

Research Article

Cultural-Based Genetic Tabu Algorithm for Multiobjective Job Shop Scheduling

Yuzhen Yang and Xingsheng Gu

Key Laboratory of Advanced Control and Optimization for Chemical Processes, Ministry of Education, East China University of Science and Technology, Shanghai 200237, China

Correspondence should be addressed to Xingsheng Gu; xsgu@ecust.edu.cn

Received 5 December 2013; Accepted 6 February 2014; Published 13 March 2014

Academic Editor: Huaicheng Yan

Copyright © 2014 Y. Yang and X. Gu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The job shop scheduling problem, which has been dealt with by various traditional optimization methods over the decades, has proved to be an NP-hard problem and difficult in solving, especially in the multiobjective field. In this paper, we have proposed a novel quadspace cultural genetic tabu algorithm (QSCGTA) to solve such problem. This algorithm provides a different structure from the original cultural algorithm in containing double brief spaces and population spaces. These spaces deal with different levels of populations globally and locally by applying genetic and tabu searches separately and exchange information regularly to make the process more effective towards promising areas, along with modified multiobjective domination and transform functions. Moreover, we have presented a bidirectional shifting for the decoding process of job shop scheduling. The computational results we presented significantly prove the effectiveness and efficiency of the cultural-based genetic tabu algorithm for the multiobjective job shop scheduling problem.

1. Introduction

The scheduling problem is one of the most important and hardest combinatorial optimization problems on account of its complexity and frequency in practical applications. The purpose of scheduling generally is to allocate a set of resources to tasks by the definition of Pinedo. Since the first appearance of the systematic method to scheduling problems was in the mid-1950s, thousands of articles on different scheduling problems have arisen in the literature, which can be categorized in accordance with shop environments, including single machine, parallel machines, flow shop, flexible flow shop, job shop, open shop, and others.

Job shop scheduling problem (JSSP) is one of the most difficult ones among all the scheduling problems. Literature [1] has summarized the approaches applied to JSSP. Gradually it has been handled using exact methods, branch and bound, and heuristic procedures based on priority rules and shifting bottleneck. However, it is a nonlinear control problem, just as other complex control systems [2]. Therefore, large-size problems are still considered to surpass the reach of exact methods. And then over the last decades, a growing

number of heuristics have been presented to solve these complex optimization problems: simulated annealing (SA), tabu search (TS), and evolutionary algorithms (EA), among others.

Among all these algorithms, the cultural algorithm has been paid attention to gradually and applied to solve scheduling problems. Similar to the development of evolutionary computation, Reynolds developed a model of the evolution of cultural systems and subsequently the cultural algorithms in 1994 [3]. The cultural algorithm proposed is a dual evolutionary system, which provides an interaction and cooperation between two different evolutions: brief and population spaces. And the acceptance and influence functions allow the dual inheritance to make interactions. After Reynolds proposed the cultural algorithm, he and his students applied the use of cultural algorithms to global optimization problems with better results [4]. They presented five knowledge sources [5–7] and proposed a full fuzzy cultural algorithm [8] and combined genetic algorithm. And incorporation of other evolutionary algorithms [9–11] with cultural algorithms has become a research focus. Cultural algorithm has been extensively and successfully applied to optimization problems

and has advantages in overcoming some weaknesses of conventional optimization methods, such as data mining, scheduling, and multiobjective optimization [12–14]. It is an intelligent technique that incorporates knowledge in the evolutionary process to make the process more effective and efficient.

It is common sense that practical scheduling problems are multiobjective by nature. And a number of solutions should be provided to decision makers to make a conclusion. Smith started the researches on multiobjective scheduling in 1956 by providing a Smith algorithm for some scheduling problems with weighted sum of completion time minimization, subject to all jobs being completed by their due dates [15]. Afterwards, plenty of scholars studied the multiobjective scheduling. However, the approaches they provided cannot be spread to a multiobjective context because they aim at a specific problem. Gradually research methods on multiobjective scheduling problem have converted from exact methods to the multiobjective decision-making theory [16, 17] and modern heuristic search methods such as GA [18, 19], particle swarm optimization (PSO) [20], simulated annealing (SA) [21, 22], and water drops algorithm [23]. Literature [24] proposed an interactive solution to $J|d_i|F_T(C_{\max}, \bar{C}, \bar{I}, T_{\max}, \bar{U})$, which is interactive with DM based on Tchebycheff-approximation in the first module and then using the greedy heuristic algorithm in the second module. But due to the disadvantages of heuristics, the genetic algorithm with its inherent global search advantage has been considered to be an efficient and effective means to solve JSSP with multiple criteria. Lei [25] put forward a simplified multiobjective genetic algorithm SMGA for stochastic job shop scheduling with makespan and total tardiness minimization. Literature [26] applied a hybrid genetic algorithm and tabu search for dynamic JSSP.

However, different from the research in the area of the single objective scheduling problem, researches that pay attention to multiple criteria have been scarce. The goal of multiobjective JSSP is to find as many different promising schedules as possible. In this work, we apply makespan and mean flow time as the objectives of our algorithm. After considering the advantage and disadvantage of the cultural algorithm, we modified the structure and strategies of the original one to coordinate with multiobjective JSSP.

In this paper, we present a novel quad-space cultural genetic tabu algorithm to solve the multiobjective scheduling problem, which consists of four spaces, including double brief spaces and double population spaces. The search process is divided into two parts, which have their own cultural structures while exchanging with each other at a predefined frequency, which is one of the novelties of our paper. The other novelty is to design two different brief spaces and their influence functions along with the multiobjective selection strategy. Last but not least, a bidirectional shifting is presented for decoding the presentation, which could help improve the solutions efficiently. The goals of this paper are twofold: first, to analyze the effects when changing the parameters of our algorithm and to verify the proposed cultural algorithm, second, use the algorithm to deal with the real multiobjective JSSP. To justify our approach, we compared our proposed

approach with other well established MOEAs- (NSGAI [27], SPEA2 [28]) and MPSO- [29] based approaches and found that the QSCGTA-based scheduling approach is able to outperform others in some fields.

The remainder of the paper is organized as follows. In Section 2, we explain the scheduling problem and multiobjective problem specifically as well as the concept and structure of both the original cultural algorithm and our novel cultural structure in Section 3. In Section 4, the four spaces cultured genetic tabu algorithm is developed to JSSP subsequently. Section 5 analyzes the performance results of QSCGTA when applied to solve common benchmarks in literature. At last, we come to our conclusion and some possible future directions.

2. Problems Formulation

JSSP is one of the most famous and hardest combinatorial optimization problems. During the past decades, a bunch of literature has been published, but no efficient algorithm has been presented yet for solving it to optimality in polynomial time.

Suppose we are given n jobs and m machines. Each machine can handle one job at most at a time. Each job consists of a sequence of operations and needs to be processed during an uninterrupted time period on a given machine. The purpose is to find a schedule, that is, the job sequence on the machines as to optimize one or more performance measurements, makespan, and mean flow time in our case. The traditional n jobs, m machines multiobjective JSSP can be described as follows:

$$\text{Minimize } \min_{X \in D} \{f_l(X)\}; \quad l = 1, \dots, k \quad (1)$$

$$f_1 = \max(T_{i,m}) \quad i = 1, \dots, n, \quad (2)$$

$$f_2 = \frac{1}{m} \sum_{i=1}^n T_{i,m}, \quad (3)$$

$$\text{Subject to: } T_{i,j} + \text{op}_{i,j} \leq T_{i,(j+1)} \quad i \in (1, n); \quad j \in (1, p_i), \quad (4)$$

$$\sum P_{(i,j),k} \leq 1 \quad (i, j) \in L(t); \quad k \in (1, m); \quad t \geq 0, \quad (5)$$

$$T_{i,j} \geq 0 \quad i \in (1, n); \quad j \in (1, p_i). \quad (6)$$

The indices and variables of the model are enumerated as follows: m : number of machines; n : number of jobs; i : job index; j : operation number index; p_i : number of operations of job i ; $T_{i,j}$: finish time of operation j th of job i ; $\text{op}_{i,j}$: processing time of operation j th of job i ; $L(t)$: set of operations being processed at time t .

Functions (1) and (2) infer the two objectives of JSSP. Constraints (3) ensure that the processing sequence of operations in each job is in accordance with the predetermined order. Constraints (4) demand that there is only one job on one machine at a time and (5) assures that the completion of all jobs should not be negative.

The conflicting character, where improving one objective may only be achieved when worsening another objective,

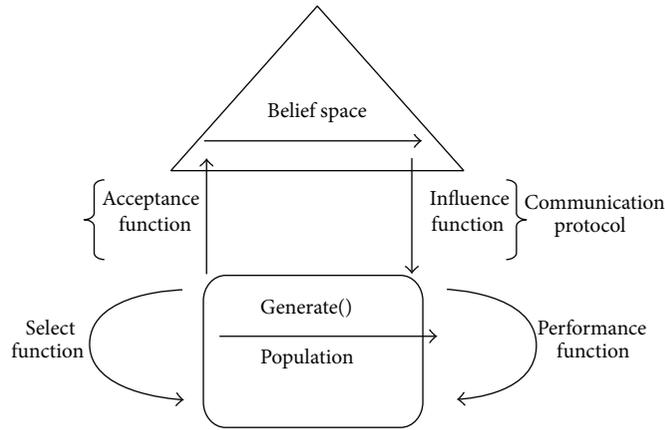


FIGURE 1: The cultural algorithm framework.

exists in objective functions generally. Therefore, obtaining an optimal scheduling solution that optimizes all the objectives is nearly impossible. However, there exists a set of equally efficient, nondominated, or noninferior solutions, known as the Pareto-optimal set. We recall the basic notion of efficient solution: a feasible solution X^* is Pareto optimal if there does not exist any other $X \in D$ such that $f_i(X) \leq f_i(X^*)$, for all i with at least one strict inequality. The set of all Pareto optima is called the Pareto optimal set and the set of all nondominant objective vectors is called Pareto front.

Many metaheuristic techniques have been proposed in the literature to search for near-optimal scheduling solutions. But the researches [30, 31] mainly focused on single machine or flow shop scheduling. Therefore, in this paper we deal with the multiobjective JSSP with makespan and mean flow time minimization.

3. A Novel Cultural Algorithm

The concept of culture can be defined in plenty of ways. Durham [32] described it as “a system of symbolically encoded conceptual phenomenon that are socially and historically transmitted within and between populations” in 1994. Therefore, culture can be taken as a vehicle for information storage. The information is accessible to all members of the society and can be useful in improving their problem solving activities. In other words, culture provides guidance and information to the new generation of a society. Without it, the only method for an individual to adapt to its environment is through plentiful trials and errors.

3.1. The Original Cultural Algorithm. Cultural algorithm is an evolution model through observing the cultural process in nature. It is a dual inheritance system that characterizes evolution at both the macroevolutionary level, which occurs at the belief space, and at the microevolutionary level, which takes place within the population space. These two spaces interface with each other through two functions: an acceptance function and an influence function. The cultural algorithm framework is shown in Figure 1.

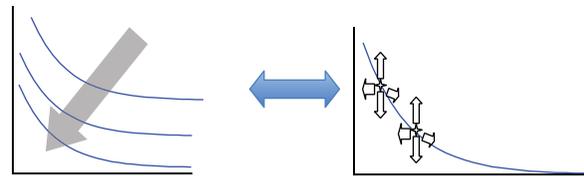


FIGURE 2: Overview of the exploration and exploitation processes.

As described in Figure 1, firstly the belief space and population space are initialized and the algorithm would repeat processing until a termination condition is met. The population space contains a set of possible solutions to the problem, and these individuals are evaluated by a performance function. Then an acceptance function decides which individuals in the current population should be selected and transformed to the current belief space. Those selected ones are adjusted with other individuals to update the belief space by an update function. Next, the updated knowledge in belief space is used to guide and influence the evolution in the population space through an influence function. Moreover, a selection function is used to select the population for the next generation. Most cultural-based algorithms applied the basic structure in Figure 1 while few modified it, such as literature [33], which designed a multilayer belief spaces structure.

3.2. The New Cultural Structure. It is well known that GA is not good at fine-tuning the solutions that are already close to the optimal solution, which means its local search ability is not as good as the global one. Hence, it is necessary to incorporate local search methods to find more effective optimal solutions, which is tabu search in this paper. We are not going to apply some local search after GA like many other researches do. The way we do it is to employ the two searching processes simultaneously with some interactions. The advantage of our approach is to gain more near-optimal solutions without suffering from premature convergence. The overview of the two processes is shown in Figure 2, in which the left one means exploration searching with genetic search and exploitation searching in the right one stands for TS.

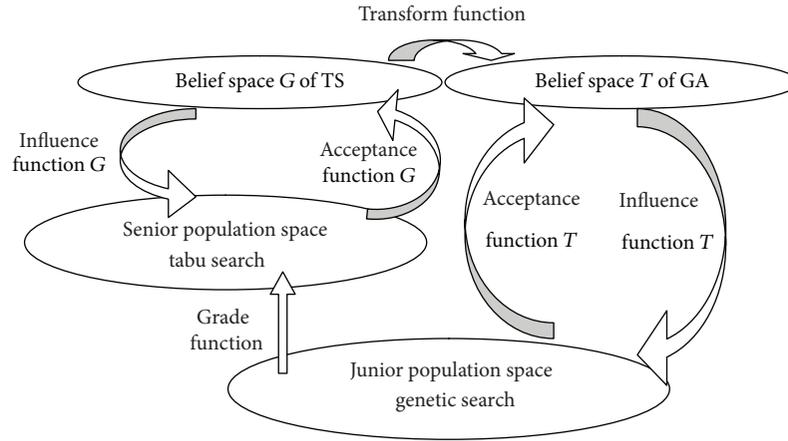


FIGURE 3: The framework of QSCGTA.

The proposed model, which consists of four spaces and dual evolution processes, takes advantage of a parallel and thorough search process compared to the original one. GA and TS are for the purpose of global and local search separately. The better individuals generated by genetic search in junior space are sent to the senior population space for further local search at regular or irregular intervals. The number and frequency of the individuals transmitted are determined by the grade function. Corresponding to the dual evolution process, there are two belief spaces guiding them separately toward the promising area through the influence functions and updated by the acceptance functions. And the best individuals in belief space of TS will be sent to the belief space of GA at predefined intervals by transform function. It is obvious that the advantage of the double search processes, besides the exploration and exploitation simultaneously, is that users can design different specific knowledge and influence functions for different goals. The flowchart of the advanced cultural algorithm is shown in Figure 3.

4. The Cultural-Based Genetic Tabu Algorithm for JSSP

Generally speaking, heuristic methods own advantages than exact methods in solving combinatorial optimization problems. Because it can provide more near-optimal solutions to decision makers. Therefore, we proposed the cultural genetic tabu algorithm to deal with the multiobjective JSSP.

4.1. The Representation and Schedule Builder

4.1.1. Representation. Chromosome representation is a key point in designing efficient evolutionary algorithms for constrained JSSPs. The reason is that different formulations in solutions correspond to different search spaces and different difficulties for further optimization operators. Although there have been all sorts of representation methods, generally speaking, these representations can be classified into the following two encoding approaches: direct approach and

indirect approach [34]. In direct approach, a schedule is encoded into a chromosome and algorithms are used to evolve those chromosomes to search for a better schedule. However, in the indirect approach, the chromosome is not a schedule, such as a sequence of dispatching rules for job assignment in priority rule-based representation. Algorithms then are used to find a better strategy to construct a schedule.

In our work, direct representation that belongs to operation-based representation with a schedule builder is used. This representation encodes a schedule as a sequence of operations and each gene represents one operation. To avoid the infeasibility raised by the precedence constraints, all operations for a job are named with the same symbol and interpreted by the order of occurrence in the given chromosome. This representation was employed by Bierwirth [35] and mathematically known as “permutation with repetition.” Considering a 3-job 3-machine problem, the chromosome contains $3 * 3$ genes, for example, given by

$$J_1 J_2 J_1 J_3 J_2 J_3 J_1 J_3 J_2 (1 \ 2 \ 1 \ 3 \ 2 \ 3 \ 1 \ 3 \ 2). \quad (6)$$

To obtain a feasible solution, the coding is interpreted as a task sequence

$$o_{11} o_{21} o_{12} o_{31} o_{22} o_{32} o_{13} o_{33} o_{23}. \quad (7)$$

By searching the permutation from left to right, a task o_{ij} of job J_i has to be scheduled on the determined machine by the technological order P_i . Therefore, one of the advantages of this representation is that any individual can be decoded to a feasible schedule. There does not exist the special case that processes the schedule operations whose technological predecessors have not been scheduled yet. Moreover, it shows another advantage in that the search template of the evolutionary search has no concern with details of particular scheduling problems. Last but not least, the decoding needed belongs to a simple mapping relation, which makes the decoding much easier.

The initial solutions are generated randomly with the length of $n * m$ in order to verify the robustness of our algorithm, which is one of the most important performances of algorithms [36].

```

Begin
for each  $o_{i,j}$ 
   $M = \text{findmachine}(o_{i,j});$     %find the corresponding machine
   $\text{count}_M = \text{count}_M + 1;$ 
   $\text{ldletime} = \text{scan}(M, T_{(i,j-1)});$  %scan the idle time on machine from the left to right
  if  $op_{i,j} < \text{ldletime}$ 
     $\text{LeftShift}(o_{i,j});$     %place the operation on that idle time
  else
     $\text{Place}(o_{i,j}, M, \text{count}_M);$  %place the operation at the end of the machine
  end
End
%all the operations have been scheduled
For each  $o_{i,j}$  %in each machine from the right to left
   $\text{SL}(o_{i,j}) = \text{lastesttime}(o_{i,j});$  %calculate the last starting time of the operation
   $\text{idletime2} = \text{scanreverse}(M, T_{(i,j+1)});$  %scan the machine from the right to left
  if  $\text{SL}(o_{i,j}) < \max(\text{idletime2})$ 
     $\text{Place}(o_{i,j}, M, \text{idletime2});$ 
  else
     $\text{Place}(o_{i,j}, M, \text{SL}(o_{i,j}));$ 
  end
end
modify the scheduling and output the results;
end

```

ALGORITHM 1: Illustration of the bidirectional decoding procedure.

4.1.2. Schedule Builder. After the representation, the chromosome must be transformed into a feasible schedule. And computational experiments performed in [37] showed that a powerful decoding strategy plays an important role in improving the final solutions in JSSP.

The schedule builder used in this paper is bidirectional decoding, which performs a kind of local search. The decoding allocates each operation on its assigned machine one by one in the order represented by the coding. When operation o_{ij} is scheduled on machine k , the idle time between operations that have already been processed on that machine is examined from left to right to find the earliest one that is not shorter than the process time of operation o_{ij} . If such an interval exists, it is allocated there, otherwise, it is allocated at the current end of machine j , which is called left shifting. After all the operations have been scheduled, we could obtain the latest starting time of each operation o_{ij} , represented by $\text{SL}(o_{ij})$ from right to left, which is the latest time at which operation o_{ij} can begin without delaying the makespan. Then the decoding allocates each operation with its latest starting time from right to left and shifting them orderly from right to left. This one is called right shifting. The left shifting has already proved to be effective in reducing the makespan. And right shifting procedure, which means reverse left shifting, can provide better performance along with left shifting.

The pseudocode of our proposed bidirectional shifting is as shown in Algorithm 1.

Through the bidirection shifting, the actual processing order of operations p and q may be opposite to the operation sequence vector. In order to allow offspring to inherit the information of their parents, the coding is reordered according to the operations' starting time in the decoded schedule.

4.2. Genetic Algorithm for Global Search. Genetic algorithms are stochastic search methods, containing complex interactions among parameters. The mechanics of the complex parameter interactions play an essential role in the performance of GA. Based on probability calculations and simulation results, Deb and Agrawal [38] observed that for simple problems, the mutation operator is the key search operator, while for complex problems, crossover operator plays an important role and performs well with an adequate population size. Based on these studies, it is recommended that the use of the crossover operator with an adequate population size is a reliable approach. Therefore, we prefer to apply the crossover with a large probability along with mutation with a small probability, as well as the analysis of population size with simulation results in the next section.

4.2.1. Crossover. Crossover is considered as the backbone of GA, which aims to inherit information of two parent solutions to offspring. Provided that the parents keep different aspects of better solutions, such as in multiobjective problems, crossover owns a good opportunity to find better offspring. Considering the repetition structure of the representation, crossover operators containing more genes should be applied instead of the one-point or two-point crossover.

In this paper, we applied two kinds of crossover and each applied with half possibility. Firstly the generalized order crossover [39] is used with a slight modification of getting two offspring in one crossover operation. We proposed a length of crossover-string between one-fourth and two-third of the total length of a chromosome. Secondly, the binary crossover, shown in Figure 4, is started by generating a binary string randomly, which consists by 0 and 1. The length of the string

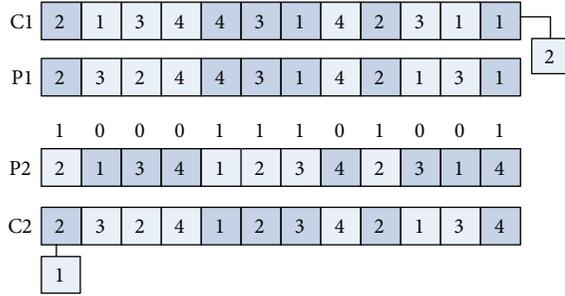


FIGURE 4: Illustration of the procedure of binary crossover.

is the same as the chromosomes. The offspring inherits the element of one parent at positions with bit 1 while inherits the element of the other one at positions with bit 0. And the redundant jobs would be replaced by the missing job sequencing. The other offspring is obtained likewise.

4.2.2. Mutation. Mutation is another important operator in GA and is usually performed after the crossover with a small probability. Considering the validation of the chromosome, we apply in this paper a swap mutation that needs no repair. Two different operations are picked randomly and then they exchange their positions.

The new individuals will be compared with the corresponding parent solutions by the following rules:

$$P_i^{t+1} = \begin{cases} P_i^t, & \text{if } \text{obj}(P_i^t) < \text{obj}(N_i) \\ N_i, & \text{otherwise,} \end{cases} \quad (8)$$

where obj is the objective functions, N_i is the i th new individual, and P_i^t is i th individual of the previous generation. And the rule of selecting individuals to perform crossover and mutation operators is 10-tournament selection to expand the selection of chromosomes, so that the ones with better performance have more chances to be chosen.

4.3. Tabu Algorithm for Further Local Search. Tabu search is one of the most efficient local search strategies for scheduling problems. It is obtained by transforming one solution to the next according to some neighborhood structures. The main elements of TS are the neighborhood structures, the tabu list length, and stopping rules. In our paper, because TS is a kind of point-search, the size should be much smaller than GA. It worked every ten iterations and shares the same stopping rules with global search. As for the neighborhood structure, we applied two kinds of neighborhood structures. Firstly, we chose the two-job exchange mutation as the neighborhood and an example of (1, 3) has been chosen as follows:

$$\begin{aligned} \text{Parent: } & \underline{2} \ \underline{3} \ \underline{2} \ \underline{4} \ \underline{4} \ \underline{3} \ \underline{1} \ \underline{4} \ \underline{2} \ \underline{1} \ \underline{3} \ \underline{1}, \\ \text{Child: } & \underline{2} \ \underline{1} \ \underline{2} \ \underline{4} \ \underline{4} \ \underline{1} \ \underline{3} \ \underline{4} \ \underline{2} \ \underline{3} \ \underline{1} \ \underline{3}. \end{aligned} \quad (9)$$

Secondly, the reversal of any two successive operations v and w on the critical path, where the two operations are not from the same job, was applied as another neighborhood structure. Because movements in the critical path

with certain constraints in multiobjective problems do not work remarkably as in single ones, movements between operations on different machines in the critical path have a certain possibility in improving the mean flow time while maintaining the same makespan. Last but not least, the tabu list length is decided by the experiments in Section 5.

4.4. Cultural Mechanism. The cultural mechanism in this paper plays a key role in guiding the evolution to promising areas, which consists of dual belief space, acceptance functions, influence functions, grade function, and transform function.

4.4.1. Grade Function and Transform Function. The grade function is proposed to decide how many and how frequent the individuals from GA should be sent to go through TS. Because of the single-point search nature of TS, we send a small population to TS compared to the population in GA. The number of grade population is self-adapted as shown in formula (10) and the frequency number is constant. And the transform function is to send one of the nondominated solutions to belief space of GA at a certain number of iterations

$$\begin{aligned} & \text{mize} \\ & = \begin{cases} \text{popsizeTS}, & \text{if } \text{exetime} = N \\ \text{popsizeTS} * A \\ * \exp\left(\frac{(1-g)}{(\text{max_iter} + 1 - g)}\right), & \text{if } \text{exetime} = n * N, \end{cases} \end{aligned} \quad (10)$$

where N means an interval the function works, A is constant coefficient, $n \in (1, \dots, (\text{max_iter}/N))$, max_iter infers the max generations, g is transforming times, and popsizeTS is the size of population for TS.

4.4.2. The Belief Space and Knowledge Structures. Generally speaking, the belief space of the cultural algorithm consists of several knowledge structures and is updated at a certain frequency. In our structure, there are two brief spaces that have different knowledge structures. Because our purpose is to enlarge the search space for GA in order to process global search while reduce the search space to guide the TS for local search, in brief space for GA, situational and topographical knowledge are adopted while situational and normative knowledge are applied for TS, while the history and domain knowledge are not applied for they are usually effective when the fitness landscape is dynamic.

The situational knowledge for GA consists of the best exemplars found along the evolution process. The structure is $[E_1, E_2, \dots, E_n]$. It records the nondominated set in GA. And the situational knowledge for TS is the same as the one for GA.

The normative knowledge represents the best district in the objective space and consists of two members (L_j^t, U_j^t) ,

which refers to lower and upper limits of a nondominated set of individuals. It means that

$$\begin{aligned} L_i^t &= \{\min \text{obj}_i(x^t), \forall x^t \in \text{NS}^t\} \quad i = 1, 2, \dots, N, \\ U_i^t &= \{\max \text{obj}_i(x^t), \forall x^t \in \text{NS}^t\} \quad i = 1, 2, \dots, N, \end{aligned} \quad (11)$$

where NS^t is the nondominated set, obj_i is the i th objective function, and N is the number of objectives.

The topographical knowledge is used to record the distribution of solutions to later help adapt the global acceleration. The space represented by normative knowledge is divided into grids of $s_1 \times s_2 \times \dots \times s_N$, where N is the number of objectives and s_i is the number of division in i th objective. Each cell will be represented by three elements $L_{\text{cell}} = [l_1, l_2, \dots, l_N]$, $U_{\text{cell}} = [u_1, u_2, \dots, u_N]$, and N_{cell} is the number of nondominated solutions in that cell. Consider

$$\begin{aligned} l_i^t &= L_i^t + p \times \frac{U_i^t - L_i^t}{s_i}, \quad i = 1, 2, \dots, N, \quad p = 0, 1, \dots, s_i - 1, \\ u_i^t &= l_i^t + \frac{U_i^t - L_i^t}{s_i}, \quad i = 1, 2, \dots, N. \end{aligned} \quad (12)$$

During each iteration, this knowledge will be updated to rebuild a new cell following the normative knowledge.

4.4.3. Acceptance Functions and Updating Brief Spaces and Knowledge. The knowledge of current belief space is updated by the individuals selected by acceptance function. The nondominated sets of populations are chosen to update the belief spaces.

The situational and normative knowledge are updated as follows:

$$\begin{aligned} E^{t+1} &= \begin{cases} E^t \cup x_i^t \setminus e, & \text{if } \exists e, \text{obj}(x_i^t) < \text{obj}(e) \\ E^t, & \text{if } \forall e, \text{obj}(e) < \text{obj}(x_i^t) \\ E^t \cup x_i^t, & \text{else,} \end{cases} \\ L_i^{t+1} &= \begin{cases} \text{obj}_i(x_k^t), & \text{if } \text{obj}_i(x_k^t) < L_i^t, x_k^t \in \text{NS}^t \\ L_i^t, & \text{otherwise} \end{cases} \\ U_i^{t+1} &= \begin{cases} \text{obj}_i(x_j^t), & \text{if } \text{obj}_i(x_j^t) > U_i^t, x_j^t \in \text{NS}^t \\ U_i^t, & \text{otherwise.} \end{cases} \end{aligned} \quad (13)$$

4.4.4. Influence Functions. Influence function is one of the key issues in cultural algorithms in guiding the evolutionary search. In this paper, we propose two totally different influence function structures for GA and TS.

The influence function applied on the genetic search consists of three parts. The first one is sending 10% individuals randomly to GA. This method will improve the quality of the population and then improve the evolution process.

The second one is applied to the mutation procedure. We apply the formula (14)~(15) to modify the mutation probability CM according to the topographical knowledge

$$\text{CM}^{t+1} = \begin{cases} \text{CM}^t + \frac{(N_{\text{best}}^t - N_{\text{best}}^{t+1})}{10}, & \text{if } N_{\text{best}}^t > N_{\text{best}}^{t+1} \\ \text{CM}^t - \frac{(N_{\text{best}}^{t+1} - N_{\text{best}}^t)}{10}, & \text{if } N_{\text{best}}^t < N_{\text{best}}^{t+1} \\ \text{CM}^t, & \text{otherwise,} \end{cases} \quad (14)$$

$$\text{CM}^{t+1} = \begin{cases} \text{CM}_{\text{max}}, & \text{if } \text{CM}^{t+1} > \text{CM}_{\text{max}} \\ \text{CM}_{\text{min}}, & \text{if } \text{CM}^{t+1} < \text{CM}_{\text{min}}, \end{cases} \quad (15)$$

where N_{best}^t is the number of nondominated ones in the cell where the best individual at t generation is located and N_{best}^{t+1} is the number of nondominated ones in the cell where the best individual at $t + 1$ generation is located. This formula implants a piecewise dynamic into the variation of the global acceleration. The values of N_{best}^t and N_{best}^{t+1} are stored in the topographical knowledge and CM should be limited in a range of $[\text{CM}_{\text{min}}, \text{CM}_{\text{max}}]$ according to formula (15).

And the last one is towards crossover. This is inspired by the phenomenon called Atavism, which is a theory in heredity holding that the reappearance of a characteristic in an organism after several generations of absence is usually caused by the chance recombination of genes. Therefore, we make the best individuals in situational knowledge crossover with the worst individuals in GA expecting that the offspring in several generations will inherit the good gene from the best individuals. We use the topographical knowledge stored in the belief space to select the best individual. Firstly, we use the roulette wheel selection to choose the least populated cell, and then randomly select a nondominated individual from that cell to be the best individual. Each cell is assigned a fitness of

$$\text{fitness}_{\text{cell}} = \frac{5}{N_{\text{cell}}}, \quad (16)$$

where N_{cell} is the number of nondominated ones located in that cell.

The influence function for TS consists of a combined effect by both situational knowledge and normative knowledge. Its strategy is, as formula (17) shows, towards tabu length L :

$$\begin{aligned} \text{Len}^{t+1} &= \begin{cases} \text{Len}^t + \Delta L, & \text{if } \text{obj}_i(x^t) < L_i^t \\ & \text{or } \text{obj}(x^t) > \text{obj}(e) \\ \text{Len}^t - \Delta L, & \text{if } \text{obj}(x^t) < \text{obj}(e) \\ \text{Len}^t, & \text{otherwise,} \end{cases} \\ \text{Len}^{t+1} &= \begin{cases} L_{\text{max}}, & \text{if } \text{Len}^{t+1} > L_{\text{max}} \\ L_{\text{min}}, & \text{if } \text{Len}^{t+1} < L_{\text{min}}. \end{cases} \end{aligned} \quad (17)$$

4.4.5. Global Archive. We design a limited archive to store the best solutions. And each new nondominated solution from

the two processes will be compared with members in the archive. If P_{new} dominates any member of the archive, P_{new} will replace the member in the archive. If P_{new} is dominated by any member of the archive, P_{new} will be disregarded. And if P_{new} neither dominates nor is dominated by the members of the archive, there will be two parts. If the archive is not full, P_{new} would be added to archive. However, if it is full, P_{new} would replace the one with the shortest distance between adjacent individuals. And the members in the archive take part in the selection process of GA along with the population.

To sum up, we proposed a novel hypergenetic algorithm incorporated with tabu search under the frame of cultural algorithm. The flowchart of our QSCGTA is shown in Figure 5.

5. Experimental Results and Discussion

5.1. The Testing Problem. Computational experiments are carried out to investigate the performance of our proposed cultural genetic tabu search. In order to evaluate the performance of our algorithm, we run the algorithm on a series of benchmark problems from the OR-Library (<http://people.brunel.ac.uk/~mastjib/jeb/info.html>), which is regarded as the standard testing problems in scheduling. The testing benchmark problems in this paper include MT problems, ABZ problems, and LA problems.

All the programs in the experiments were written in Matlab and all the experiments were running on platform using Intel Core 4 Quad 2.4 GHZ CPU with 2 GB RAM. First we applied orthogonal experiments to decide our parameter settings in the QSCGTA. And then we compared our algorithm with other well-known multiobjective evolutionary algorithms, which are NSGAI, SPEA2, and MPSO.

A set of solutions that are superior to the rest of the solutions exists in multiobjective optimization. Therefore, new approaches, which differ from the single objective, are required to compare the performance of the algorithms. The performance measures that are used are as follows.

- (1) Number of Pareto solution (NPS): this performance measurement is calculated by counting the number of nondominated solutions obtained. A larger number corresponds to better performance.
- (2) Hyperarea ratio (HR) [40]: this indicator shows the ratio between the area dominated by solutions and the entire solution area. A bigger ratio corresponds to a better solution. Given the difficulty in deciding the extent of the entire solution area, we used the rectangle constituted by the maximum and minimum of each objective as PF_{TRUE} :

$$HR = \frac{H_{PF_{\text{KNOWN}}}}{H_{PF_{\text{TRUE}}}}. \quad (18)$$

- (3) Spread of nondominance solutions (SNS): this criterion, which is known as an indicator of diversity, is calculated through the following formula. Larger

TABLE 1: Levels of the parameters.

Levels	PS (A)	CP (B)	MP (C)	TL (D)
1	50	0.6	0.1	5
2	100	0.7	0.2	10
3	150	0.8	0.3	15
4	200	0.9	0.4	20

values of this criterion correspond to higher quality solutions:

$$SNS = \sqrt{\frac{\sum_{i=1}^n (\text{MID} - c_i)^2}{n-1}},$$

$$\text{MID} = \frac{\sum_{i=1}^n c_i}{n}, \quad (19)$$

$$c_i = \sqrt{f_{1i}^2 + f_{2i}^2},$$

where n is the number of nondominated solutions and f_{1i} and f_{2i} are the values of the two objectives.

5.2. Computation Results

5.2.1. Parameters Selection through Orthogonal Experiment. The parameters, which can be uncertainty, had a great influence on the performance of the algorithm [41]. There have been several parameters that should be set properly in order to make the algorithm more effective, which are population size of genetic search (PS), the crossover possibility (CP), the mutation possibility (MP), and tabu list length (TL).

If the authors apply the factorial experiment to design the parameters, it requires $4^4 = 256$ experiments. It is a waste of time and resource. Therefore, the orthogonal experiment is carried out to tune the adjustable parameters of our algorithm, which just needs $4^2 = 16$ experiments.

The orthogonal experiment used here is to design experiments to investigate how different parameters affect the mean and variance of a process performance characteristic. The experiment design involves using orthogonal arrays to organize the parameters, which affect the process, and levels at which the parameters should be varies.

Generally first we considered the hyperarea ratio of problem MT10, LA25, LA28, LA36, and ABZ7 for ten times at 500 generations as the objective. Then we chose the four levels for population size of genetic search (PS), the crossover possibility (CP), the mutation possibility (MP), tabu list length (TL). The levels of all the parameters are listed in Table 1.

Here we chose the L16 orthogonal array and Table 2 shows the experiments results of MT10. The best HR of each column is shown in bold. Then after the range analysis, it is obvious that the major one is the size of population followed by mutation possibility and crossover possibility. And it also shows that the parameters PS = 150, CP = 0.8, MP = 0.4, and TL = 15 work better for MT10.

The parameter settings for the other four instances are applied the same way as MT10 and the results are shown in

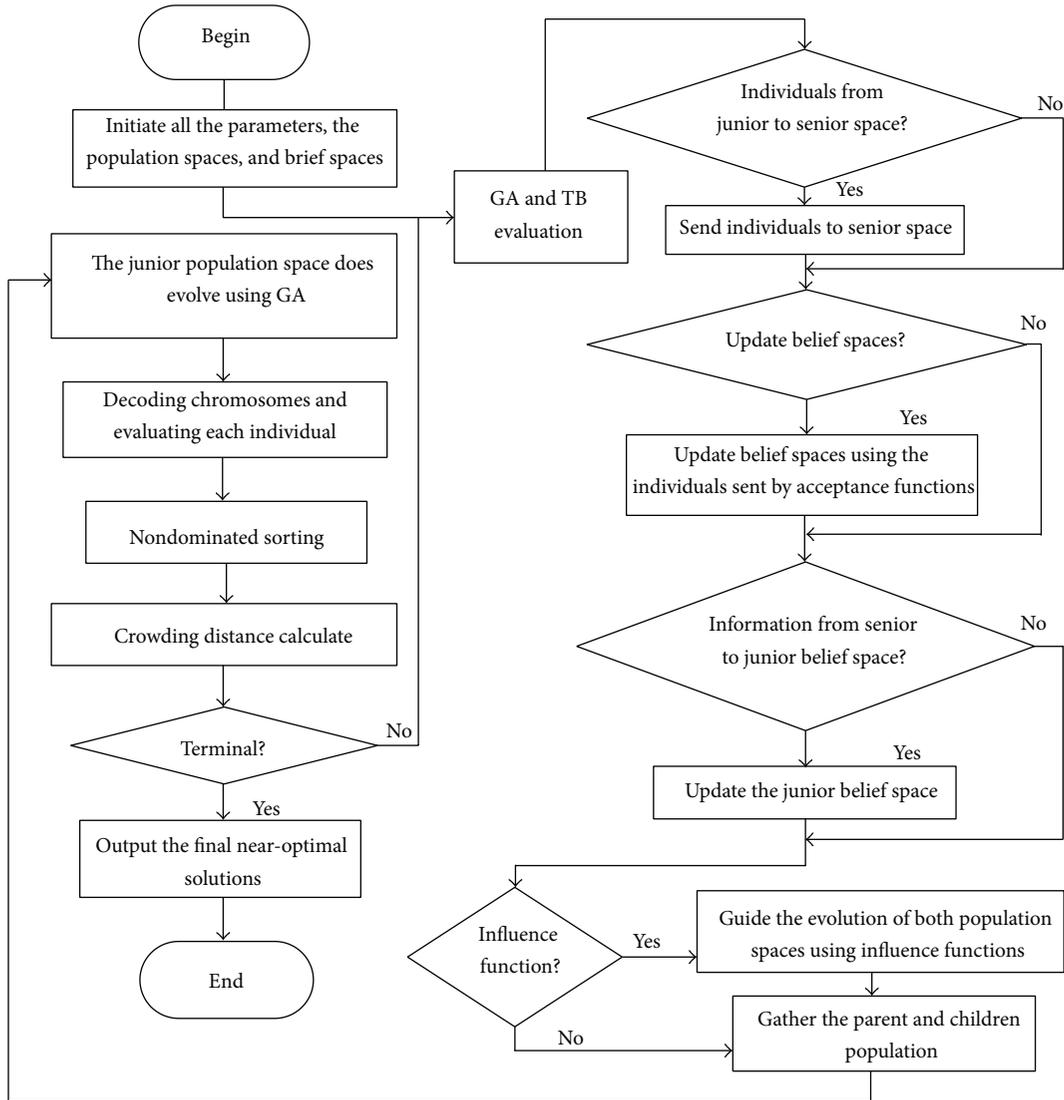


FIGURE 5: The flowchart of multiobjective QSCGTA.

Tables 3, 4, 5, and 6. The best HR of each column is shown in bold. Based on the data in tables, the best parameters settings are not completely consistent in all the problems. Concluding from all these tables, the authors set 150 for population, 0.4 for mutation possibility, 0.8 for crossover possibility, and 15 for tabu list length.

5.2.2. *Multiobjective Comparison.* Multiobjective optimization has two goals: one is to find a set of solutions as close as possible to the Pareto front, while the other is to find a set of solutions as diverse as possible.

The parameter settings of our algorithm are the same as above and the one for MPSO is the same as literature [29]. The parameters for NSGAI and SPEA2 are decided by the aforementioned orthogonal experiment and the settings are as follows.

- (1) The initial population is randomly generated and the number is set to 150.

- (2) Tournament selection is chosen.
- (3) The GOX and swap are used as crossover and mutation operators.
- (4) The ratios of GOX and swap are set to 0.8 and 0.4, respectively.
- (5) The size of archive is set to 40 and number of iteration is set to 500.

Each benchmark problem was tested for twenty times with different seeds. Then all the final generations were combined and the nondominated sorting was performed to constitute the final nondominated solutions.

The results in Table 7 showed that QSCGTA has the highest average hyperarea ratios in 14 of 18 instances, followed by PSO with 3 instances, SPEA2 with one instance, and NSGAI with none. With regard to number and spread of nondominance solutions, our approach found more near-optimal solutions with diversity than MPSO, SPEA2, and

TABLE 2: Range analysis of the orthogonal experiment for MT10.

	A	B	C	D	Empty	Results (HR)
	1	2	3	4	5	
1	1	1	1	1	1	0.0027
2	1	2	2	2	2	0.0635
3	1	3	3	3	3	0.5675
4	1	4	4	4	4	0.2822
5	2	1	2	3	4	0.4233
6	2	2	1	4	3	0.2439
7	2	3	4	1	2	0.5076
8	2	4	3	2	1	0.0566
9	3	1	3	4	2	0.5441
10	3	2	4	3	1	0.8328
11	3	3	1	2	4	0.6879
12	3	4	2	1	3	0.4633
13	4	1	4	2	3	0.9712
14	4	2	3	1	4	0.5825
15	4	3	2	4	1	0.4239
16	4	4	1	3	2	0.4315
k_1	0.2290	0.4853	0.3415	0.3890	0.3290	
k_2	0.3079	0.4307	0.3435	0.4448	0.3867	
k_3	0.6320	0.5465	0.4377	0.5638	0.5615	
k_4	0.6023	0.3084	0.6485	0.3735	0.4940	
Range	0.4030	0.2381	0.3070	0.1903	0.2325	
Order of test factors			A > C > B > D			
Optimal level	A3	B3	C4	D3		

TABLE 3: Results of parameter settings for LA25.

LA25	A	B	C	D
k	1	2	3	4
k_1	0.4387	0.4048	0.5426	0.3867
k_2	0.7944	0.5185	0.3827	0.4956
k_3	0.2409	0.5372	0.3646	0.5208
k_4	0.4611	0.4747	0.6453	0.5320
Range	0.5535	0.1324	0.2807	0.1453

TABLE 5: Results of parameter settings for LA36.

LA36	A	B	C	D
k	1	2	3	4
k_1	0.4552	0.3858	0.5321	0.4192
k_2	0.3872	0.3948	0.4202	0.3370
k_3	0.6345	0.6725	0.4628	0.4482
k_4	0.4889	0.5127	0.5507	0.4614
Range	0.2473	0.2866	0.1305	0.1112

TABLE 4: Results of parameter settings for LA28.

LA28	A	B	C	D
k	1	2	3	4
k_1	0.4724	0.3571	0.3343	0.3309
k_2	0.2277	0.6099	0.4581	0.4901
k_3	0.4347	0.4316	0.5659	0.3810
k_4	0.6196	0.3556	0.4961	0.3924
Range	0.3919	0.2543	0.2315	0.1592

TABLE 6: Results of parameter settings for ABZ7.

ABZ7	A	B	C	D
k	1	2	3	4
k_1	0.2854	0.3627	0.2734	0.3707
k_2	0.2108	0.2395	0.3141	0.3541
k_3	0.6236	0.1939	0.5829	0.3837
k_4	0.4304	0.5540	0.3798	0.2416
Range	0.2196	0.3601	0.2095	0.1422

NSGAI as well. Because the application of TS along with GA guides the search with more possibilities towards promising areas than traditional TS. Moreover it makes the search process converge more quickly than traditional GA. Besides, the cultural frame quickens the two searching processes.

To further test the stability of algorithms, we run the four algorithms twenty times independently, under the aforementioned environment, on randomly selected instances with different sizes, which are LA03, MT10, LA17, LA27, LA40, and ABZ7. The box plots of hyperarea ratio on these

TABLE 7: Results of algorithms on multiobjective JSSP.

	$N * M$	NSGAI			SPEA2			MPSO			QSCGTA		
		NPS	HR	SNS	NPS	HR	SNS	NPS	HR	SNS	NPS	HR	SNS
LA03	10 * 5	2	0.44	5.17	5	0.48	2.71	4	0.48	3.12	5	0.99	8.94
LA04	10 * 5	1	0.42	—	2	0.94	1.96	2	0.94	1.96	2	0.94	1.96
LA05	10 * 5	1	—	—	1	—	—	1	—	—	1	—	—
Mt10	10 * 10	3	0.53	1.11	1	0.69	—	3	0.93	2.13	4	0.94	12.00
LA16	10 * 10	2	0.45	5.19	5	0.62	4.85	3	0.60	21.56	6	0.86	10.41
LA17	10 * 10	1	0.39	—	7	0.60	14.17	6	0.75	12.87	7	0.89	18.06
LA18	10 * 10	3	0.42	7.70	3	0.52	1.75	5	0.79	2.81	5	0.78	9.70
LA22	15 * 10	4	0.32	9.31	6	0.69	18.23	3	0.73	3.15	2	0.99	23.54
LA23	15 * 10	3	0.51	5.25	1	0.43	—	1	0.69	—	7	0.79	22.00
LA25	15 * 10	2	0.54	9.42	7	0.86	1.68	10	0.77	9.36	5	0.95	6.07
LA27	20 * 10	2	0.44	4.10	3	0.59	5.86	2	0.70	6.07	5	0.83	4.62
LA28	20 * 10	3	0.52	6.38	3	0.54	3.42	3	0.71	8.88	5	0.82	9.67
LA29	20 * 10	1	0.29	—	2	0.81	5.28	4	0.64	7.33	2	0.67	7.22
LA36	15 * 15	3	0.38	9.15	6	0.52	3.89	3	0.63	3.92	8	0.93	14.71
LA38	15 * 15	1	0.30	—	2	0.75	14.25	2	0.79	12.23	4	0.95	7.35
LA40	15 * 15	4	0.37	7.69	1	0.61	—	3	0.96	1.19	5	0.81	7.73
ABZ7	20 * 15	3	0.43	2.51	5	0.52	4.03	3	0.63	0.79	4	0.95	11.77
ABZ8	20 * 15	1	0.37	—	2	0.78	0.21	1	0.52	—	5	0.87	5.77
ABZ9	20 * 15	3	0.52	9.95	6	0.70	1.79	3	0.77	26.84	2	0.73	13.32

TABLE 8: Performance of algorithms on instances generated randomly.

$N * M$	NSGAI		SPEA2		MPSO		QSCGTA	
	HR	SNS	HR	SNS	HR	SNS	HR	SNS
10 * 10	0.39	9.06	0.37	13.59	0.50	5.49	0.46	6.49
10 * 15	0.34	8.85	0.50	6.10	0.75	6.36	0.85	8.85
10 * 20	0.51	5.19	0.72	4.79	0.81	6.89	0.81	11.58
15 * 15	0.22	5.32	0.48	4.66	0.70	8.15	0.83	12.14
15 * 20	0.43	4.58	0.61	6.34	0.78	6.67	0.87	8.20
20 * 20	0.37	3.14	0.50	5.17	0.68	7.99	0.92	7.24

instances are shown in Figure 6. This figure clearly indicates that the QSCGTA outperforms the NSGAI and SPEA2 both in stability and distribution. With regard to MPSO, QSCGTA is with a less concentrated distribution, however, with better optimal and mean. Moreover, observing from the computation time comparison in Figure 7, QSCGTA requires about a quarter of time MPSO needed, half time of NSGAI, and slightly larger than SPEA2. In other words, QSCGTA owns advantages over the other three algorithms.

Moreover, to further verify the performance of our algorithm, the performance for HR is computed for each algorithm on randomly generated instances. Several problem scenarios were generated by varying one or more of the number of machines and number of jobs. For each scenario, 20 problems were randomly generated by setting operation processing times from a uniform distribution in the interval [5, 100], and the average values across these instances were recorded. All jobs had randomly assigned routing through the system. The results on instances generated randomly,

shown in Table 8, also validated the exploration and exploitation of our QSCGTA.

6. Conclusions

Multiobjective scheduling has become the main research field in scheduling problems because of the multiobjective character, by nature, of many real-world scheduling problems. Due to the complexity of the job shop scheduling problem, many researches have been focused on multiobjective single machine problems or flow shop problems. The researches on multiobjective job shop problems are very rare. Therefore, in this paper, we have proposed QSCGTA for solving the multiobjective JSSP. The GA and TS have been incorporated in the frame of a novel cultural algorithm to search for the Pareto-optimal schedules.

The experiments indicated that our approach is suitable for applied benchmark problems and obviously yielded better performance in terms of solutions, stability, and computation

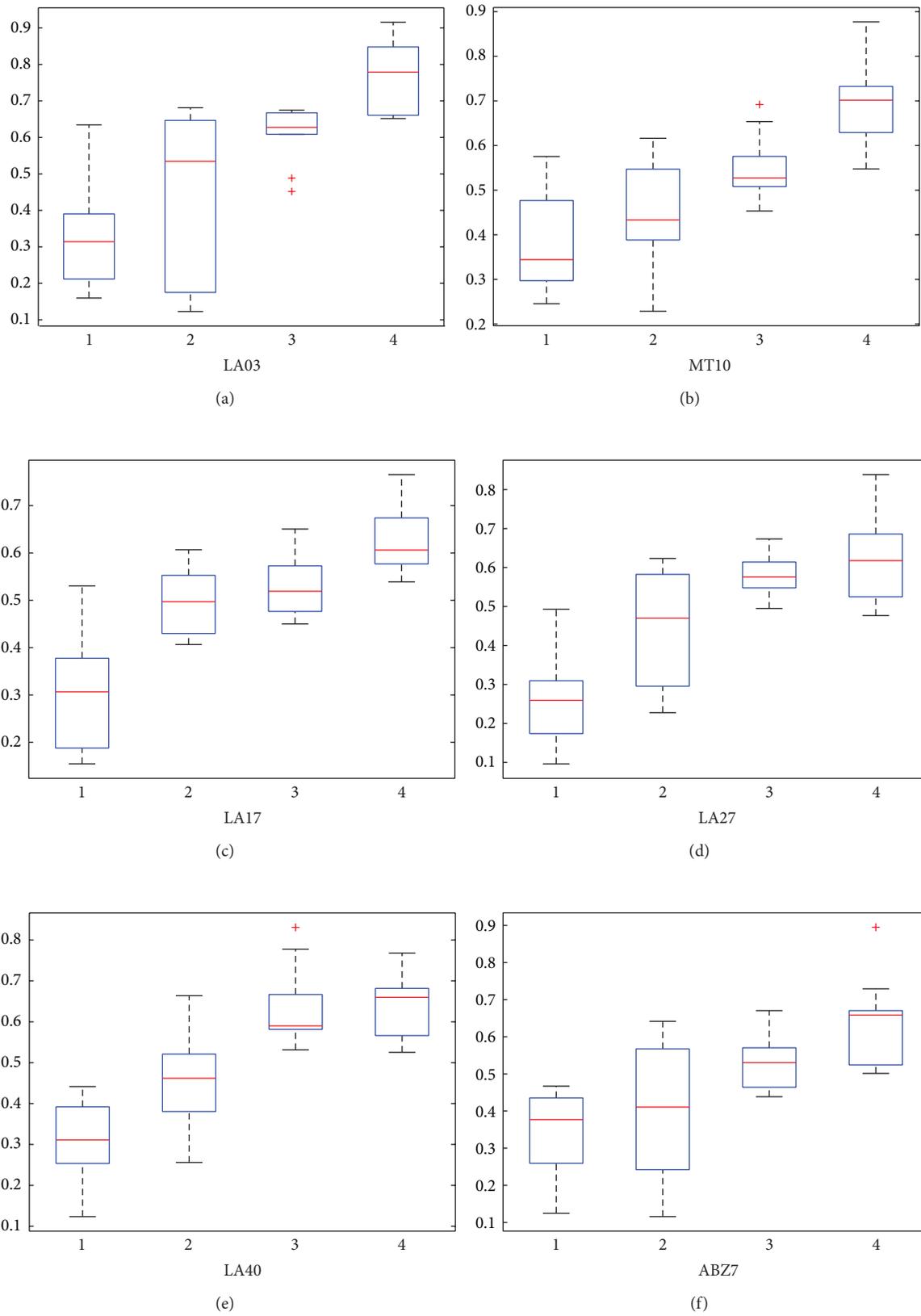


FIGURE 6: The box plots of hyperarea ratio for different problems. Column numbers refer to (1) NSGAI, (2) SPEA2, (3) MPSO, and (4) QSCGTA.

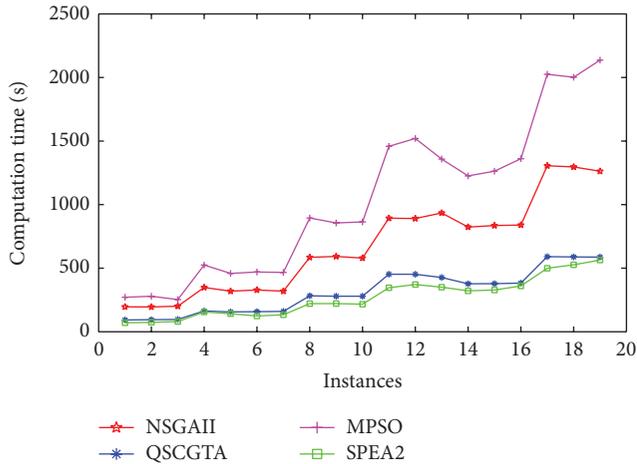


FIGURE 7: Computational time of four algorithms for different problems.

time compared with the other three algorithms. The main strength of our approach is in combination with global and local search under a novel cultural algorithm frame in order to produce diverse solutions while maintaining the convergence of the nondominated solutions. In each generation, only a predefined number of the best solutions of GA is selected for applying TS. It provides more diversity toward Pareto-optimal solutions. All in all, our proposed QSCGTA-based evolutionary scheduling approach accomplished the goals of multiobjective job shop scheduling problems both in convergence and diversity.

More comprehensive studies can be applied to extend the QSCGTA. Other possible criteria in multiobjective optimization will be considered. Furthermore, more local search methods will be analyzed to integrate to the QSCGTA algorithm.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors appreciate the support of National Natural Science Foundation of China (Grant nos. 61174040 and 61104178), Shanghai Commission of Science and Technology (Grant no. 12JC1403400), and Fundamental Research Funds for the Central Universities.

References

[1] J. F. Gonçalves, J. J. M. Mendes, and M. G. C. Resende, "A hybrid genetic algorithm for the job shop scheduling problem," *European Journal of Operational Research*, vol. 167, no. 1, pp. 77–95, 2005.

[2] H. Zhang, H. Yan, F. Yang, and Q. Chen, "Quantized control design for impulsive fuzzy networked systems," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 6, pp. 1153–1162, 2011.

[3] R. G. Robert, "An introduction to cultural algorithms," in *Proceeding of the 3rd Annual Conference Evolution Programming*, pp. 131–136, Singapore, 1994.

[4] C. Chung and R. G. Reynolds, "A testbed for solving optimization problems using cultural algorithms," in *Proceedings of the 5th Annual Conference on Evolutionary Programming*, pp. 225–236, Cambridge, Mass, USA, 1996.

[5] R. G. Reynolds and X. Jin, "Using region-schema to solve nonlinear constraint optimization problems: a culture algorithm approach," in *Proceedings of the Conference in honor of John H. Holland*, The Netherlands, 1999.

[6] S. M. Saleem, *Knowledge-based solution to dynamic optimization problems using cultural algorithms [Ph.D. thesis]*, Wayne State University, Detroit, Mich, USA, 2001.

[7] C. J. Chung, *Knowledge-based approaches to self-adaptation in cultural algorithms [Ph.D. thesis]*, Wayne State University, Detroit, Mich, USA, 1997.

[8] S. Zhu and R. G. Reynolds, "Fuzzy cultural algorithms with evolutionary programming," *Evolutionary Programming VII*, vol. 1447, pp. 209–218, 1998.

[9] A. V. Abs da Cruz, M. A. C. Pacheco, M. Vellasco, and C. R. H. Barbosa, "Cultural operators for a quantum-inspired evolutionary algorithm applied to numerical optimization problems," in *Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach*, vol. 3562, pp. 1–10, 2005.

[10] M. Daneshyari and G. G. Yen, "Constrained multiple-swarm particle swarm optimization within a cultural framework," *IEEE Transactions on Systems, Man, and Cybernetics A*, vol. 42, no. 2, pp. 475–490, 2012.

[11] Y. Sun, L. B. Zhang, and X. S. Gu, "A hybrid co-evolutionary cultural algorithm based on particle swarm optimization for solving global optimization problems," *Neurocomputing*, vol. 98, pp. 76–89, 2012.

[12] X. D. Jin, *Solving constrained optimization problems using cultural algorithms and regional schemata [Ph.D. thesis]*, Wayne State University, Detroit, Mich, USA, 2001.

[13] M. Daneshyari and G. G. Yen, "Cultural-based multiobjective particle swarm optimization," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 41, no. 2, pp. 553–567, 2011.

[14] H. F. Zhang, J. Z. Zhou, N. Fang et al., "Daily hydrothermal scheduling with economic emission using simulated annealing technique based multi-objective cultural differential evolution approach," *Energy*, vol. 50, pp. 24–37, 2013.

[15] W. E. Smith, "Various optimizers for single-stage production," *Naval Research Logistics Quarterly*, vol. 3, pp. 59–66, 1956.

[16] E. P. C. Kao, "A multiple objective decision theoretic approach to one-machine scheduling problems," *Computers and Operations Research*, vol. 7, no. 4, pp. 251–259, 1980.

[17] R. L. Daniels, "Incorporating preference information into multi-objective scheduling," *European Journal of Operational Research*, vol. 77, no. 2, pp. 272–286, 1994.

[18] A. H. Kashan, B. Karimi, and F. Jolai, "An effective hybrid multi-objective genetic algorithm for bi-criteria scheduling on a single batch processing machine with non-identical job sizes," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 6, pp. 911–922, 2010.

[19] P.-C. Chang and S.-H. Chen, "The development of a sub-population genetic algorithm II (SPGA II) for multi-objective

- combinatorial problems,” *Applied Soft Computing Journal*, vol. 9, no. 1, pp. 173–181, 2009.
- [20] G. Zhang, X. Shao, P. Li, and L. Gao, “An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem,” *Computers and Industrial Engineering*, vol. 56, no. 4, pp. 1309–1318, 2009.
- [21] Y. Xu, R. Qu, and R. Li, “A simulated annealing based genetic local search algorithm for multi-objective multicast routing problems,” *Annals of Operations Research*, vol. 206, pp. 527–555, 2013.
- [22] T. Loukil, J. Teghem, and P. Fortemps, “A multi-objective production scheduling case study solved by simulated annealing,” *European Journal of Operational Research*, vol. 179, no. 3, pp. 709–722, 2007.
- [23] S. H. Niu, S. K. Ong, and A. Y. C. Nee, “An improved intelligent water drops algorithm for solving multi-objective job shop scheduling,” *Engineering Applications of Artificial Intelligence*, vol. 26, pp. 2431–2442, 2013.
- [24] K. Huckert, R. Rhode, O. Roglin, and R. Weber, “On the interactive solution to a multicriteria scheduling problem,” *Zeitschrift für Operations Research*, vol. 24, no. 1, pp. A47–A60, 1980.
- [25] D. Lei, “Simplified multi-objective genetic algorithms for stochastic job shop scheduling,” *Applied Soft Computing Journal*, vol. 11, no. 8, pp. 4991–4996, 2011.
- [26] L. P. Zhang, L. Gao, and X. Y. Li, “A hybrid genetic algorithm and tabu search for a multi-objective dynamic job shop scheduling problem,” *International Journal of Production Research*, vol. 51, no. 12, pp. 3516–3531, 2013.
- [27] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [28] E. Zitzler, M. Laumanns, and L. Thiele, “SPEA2: improving the strength pareto evolution algorithm for multiobjective optimization,” in *Proceedings of the Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems (EUROGEN '01)*, 2001.
- [29] R. Tavakkoli-Moghaddam, M. Azarkish, and A. Sadeghnejad-Barkousaraie, “A new hybrid multi-objective Pareto archive PSO algorithm for a bi-objective job shop scheduling problem,” *Expert Systems with Applications*, vol. 38, no. 9, pp. 10812–10821, 2011.
- [30] V. T’kindt, N. Monmarché, F. Tercinet, and D. Laügt, “An ant colony optimization algorithm to solve a 2-machine bicriteria flowshop scheduling problem,” *European Journal of Operational Research*, vol. 142, no. 2, pp. 250–257, 2002.
- [31] R. Qing-dao-er-ji and Y. Wang, “Inventory based bi-objective flow shop scheduling model and its hybrid genetic algorithm,” *Mathematical Problems in Engineering*, vol. 2013, Article ID 976065, 7 pages, 2013.
- [32] W. Durham, *Co-Evolution: Genes, Culture, and Human Diversity*, Stanford University, Stanford, Calif, USA, 1994.
- [33] H.-Y. Huang, X.-S. Gu, and M.-D. Liu, “Research on cultural algorithm for solving nonlinear constrained optimization,” *Acta Automatica Sinica*, vol. 33, no. 10, pp. 1115–1120, 2007.
- [34] R. Cheng, M. Gen, and Y. Tsujimura, “A tutorial survey of job-shop scheduling problems using genetic algorithms—I. Representation,” *Computers and Industrial Engineering*, vol. 30, no. 4, pp. 983–997, 1996.
- [35] C. Bierwirth, “A generalized permutation approach to job shop scheduling with genetic algorithms,” *OR Spektrum*, vol. 17, no. 2-3, pp. 87–92, 1995.
- [36] H. Yan, Z. Su, H. Zhang, and F. Yang, “Observer-based H_∞ control for discrete-time stochastic systems with quantisation and random communication delays,” *IET Control Theory & Applications*, vol. 7, no. 3, pp. 372–379, 2013.
- [37] T. Yamada, *Studies on meta heuristics for jobshop and flowshop scheduling problems [Ph.D. thesis]*, Kyoto University, Kyoto, Japan, 2003.
- [38] K. Deb and S. Agrawal, “Understanding interactions among genetic algorithm parameters,” in *Foundations of Genetic Algorithms V*, pp. 265–286, Morgan Kaufmann, San Mateo, Calif, USA, 1999.
- [39] C. Bierwirth, D. C. Matfield, and H. Kopfer, “On permutation representation for scheduling problems,” in *Parallel Problem Solving from Nature*, vol. 1141, pp. 310–318, 1996.
- [40] E. Zitzler, *Evolutionary algorithms for multiobjective optimization: methods and applications [Ph.D. thesis]*, Shaker Verlag, Aachen, Germany, 1999.
- [41] H. Zhang, Z.-H. Guan, and G. Feng, “Reliable dissipative control for stochastic impulsive systems,” *Automatica*, vol. 44, no. 4, pp. 1004–1010, 2008.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

