/10	3/8	9/3	-/-
	O^3_{2k}	9/3	1/15	2/10	-/-
J_3	O^1_{3k}	8/6	-/-	3/12	5/10
	O^2_{3k}	-/-	5/8	8/6	1/18

A. Experimental setup

In our experiment, we select t three partial flexible job-shop scheduling instances and one total flexible job-shop scheduling instance as the test problems. The instances contain ZJ01 (problem 3×4 with 8 operations), ZJ02 (problem 5×5 with 15 operations) and ZJ03 (problem 10×10 with 30 operations) presented TABLE I-III. In the example of TABLE I-III, symbol "-" indicates that the assignment is impossible. The only total flexible job-shop scheduling instance is MF02 (problem 4×5 with 12 operations) which was defined by Frutos et al. (2010).In our study, the hypervolume metric and the nondonimated solutions metric will be applied. The parameters was implemented are shown in TABLE III. .

TABLE II. ZJ02 /PROBLEM 5×5 WITH 15 OPERATIONS

J_j	O^i_{jk}	M1	M2	M3	M4	M5
		τ^i_{j1}/v^i_{j1}	τ^i_{j2}/v^i_{j2}	τ^i_{j3}/v^i_{j3}	$\tau^i_{j4}/^i_{j4}$	τ^i_{j5}/v^i_{j5}
J_1	O^1_{1k}	2/7	5/3	-/-	1/15	2/7
	O^2_{1k}	-/-	5/4	4/5	7/3	5/4
	O^3_{1k}	-/-	4/3	5/3	3/4	5/3
J_2	O^1_{2k}	2/12	5/4	4/6	7/3	-/-
	O^2_{2k}	5/5	6/4	-/-	9/3	8 /3
	O^3_{2k}	4/40	5/32	4/40	-/-	5/32
J_3	O^1_{3k}	9/3	-/-	6/4	7/3	9/3
	O^2_{3k}	6/3	8/3	18/2	-/-	4/4
	O^3_{3k}	2/7	1/18	4/3	2/7	-/-
	O^4_{3k}	4/3	5/3	2/7	1/15	-/-
J_4	O^1_{4k}	1/36	5/7	2/18	4/9	12/3
	O^2_{4k}	5/3	1/15	-/-	2/7	2/7
J_5	O^1_{5k}	9/3	-/-	8/3	6/4	9/3
	O^2_{5k}	5/3	-/-	9/3	4/9	12/3
	O^3_{5k}	-/-	1/35	2/18	12/2	5/3

TABLE III. ZJ03 /PROBLEM10×10 WITH 30 OPERATIONS

J_j	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
	τ^i_{j1}/v^i_{j1}	τ^i_{j2}/v^i_{j2}	τ^i_{j3}/v^i_{j3}	$\tau^i_{j4}/^i_{j4}$	τ^i_{j5}/v^i_{j5}	τ^i_{j6}/v^i_{j6}	τ^i_{j7}/v^i_{j7}	τ^i_{j8}/v^i_{j8}	$\tau^i_{j9}/^i_{j9}$	τ^i_{j10}/v^i_{j10}
J_1	-/-	1/11	4/2	6/1	9/1	3/3	5/2	2/5	8/1	-/-
	1/14	-/-	3/4	4/3	8/1	10/1	4/3	11/1	-/-	3/4
	10/1	4/4	5/3	9/2	8/2	4/4	15/1	2/8	-/-	-/-
J_2	4/3	8/1	7/1	1/13	9/1	6/2	2/1	-/-	-/-	1/13
	-/-	-/-	11/1	6/3	7/2	5/3	3/6	1/9	2/9	3/18
	8/1	5/2	8/2	1/9	4/2	3/5	8/1	-/-	-/-	1/11
J_3	7/1	1/11	8/1	4/2	9/1	1/11	-/-	-/-	3/3	4/2
	5/2	-/-	-/-	9/1	1/13	7/1	2/13	6/2	2/7	8/1
	7/2	3/5	1/15	6/2	-/-	8/1	-/-	2/7	5/3	2/7
J_4	6/1	2/3	1/4	1/7	2/3	-/-	-/-	5/1	4/1	2/3
	9/1	9/7	6/4	-/-	1/14	2/23	21/1	5/9	-/-	5/8
	11/1	1/2	6/2	2/7	5/2	-/-	-/-	2/7	1/14	4/3
J_5	6/1	9/1	-/-	3/3	-/-	8/1	7/1	1/11	2/5	4/2
	5/7	-/-	-/-	3/12	5/9	2/18	28/1	7/5	4/9	5/7
	6/1	4/2	3/3	7/1	1/9	-/-	9/5	-/-	2/5	4/3
J_6	4/2	1/11	6/1	3/12	9/1	8/1	4/2	2/5	5/9	10/1
	6/1	5/1	1/9	3/3	6/1	5/1	7/1	4/2	6/1	10/1
	8/1	9/1	10/1	1/4	2/2	5/7	3/3	10/1	1/8	2/7
J_7	1/11	-/-	2/5	4/2	5/2	3/3	9/1	8/1	2/4	4/2
	2/3	-/-	-/-	-/-	2/3	3/2	4/1	5/1	9/1	1/12
	6/1	5/1	4/2	2/4	3/3	2/4	7/1	5/1	-/-	-/-
J_8	2/5	3/3	6/1	-/-	5/5	2/4	1/10	-/-	8/1	7/1
	4/1	5/1	2/3	3/2	4/1	7/1	5/1	-/-	-/-	-/-
	1/16	2/23	13/1	5/9	2/23	3/15	6/7	-/-	-/-	2/21
J_9	6/9	2/21	3/19	21/2	1/11	5/10	3/9	2/5	-/-	-/-
	2/11	-/-	-/-	12/1	15/1	2/12	1/14	1/16	16/1	1/18
	9/8	8/9	7/10	4/18	5/14	8/9	7/10	4/18	8/9	2/24
J_{10}	4/5	3/6	-/-	1/20	-/-	2/1	20/2	6/3	8/1	6/2
	3/5	1/17	8/2	1/17	9/1	1/18	4/4	-/-	-/-	15/1
	9/2	2/11	4/5	11/3	2/11	-/-	-/-	10/2	4/5	1/9

TABLE IV. PARAMETERS OF OUR EXPERIMENTS

Parameters and characteristics	Value/type
Maximum Size of Dominant Population	100
Maximum Size of Active Population	30
Size of Clone Population	100
Number of Generations	100
Type of Crossover :Probability	order crossover: 0.5 uniform crossover: 0.5
Type of mutation: Probability	allele-based mutation:0.4 immigration mutation: 0.4
Type of Local Search	UDSA DCSA DMSA
Initial Temperature (Ti)	800
Final Temperature (Tf)	0.1
Cooling Factor (a)	0.75

B. *Experimental results*

1) Validating the effectiveness of the directed local search operators

We applied the ONNIA, NNIAM, NNIACM and NNIAMC to problems ZJ01, ZJ02, ZJ03 and MF02. ZJ01, ZJ02 and ZJ03 are partial flexible job-shop scheduling instances. MF02 is total flexible job-shop scheduling instance. Thirty independent runs on each test problem are performed in the following experiments. We show the box plots of hypervolume metric and nondonimated solutions metric of ONNIA, NNIAM, NNIACM and NNIAMC in Fig.5. Note that the hypervolume metric values are normalized by the difference of maximum and minimum hypervolume value for each problem. The hypervolume metric is the main index and the nondonimated solutions metric is the minor reference to estimate the four algorithms. In terms of hypervolume, it could be seen that NNIAM has obtained the best results for solving the ZJ01. For ZJ02, ZJ03 and MF02, the NNIAM, NNIACM and NNIAMC have obtained better measures than ONNIA. Meanwhile, the NNIAM, NNIACM and NNIAMC have obtained equal results for solving the ZJ02, ZJ03 and MF02. In terms of nondonimated solutions, the ONNIA, NNIAM, NNIACM and NNIAMC have obtained the proximate results for solving the ZJ01, ZJ02, ZJ03 and MF02.

Considering the experimental results above, we believe that the following conclusions could hold. For ZJ01, ZJ02, ZJ03 and MF02, the NNIACM and NNIAMC have little superiority compared with NNIAM. The DCSA operator and DMSA operator both used in the local search procedure of memetic algorithm framework could not obtain better measures than the DMSA operator used alone. Hence, in the same generation, these two different employed directed local search operators might disturb each other in the local search procedure.

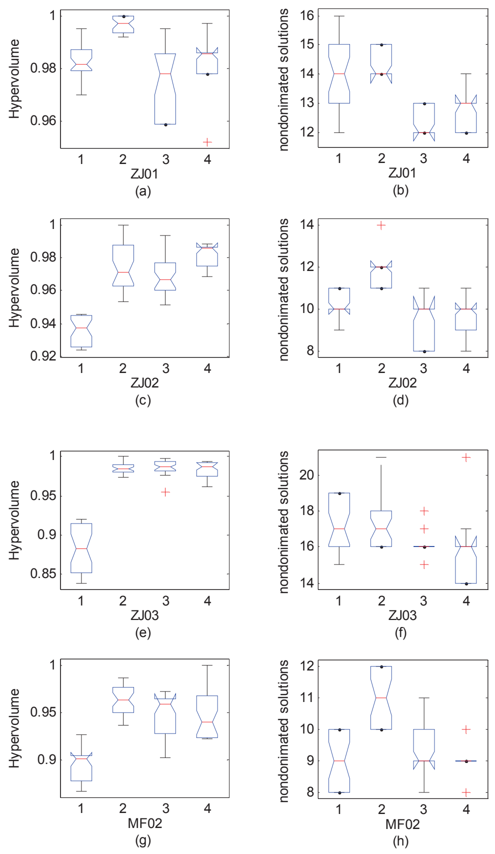

Figure 5. Statistical values of hypervolume and nondonimated solutions for ZJ01, ZJ02, ZJ03 and MF02 by ONNIA (1), NNIAM (2), NNIACM (3) and NNIAMC (4).

2) Comparisons among ONNIA, NNIAU and NNIAT

In the local search procedure of memetic algorithm framework, we used the DCSA operator or the DMSA operator at equivalent probability after the UDSA operator firstly applied, and then the algorithm is called NNIAT. We applied the ONNIA, NNIAU and NNIAT to problems ZJ01, ZJ02, ZJ03 and MF02. Thirty independent runs on each test problem are performed in the following experiments. We show the box plots of hypervolume metric and nondonimated solutions metric of ONNIA, NNIAU and NNIAT in Fig.6. Note that the hypervolume metric values are normalized by the difference of maximum and minimum hypervolume value for each problem. The hypervolume metric is the main index and the nondonimated solutions metric is the minor reference to estimate the three algorithms. In terms of hypervolume, it could be seen that NNIAT has obtained the best results and the ONNIA has obtained the worst results in the three algorithms for solving the ZJ01, ZJ02, ZJ03 and MF02. In terms of nondonimated solutions, the ONNIA, NNIAU and NNIAT have obtained the proximate results for solving the ZJ01, ZJ02, ZJ03 and MF02.

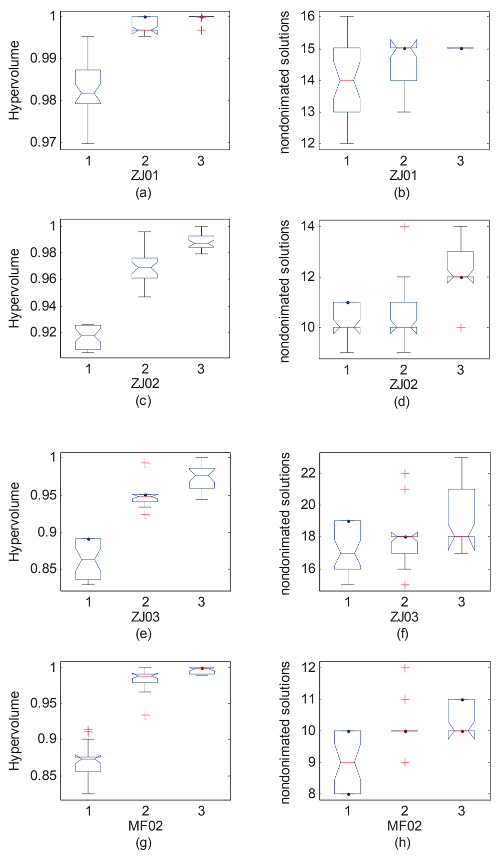

Figure 6. Statistical values of hypervolume and nondonimated solutions for ZJ01, ZJ02,ZJ03 and MF02 by ONNIA (1), NNIAU (2) and NNIAT(3).

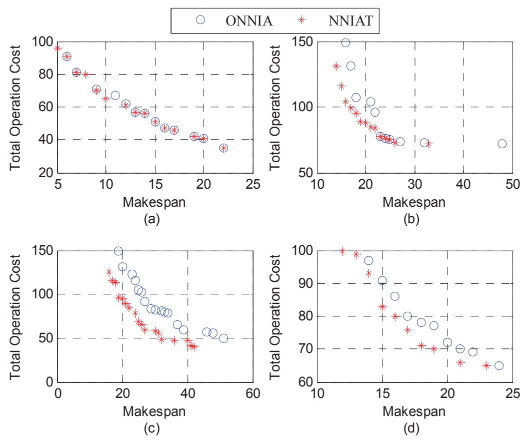

Figure 7. Plots of typical final approximate fronts for ZJ01 problem (a), ZJ02 problem (b), ZJ03 problem (c) and MF02 problem (d) obtained from ONNIA and NNIAT.

Overall, considering the experimental results above, we could conclude that the NNIAT with three different local search operators could be an efficient and effective algorithm for the FJSPs, and the combination of UDSA operator, DCSA operator and DMSA operator in NNIAT is rational.

TABLE V. STATISTICAL VALUES OF HYPERVOLUME FOR ZJ01S, ZJ02S,ZJ03S AND MF02S BY ONNIA, NNIAU AND NNIAT

Test	Normalized values of hypervolume		
	ONNIA	NNIAU	NNIAT
ZJ011	0.98(0.004)	0.99(0.003)	**0.99(0.001)**
ZJ012	0.98(0.006)	0.99(0.002)	**1(0)**
ZJ013	0.99(0.005)	0.99(0.001)	**1(0)**
ZJ014	0.99(0.005)	0.99(0.001)	**1(0)**
ZJ015	0.98(0.009)	0.99(0.006)	**0.99(0.001)**
ZJ016	0.99(0.003)	0.99(0.006)	**1(0)**
ZJ017	0.99(0.004)	0.99(0.002)	**1(0)**
ZJ018	0.99(0.012)	0.99(0.006)	**1(0)**
ZJ019	0.98(0.014)	0.99(0.002)	**1(0)**
ZJ0110	0.99(0.002)	0.99(0.001)	**1(0)**
ZJ021	0.88(0.028)	0.94(0.028)	**0.98(0.007)**
ZJ022	0.93(0.028)	0.96(0.024)	**0.97(0.019)**
ZJ023	0.91(0.023)	0.96(0.017)	**0.97(0.013)**
ZJ024	0.90(0.028)	0.97(0.012)	**0.99(0.008)**
ZJ025	0.92(0.028)	0.96(0.024)	**0.97(0.014)**
ZJ026	0.92(0.031)	0.96(0.009)	**0.98(0.011)**
ZJ027	0.94(0.029)	0.97(0.012)	**0.99(0.007)**
ZJ028	0.84(0.062)	0.94(0.020)	**0.97(0.027)**
ZJ029	0.93(0.013)	0.97(0.005)	**0.98(0.014)**
ZJ0210	0.86(0.038)	0.94(0.051)	**0.98(0.003)**
ZJ031	0.89(0.013)	0.96(0.018)	**0.98(0.008)**
ZJ032	0.85(0.030)	0.94(0.007)	**0.97(0.013)**
ZJ033	0.88(0.026)	0.94(0.019)	**0.98(0.012)**
ZJ034	0.82(0.010)	0.90(0.016)	**0.94(0.028)**
ZJ035	0.83(0.040)	0.93(0.020)	**0.98(0.014)**
ZJ036	0.88(0.033)	0.94(0.008)	**0.97(0.020)**
ZJ037	0.84(0.024)	0.95(0.020)	**0.98(0.011)**
ZJ038	0.86(0.040)	0.96(0.005)	**0.98(0.006)**
ZJ039	0.88(0.025)	0.96(0.014)	**0.98(0.009)**
ZJ0310	0.84(0.023)	0.92(0.022)	**0.96(0.018)**
MF021	0.92(0.017)	0.97(0.014)	**0.99(0.006)**
MF022	0.92(0.008)	0.98(0.008)	**0.99(0.005)**
MF023	0.94(0.004)	0.98(0.005)	**0.99(0.003)**
MF024	0.92(0.008)	0.98(0.006)	**0.99(0.006)**
MF025	0.80(0.018)	**0.98(0.010)**	0.97(0.012)
MF026	0.91(0.007)	0.98(0.008)	**0.99(0.008)**
MF027	0.94(0.006)	0.98(0.014)	**0.99(0.001)**
MF028	0.97(0.005)	0.98(0.009)	**0.99(0.003)**
MF029	0.94(0.012)	0.97(0.014)	**0.98(0.008)**
MF0210	0.94(0.013)	0.98(0.013)	**0.99(0.002)**

V. CONCLUDING REMARKS

In this paper, we studied the flexible job-shop scheduling problem. The considered objective is to minimize the makespan and the total operating cost. The FJSP is the NP-hard problem that has been more thoroughly studied in the literature, because of its practical interest. The Nondominated Neighbor Immune Algorithm with three diverse simulated annealing operators (NNIAT) is presented for solving this problem. We researched the effectiveness of the two directed local search operators and then the experiments results could show that the NNIAT is an efficient and effective algorithm for the FJSPs. Our future study is recommended to investigate a combination of diverse directed local search operators for solving many-objective flexible job-shop scheduling problem. Furthermore, evaluating of our proposed algorithm in other combinatorial and practical problems is worthy to be researched.

References

[1] MR. Garey, DS. Johnson, R. Sethi, The complexity of flowshop and job-shop scheduling. Mathematics of Operations Research 1976;1:117–129.

[2] P. Bruker, R. Schlie, Job-shop scheduling with multi-purpose machines. Computing 1900;45(4):369–375.

[3] B. Jurisch. Scheduling jobs in shops with multi-purpose machines. Ph.D. dissertation, Fachbereich Mathematik/Informatik, Universitat Osnabruck; 1992.

[4] E. Hurink, B. Jurisch, M. Thole, Tabu search for the job-shop scheduling problem with multi-purpose machine. Operations Research Spektrum 1994;15:205–15.

[5] JB. Chamber, Classical and flexible job-shop scheduling by tabu search. Ph.D. dissertation, University of Texas at Austin, USA; 1996.

[6] M. Mastrolilli, LM. Gambardella, Effective neighborhood functions for the flexible job shop problem. Journal of Scheduling 2000;3(1):3–20.

[7] I. Kacem, S. Hammadi, P. Borne, Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems. IEEE Transaction Systems, Man, and Cybernetics-Part C 2002;32(1):1–13.

[8] HP. Zhang, M. Gen, Multistage-based genetic algorithm for flexible job-shop scheduling problem. International Journal of Complexity 2005;11:223–32.

[9] W. Xia, Z. Wu, An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problem. Computers & Industrial Engineering 2005;48:409–25.

[10] J. Gao, L. Sun, M. Gen,A hybird genetic and variable neighborhood descent algorithm for flexible job-shop scheduling problems. Computers & Operation Research 2008;35(9):2892–2907.

[11] F. Pezzella, G. Morganti, G. Ciaschetti, A genetic algorithm for the Flexible Job-shop Scheduling Problem. Computers & Operation Research 2008;35(10):3202–3212.

[12] M. Frutos, AC. Olivera, F. Tohmé, A memetic algorithm based on a NSGAII scheme for the flexible job-shop scheduling problem. Annals of Operation Researches 2010;181(1):745–765.

[13] G. Moslehi, M. Mahnam, A Pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search. International Journal of Production Economics 2011;129(1):14–22.

[14] SHA. Rahmati, M. Zandieh, A new biogeography-based optimization (BBO) algorithm for the flexible job-shop scheduling problem. The International Journal of Advanced Manufacturing Technology 2012;58:1115–1129.

[15] JD. Knowles, DW. Corne, Approximating the non-dominated front using the Pareto archived evolution strategy. Evolutionary Computation 2000;8(2):149–172.

[16] DW. Corne, JD. Knowles, MG. Oates, The Pareto-envelope based selection algorithm formulti-objective optimization. In: Parallel problem solving from nature, PPSN VI 2000; 869–878.

[17] DA. Van Veldhuizen, GB. Lamont, Multi-objective optimization with messy genetic algorithms. In: Proceedings of the 2000 ACM symposium on applied computing. ACM Press 2000; p. 470–476.

[18] CA. Coello Coello, GT. Pulido, Multi-objective optimization using a micro-genetic algorithm. In: Proceedings of genetic and evolutionary computation conference, GECCO 2001; 274–282.

[19] E. Zitzler, L. Thiele, Multi-objective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Transactions on Evolutionary Computation 1999;3(4):257–271.

[20] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: improving the strength Pareto evolutionary algorithm. In: Evolutionary methods for design, optimization and control with applications to industrial problems, Athens, Greece 2002; 95–100.

[21] N. Srinivas, K. Deb, Multi-objective optimization using nondominated sorting in genetic algorithms. Evolutionary Computation 1993;2(3):221–248.

[22] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 2002;6(2):182–197.

[23] CA. Coello Coello, GT. Pulido, MS. Lechuga, Handing multiple objectives with particle swarm optimization. IEEE Transactions on Evolutionary Computation 2004;8(3):256–279.

[24] QF. Zhang, H. Li, MOEA/D: a multi-objective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation 2007;11(6):712–731.

[25] QF. Zhang, AM. Zhou, Y. Jin, RM-MEDA: a regularity model based multi-objective estimation of distribution algorithm. IEEE Transactions on Evolutionary Computation 2008;12(1):41–63.

[26] S. Bandyopadhyay, S. Saha, U. Maulik, K. Deb, A simulated annealing based multi-objective optimization algorithm: AMOSA. IEEE Transactions on Evolutionary Computation 2008; 12(3):269–283.

[27] J. Bader, E, Zitzler, HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization. Evolutionary Computation 2011;19(1):45–76.

[28] MG. Gong, LC. Jiao, HF. Du, LF. Bo, Multi-objective immune algorithm with nondominated neighbor-based selection. Evolutionary Computation 2008;16(2):225–255.

[29] DD. Yang, LC. Jiao, MG. Gong, Adaptive Multi-objective Optimization Based on Nondominated Solutions. Computational Intelligence 2009;25(2):84–108.

[30] DD. Yang, LC. Jiao, MG. Gong, J. Feng. Adaptive Ranks and K-Nearest Neighbour List based Multiobjective Immune Algorithm. Computational Intelligence 2010;26(4):359–385.

[31] DD. Yang, LC. Jiao, MG. Gong, F. Liu, Artificial Immune Multi-objective SAR Image Segmentation with Fused Complementary Features. Information Sciences 2001;181(13):2797–2812.

[32] K. Deb, Multi-Objective Optimization Using Evolutionary Algorithms. John Wiley and Sons, Chichester, UK; 2001.

[33] CA. Coello Coello, DA.Van Veldhuizen, GB. Lamont, Evolutionary algorithms for solving multi-objective problems. Kluwer, New York; 2002.

[34] M. Saidi-mehrabad, P. Fattahi, Flexible job-shop scheduling with tabu search algorithms. International Journal of Advanced Manufacturing Technology 2007;32(5):563–570.

[35] M. Gen, Y. Tsujimura, E. Kubota, Solving job-shop scheduling problem using genetic algorithms. In: Proceedings of the 16th international conference on computer and industrial engineering, Ashikaga, Japan; 1994. 2: 1577-1582.

[36] M. Gen, R. Cheng, Genetic algorithms & engineering design. NewYork: Wiley; 1997.

[37] KA. Dowsland, Simulated annealing, modern heuristic techniques for combinatorial problems. Oxford: Blackwell Sci.; 1993.