

## Research Article

# Multipopulation Genetic Algorithms with Different Interaction Structures to Solve Flexible Job-Shop Scheduling Problems: A Network Science Perspective

Ding-Shan Deng <sup>1</sup>, Wei Long <sup>1</sup>, Yan-Yan Li <sup>1</sup> and Xiao-Qiu Shi <sup>2</sup>

<sup>1</sup>School of Mechanical Engineering, Sichuan University, Chengdu 610065, China

<sup>2</sup>School of Manufacturing Science and Engineering, Southwest University of Science and Technology, Mianyang 621000, China

Correspondence should be addressed to Yan-Yan Li; [yyl\\_scu@163.com](mailto:yyl_scu@163.com)

Received 22 April 2020; Revised 1 July 2020; Accepted 24 September 2020; Published 27 November 2020

Academic Editor: Manjit Kaur

Copyright © 2020 Ding-Shan Deng et al. 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.

Populations of multipopulation genetic algorithms (MPGAs) parallelly evolve with some interaction mechanisms. Previous studies have shown that the interaction structures can impact on the performance of MPGAs to some extent. This paper introduces the concept of complex networks such as ring-shaped networks and small-world networks to study how interaction structures and their parameters influence the MPGAs, where subpopulations are regarded as nodes and their interaction or migration of elites between subpopulations as edges. After solving the flexible job-shop scheduling problem (FJSP) by MPGAs with different parameters of interaction structures, simulation results were measured by criteria, such as success rate and average optimal value. The analysis reveals that (1) the smaller the average path length (APL) of the network is, the higher the propagation rate will be; (2) the performance of MPGAs increased first and then decreased along with the decrease of APL, indicating that, for better performance, the networks should have a proper APL, which can be adjusted by changing the structural parameters of networks; and (3) because the edge number of small-world networks remains unchanged with different rewiring possibilities of edges, the change in performance indicates that the MPGA can be improved by a more proper interaction structure of subpopulations as other conditions remain unchanged.

## 1. Introduction

Genetic algorithm (GA) [1], an original metaheuristic, is easy to fall into local optima when employed to solve resource-constrained project scheduling problems [2] such as flexible job-shop schedule problems (FJSPs) due to the complexity of searched space and high dimensions [3].

To maintain population diversity (enhancing the search diversity) and avoid premature convergence, a multipopulation genetic algorithm (MPGA) [4] is one feasible method where subpopulations are generated and individuals migrate periodically among them. In the evolutionary process, when intra-subpopulation evolution pushes individuals towards different local optima, migration can introduce new genes into the subpopulations [5]. As in [5, 6],

the migration can be implemented in different topologies that define how subpopulations are connected, which can influence the outcomes of MPGAs to some extent.

To more effectively research on the performance with different interaction structures between subpopulations, we can consider subpopulations as nodes and their interaction or migration of elites between subpopulations as edges so that MPGA can be regarded as complex networks. Similarly, but not identically, Du et al. [7] proposed the networked evolutionary algorithm where nodes represent information process units, i.e., individuals, and connections denote information transmission links. Payne et al. [8] shed light on dynamic population structures, wherein edges are dynamically rewired according to the genotypic or phenotypic properties of individuals or according to the success of prior

interindividual interaction. Chao et al. [9, 10] proposed a multiobjective cellular grey wolf optimizer with a topological structure for hybrid flow-shop scheduling, in which each wolf or individual can be regarded as a grid of a lattice structure, and the interaction among them is restricted to the neighborhood. In the aforementioned three papers, the nodes of networks were regarded as individuals; however, this paper mainly discusses the interactions of subpopulations. For example, Leitao et al. [5] conducted simulations to study how the topology of island models (subpopulations) impacts the effectiveness and diversity of evolutionary algorithms with three types of networks, i.e., the ring and torus, as well as the fully connected networks. Besides, Zhang and Li [11] proposed a multiobjective evolutionary algorithm based on decomposition (MOEA/D). It can also be regarded as a multipopulation algorithm, and each subpopulation has only one individual. Fu et al. [12] extended and used the idea of MOEA/D to realistic problems, e.g., flow-shop scheduling problem, and each subpopulation contains multiple individuals. MOEA/D decomposes a multiobjective optimization problem into a number of scalar optimization subproblems, and the offspring individual to one subproblem can be generated upon the parent individuals from its neighboring subproblems. This neighborhood relation can also be treated as a ring-shaped network. Unlike MPGA discussed in this paper, the subpopulations of MOEA/D are heterogeneous, representing different subproblems, and cannot evolve independently to get an optimal solution. Additionally, other networks such as 4D hypercubes [13] and a  $4 \times 4$  toroidal mesh are also utilized [4].

When it comes to scheduling problems [14, 15], MPGAs are widely used to get the optimal solution. Kimms et al. [16] introduced an MPGA as a procedure to solve the synchronized and integrated two-level lot sizing and scheduling problem; in the migration process, a copy of each best individual is inserted into the next population replacing a random individual. Therefore, the interaction of subpopulations can be abstracted as a ring-shaped network. Zandieh and Karimi [17] proposed an MPGA to search the Pareto optimal solution for a multiobjective group scheduling problem in hybrid flexible flow-shop with sequence-dependent setup times, where the interaction of subpopulations can be abstracted as fully connected networks.

We specifically choose the FJSP [18–21] as a measurement to test the performance of the MPGA as interaction structures of subpopulations vary. Research on the FJSP is essential for enterprises, especially for small-medium enterprises, to carry out production planning and scheduling in order to meet the delivery dates under a complex market [22]. For instance, Zhang et al. [23] put forward the MPGA based on the multiobjective scheduling of flexible job-shop, in which the abstracted topology of migration is a centralized network.

As summarized in Table 1, complex networks, such as scale-free networks and small-world networks, have been employed in cellular evolutionary algorithms to control interaction behaviors between individuals. However, for MPGAs, the focus of interaction structures among subpopulations is mainly rings, tori, hypercubes, and so on, and

complex networks have rarely been utilized to represent subpopulations and their interaction purposefully.

Therefore, in our previous studies [24, 25], we addressed how seven different network structures, including the ring-shaped network and the small-world network, influence the propagation rate of advantageous genes and thus affect the performance of MPGA for solving the FJSP. However, only the scale of networks is discussed, and the influence of other structural parameters is not researched. Thus, in this paper, we mainly take advantage of ring-shaped networks and small-world networks [26] to study the structural influence on the performance of MPGAs to solve the FJSP because ring-shaped networks are elementary topology widely used in MPGAs intentionally or unintentionally, and small-world networks with a definite node number will possess the same edge number as the rewiring possibility of edges varies, which would make them convenient and easy to study the influence of interaction structures as other conditions remain unchanged.

Concretely, we change the inherent structural parameter of the corresponding network constantly and record the simulation results of MPGAs with the network. The success rate (SR) and average optimal value (AOV) are utilized to measure the results of MPGAs. Besides, the Hamming distance index (HDI) is introduced to evaluate the difference between elite individuals and characterize the propagation rate of advantageous genes, which provide an insight into how different interaction structures of subpopulations affect the performance of MPGAs.

The sections below are organized as follows. In Section 2, basic knowledge of complex networks and MPGA with networks for solving the FJSP, as well as the evaluation index, are introduced. Then, experiments and results analysis are reported in Section 3. Conclusions follow in Section 4.

## 2. Preliminaries

*2.1. Network Models.* A network consists of some nodes connected by some edges with a certain topology (structure) [27]. Some basic concepts utilized to characterize the network are introduced in the following. On top of that, typical models of the ring-shaped network and small-world network are elaborated.

*2.1.1. Degree and Average Path Length.* The concept of degree is the most fundamental character and measure of a node in a network (in this paper, networks are undirected networks), and the degree of a node is defined to be the number  $k_i$  of its existing edges. The average degree of a network is the average value of all such node degrees  $k_i$  over the entire network, denoted by  $\bar{k}$ .

The average distance or average path length (APL) of a network is defined to be the average value of all distances over the network:

$$\text{APL} = \frac{2}{N \times (N - 1)} \sum_{i < j} d_{ij}, \quad (1)$$

TABLE 1: Summary of background studies.

References	Networks or topologies	Denotation of nodes
[4]	Toroidal mesh networks; hypercubes	Subpopulations
[5]	Fully connected networks; rings; tori	Subpopulations
[7]	BA networks	Individuals
[8]	Scale-free networks; small-world networks	Individuals
[9, 10]	Lattices	Individuals
[11, 12]	Rings	Subpopulations
[13]	4D hypercubes	Subpopulations
[16]	Ring-shaped networks	Subpopulations
[17]	Fully connected networks	Subpopulations
[23]	Centralized networks	Subpopulations

where  $N$  is the total number of nodes in the network and  $d_{ij}$  is the distance between node  $i$  and node  $j$ , i.e., the total number of edges connecting them through the shortest route [27].

**2.1.2. Ring-Shaped Networks.** A network in which every node has the same degree, i.e., the same number of connecting edges, is called a regular network. A typical sparse regular network is a nearest-neighbor coupled network, where every node is connected to  $2K$ -nearest neighbors,  $K$  nodes on each side. Particularly, as shown in Figure 1, such a network with a periodic boundary connectivity condition is a ring-shaped network [27].

**2.1.3. Small-World Networks.** The so-called WS small-world networks [28] were first proposed by Watts and Strogatz in 1998, which have both features of large clustering coefficients and short average path lengths. This model is generated by rewiring the edges of a ring-shaped network one by one with probability  $p$ , in which the case of  $p = 0$  corresponds to a regular network, and  $p = 1$  corresponds to a kind of ER random-graph networks.

A WS small-world network can be generated by the following algorithm:

- (1) Start from a ring-shaped network with  $N$  nodes, where each node is connected to its  $2K$  ( $K > 0$ ) neighbors,  $K$  nodes on each side
- (2) For every pair of connected nodes in the ring-shaped network, reconnect the edge with possibility  $p$  in such a way that the beginning end of the edge is kept, but the other end is disconnected, and then rewire it to a node randomly

This operation is conducted edge by edge on the original ring-shaped network, once and once only, either clockwise or counterclockwise. Additionally, self-loop and multiple edges are avoided. Figure 2 shows an example of WS small-world networks. Since the WS small-world network is the original and most widely used topology, the term “small-world network” mentioned in this paper is always the WS small-world network.

Additionally, compared to WS small-world networks, the algorithm can be modified by replacing “random rewiring edges” with “random adding edges,” resulting in

the NW small-world network model [29]. However, the edge number of NW small-world networks is changed with the rewiring process, so we discard the use of NW small-world networks.

**2.2. Flexible Job-Shop Scheduling Problem.** FJSP, which is also introduced in [24, 30], can be divided into two sub-problems: machine subproblem and operation subproblem, namely, selecting a specific machine for each operation and arranging a proper processing order of all operations. Usually, there exist a job set within  $n$  jobs, labeled as  $J = \{J_i\}_{i=1}^n$ , and a machine set within  $m$  machines, labeled as  $M = \{M_k\}_{k=1}^m$ . For each job  $J_i$ , there is an operation set within  $l_i$  operations, labeled as  $O_i = \{O_{ij}\}_{j=1}^{l_i}$ . Each operation  $O_{ij}$  can only be processed on a specific group of machines, labeled as  $S_{ij} \subseteq M$ . The goal of the scheduling problem is to assign operations with proper orders to proper machines at the proper time, to pursue some given objectives such as minimizing maximum machine workload and minimizing maximum makespan. For simplicity without losing generality, we choose to minimize makespan as the objective. The mathematical model is as follows [25]:

$$\min F_{\max} = \max(F_{ij}), \quad \forall i, j, \quad (2)$$

s.t.

$$F_{ijk} \geq 0, \quad \forall i, j, k, \quad (3)$$

$$P_{ijk} \geq 0, \quad \forall i, j, k, \quad (4)$$

$$B_{ijk} \geq 0, \quad \forall i, j, k, \quad (5)$$

$$X_{ijk} \in \{0, 1\}, \quad \forall i, j, k, \quad (6)$$

$$i, j, l, i', j' \in \{1, 2, 3, \dots\}, \quad (7)$$

$$\sum_{k \in S_{ij}} X_{ijk} = 1 \wedge F_{ijk} - B_{ijk} = P_{ijk}, \quad \forall i, j, \quad (8)$$

$$F_{ij} - F_{i(j-1)} \geq P_{ijk} \cdot X_{ijk}, \quad \forall i, j, k, \quad (9)$$

$$F_{i'j'k} \leq B_{ijk} \vee F_{ijk} \leq B_{i'j'k}, \quad \forall i', j' \neq i, j. \quad (10)$$

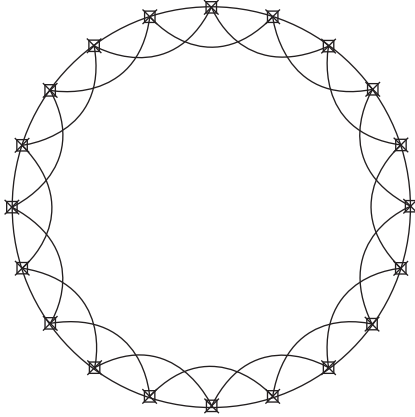


FIGURE 1: Ring-shaped networks.

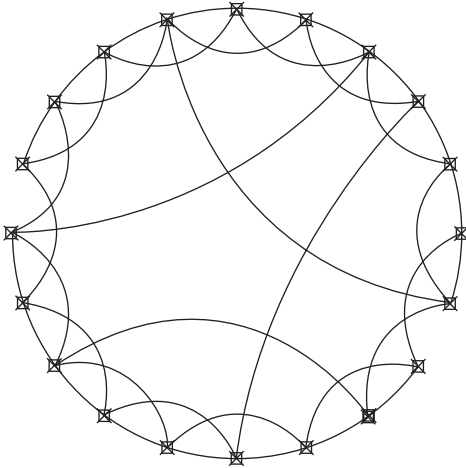


FIGURE 2: WS small-world networks.

Table 2 lists the notations used in the FJSP. For equation (6), as  $O_{ij}$  is arranged to be processed on  $M_k$ ,  $X_{ijk} = 1$ ; otherwise, 0. Equation (7) denotes the domains of variables. Equation (8) guarantees that each operation can only be processed at one machine without disruption, and ‘ $\wedge$ ’ represents logical AND. Therefore,  $F_{ij} = F_{ijk}, \forall i, j, \exists k, k \in S_{ij}$ . Equation (9) ensures each job will be processed in the correct order. Equation (10) ensures that one machine can only process one operation at the same time, and ‘ $\vee$ ’ represents logical OR.

Besides, Table 3 lists a four-step process of the FJSP for 2 jobs with a total of 4 operations ( $4 \times 4$ ), where ‘—’ means the corresponding operation cannot be processed in that machine. The fourth number “19” in the first line, for example, represents that  $O_{11}$  can be processed in  $M_4$  within time horizon 19.

**2.3. MPGA with Networks for Solving the FJSP.** For using an MPGA with networks to solve the FJSP, multiple operations are included, such as encoding, decoding, crossing, mutation, and migration.

**2.3.1. Encoding.** In this algorithm, we employ an integer encoding method [31] to generate individuals, each of which can represent a feasible solution of the FJSP. The process of encoding can be divided into two stages, to wit, encoding for machine subproblem and operation subproblem. For the former, an encoded solution is denoted by a string of integers whose length is equal to the total operation number of all jobs, in which an integer in each position represents an operation, and the value represents the ordinal number of machines in the candidate machine set ( $S_{ij}$ ) of this operation. For the FJSP mentioned in Table 3, suppose there is an encoded solution (1 2 3 1) within four integers which reflect that there are a total of four operations in the two jobs. The first integer 1 means that  $O_{11}$  is arranged with  $M_1$ . For the second integer, because the candidate machine set of  $O_{12}$  is  $S_{12} = \{M_2, M_4\}$ , integer 2 denotes that  $O_{12}$  will be processed on the fourth machine, namely,  $M_4$ , rather than  $M_2$ . Similarly,  $O_{21}$  is arranged to be processed on  $M_4$  and  $O_{22}$  on  $M_3$ .

For the latter one, a string of integer represents an encoded solution, and the number of integers equals the total operation number of all jobs too. The value of an integer denotes the serial number of jobs, and if a job has  $l_i$  operations, the job number will appear  $l_i$  times. The sequence of integers represents the processing order of the corresponding jobs. For example, there is an encoded solution (2 1 1 2) for the FJSP as shown in Table 3, in which integers “1” and “2” appear twice, respectively, due to both  $J_1$  and  $J_2$  including two operations. In detail, the fourth integer is “2,” and it is the second occurrence of “2” in this string, indicating the processing of the second operation of the second job, namely,  $O_{22}$ . Therefore, the processing order of operations indicated by this solution is  $O_{21}$ ,  $O_{11}$ ,  $O_{12}$ , and  $O_{22}$ .

Therefore, we combine (1 2 3 1) and (2 1 1 2) together, getting an individual (1 2 3 1 2 1 1 2), which means  $O_{21}$  will be processed on  $M_4$ ,  $O_{11}$  on  $M_1$ ,  $O_{12}$  on  $M_4$ , and  $O_{22}$  on  $M_3$  sequentially.

**2.3.2. Decoding.** An encoded individual must be decoded into an original solution of the FJSP to calculate the fitness. The decoding algorithm proposed by Shi et al. [30] was adopted in this paper as follows:

At first, we convert an individual into a matrix, labeled as  $H_d$ . For instance, [3 1 2 1 2 1 1 2], a given individual for the FJSP in Table 3, can be converted into a matrix as shown in Figure 3.

In matrix  $H_d$ , the first column stores the sequence numbers of jobs; the second column stores the sequence numbers of operations of the corresponding job; the third column stores the sequence numbers of machines in  $S_{ij}$ ; and the fourth column stores the time horizon as the corresponding operation processed in the specific machine. Besides, the last two columns are reserved to store the start time and completion time, respectively. For instance, the first row of  $H_d$  represents that  $O_{21}$  is processed on  $M_3$  within time horizon 19.

TABLE 2: Notations related to the FJSP.

Symbol	Meaning
$n$	The number of jobs
$J$	The job set
$J_i$	The $i$ th job
$m$	The number of machines
$M$	The machine set
$M_k$	The $k$ th machine
$l_i$	The number of operations of the $i$ th job
$O_i$	The operation set of the $i$ th job
$O_{ij}$	The $j$ th operation of the $i$ th job
$S_{ij}$	The candidate machine set $O_{ij}$ can be processed on
$F_{\max}$	Makespan
$F_{ij}$	Completion time of $O_{ij}$
$F_{ijk}$	Finish time of $O_{ij}$ on $M_k$
$P_{ijk}$	Processing time of $O_{ij}$ on $M_k$
$B_{ijk}$	Start time of $O_{ij}$ on $M_k$
$X_{ijk}$	Indicate if $O_{ij}$ is processed on $M_k$

TABLE 3: One instance of the FJSP.

Jobs	Operations	Machines			
		$M_1$	$M_2$	$M_3$	$M_4$
$J_1$	$O_{11}$	24	—	16	19
	$O_{12}$	—	21	—	13
$J_2$	$O_{21}$	9	—	6	8
	$O_{22}$	—	—	8	6

Next, we update the start time and completion time row by row. If  $O_{ij}$  represented by the corresponding row is the first operation of  $J_i$  and  $M_k$  selected to process  $O_{ij}$  has not processed any other operation, then  $B_{ijk} = 0$ , and  $F_{ijk} = B_{ijk} + P_{ijk}$ . Assign  $B_{ijk}$  and  $F_{ijk}$  into columns 5 and 6 of  $H_d$ .

If  $O_{ij}$  is the first operation of  $J_i$  and  $M_k$  has processed other operations, then find all idle-time intervals of  $M_k$  denoted by  $[S_q E_q]$  ( $q = 1, 2, \dots$ ), check all idle intervals one by one to find the first one whose duration is longer than  $P_{ijk}$ , and set  $B_{ijk} = S_q$  and  $F_{ijk} = B_{ijk} + P_{ijk}$ .

If  $O_{ij}$  is not the first operation of  $J_i$ , the former operation  $O_{i(j-1)}$  is finished on  $M_{k'}$ , and  $M_k$  has not been assigned to any operation, then  $B_{ijk} = F_{i(j-1)k'}$  and  $F_{ijk} = B_{ijk} + P_{ijk}$ .

If  $O_{ij}$  is not the first operation of  $J_i$  and  $M_k$  has been assigned to other operations, then search for and check all idle-time intervals of  $M_k$  from left to right on the timeline. If  $E_q - S_q \geq P_{ijk}$  and  $F_{i(j-1)k'} \leq S_q$ , then  $B_{ijk} = S_q$  and  $F_{ijk} = B_{ijk} + P_{ijk}$ ; if

$E_q - S_q \geq P_{ijk}$ ,  $F_{i(j-1)k'} \geq S_q$ , and  $E_q - F_{i(j-1)k'} \geq P_{ijk}$ , then  $B_{ijk} = F_{i(j-1)k'}$  and  $F_{ijk} = B_{ijk} + P_{ijk}$ . Since  $E_q$  of the last interval is positive infinity, an idle-time interval can always be found to satisfy one of these conditions. In this way, there is no earlier idle-time interval on  $M_k$  for  $J_i$  to insert without delaying or interrupting other operations, indicating that an active schedule is achieved [32]. The optimal schedule for any regular measure of performance, including makespan, will be a

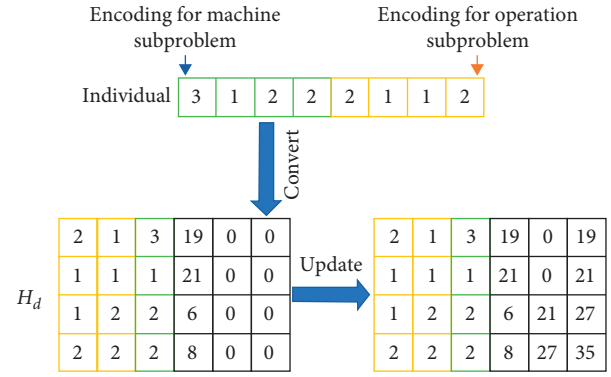


FIGURE 3: Illustration of decoding.

member of the active schedule set [33]. Therefore, the optimal value of makespan can be found by MPGAs with this decoding method.

At last, after updating columns 5 and 6 of all rows, the maximum value of column 6 can be found as the makespan of the FJSP, and our goal is to minimize makespan through MPGAs. As shown in Figure 3, the makespan of our example is 35.

**2.3.3. Crossing.** The processing of crossing is also divided into two stages. For the machine subproblem, we implement the standard two-point crossing process [30]. In this process, we choose two individuals randomly, called parent 1 and parent 2, both of which will be divided into three parts with the same size at random common breakpoints.

We partially select parts from parent 1 and parent 2 to shape offspring 1, and remaining parts compose offspring 2, which are indicated in Figure 4.

For the code of the operation subproblem [34], all jobs are randomly divided into two groups: group 1 and group 2. Offspring 1 inherits the integers of parent 1 belonging to group 1 and those of parent 2 belonging to group 2; in the same method, offspring 2 inherits the integers of parent 1 belonging to group 2 and those of parent 2 belonging to group 1, respectively, meanwhile preserving the sequence of these integers. For example, as indicated in Figure 5, jobs 1 and 2 belong to group 1, and jobs 3 and 4 belong to group 2. To get offspring 1, we first preserve the integers belonging to group 1, i.e., job 1 and job 3. Next, we insert integers of group 2, i.e., job 2 and job 4, from parent 2 into parent 1. Meanwhile, the original sequence of integers representing jobs 2 and 4 is also preserved. In the same way, we get offspring 2.

**2.3.4. Mutation.** For MPGAs, mutation can be a supplementary strategy to maintain diversity [35]. Based on the characteristic of integer encoding, it is divided into two stages. In the process of the machine subproblem, we randomly choose several individuals based on the mutation probability. Then, in several random positions of these selected individuals, the original integers can be replaced by alternative integers, which should be smaller than the total

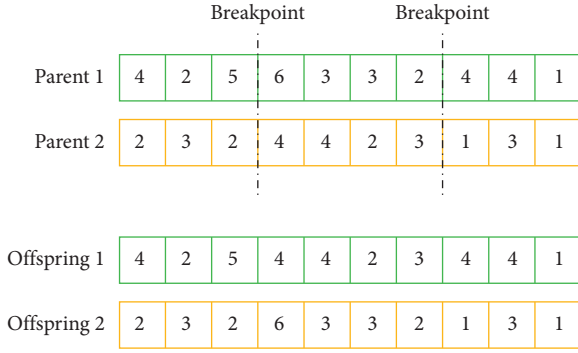


FIGURE 4: Crossing of the machine subproblem.

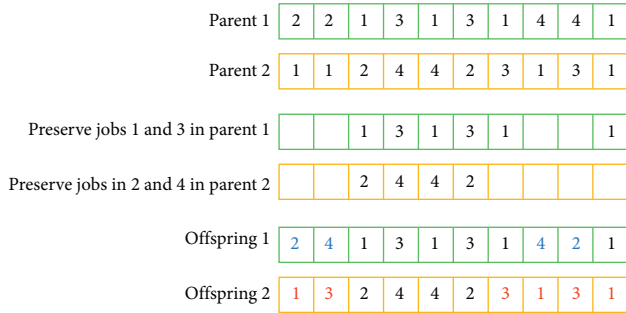


FIGURE 5: Crossing of the operation subproblem.

number of the corresponding candidate machines. For the operation subproblem, in the same way, some individuals are randomly selected. Within each selected individual, several pairs of integers are randomly selected, and the positions of two integers, within each pair, are swapped.

**2.3.5. Migration.** The migration strategy is often utilized to mitigate the premature convergence of evolutionary algorithms [36]. To obtain an MPGA with networks, we divide the population of standard genetic algorithms into subpopulations, denoted by nodes. Communications between nodes, denoted by edges, occur when certain individuals in one node migrate to another periodically based on different networks. Figure 6 displays an MPGA with a certain network.

In the process of migration, we select a subpopulation randomly in an MPGA, denoted by one node of the network, and find all neighbor nodes (subpopulations). Next, we find the best elite of these subpopulations which will replace a random individual in each of these selected subpopulations.

**2.3.6. Implementation of the MPGA with Networks to Solve the FJSP.** As shown in Algorithm 1, we initialize  $N_s$  subpopulations, and each of them can be run parallel [37]. Within each subpopulation, there are  $S$  individuals, namely,  $N_s \times S$  individuals, in total, and we first decode the individuals to get fitness, i.e., makespan. Then, tournament selection strategy is employed to generate the next subpopulation and maintain the subpopulation size unchanged. After that, crossing and mutation operators are

implemented to generate new individuals. After all of the subpopulations are updated, we calculate the fitness again and select the elites of each subpopulation into the elite set. At last, our proposed migration strategy is utilized to propagate advantageous genes. As iterating to the maximum generation, the best fitness and individual are output.

**2.4. Evaluation Indexes.** An index to evaluate the propagation rate of advantageous genes among subpopulations under different network structures is needed. All the best individuals of each subpopulation are selected into an elite set. With the accumulation of advantageous genes through the operators of MPGA, the difference between individuals becomes small. Therefore, if the propagation rate of advantageous genes is larger, the difference between these elites will be smaller. Thus, we introduce the HDI to evaluate the difference between elite individuals. The HDI is calculated as follows [24]:

$$\text{HDI} = \sum_{i=1}^{100} \sum_{j=1}^{2 \times J_T} \frac{(1 - \delta(h_{ij}^1, h_{ij}^2))}{(100 \times 2 \times J_T)}, \quad (11)$$

where we only sample 100 pairs of individuals to limit time consumption and  $\delta(\dots)$  is the Kronecker function. As the two independent variables are unequal, the value of  $\delta(\dots)$  is 0; otherwise, 1.  $J_T$  represents the total operation number of all jobs, and  $h_{ij}^1$  and  $h_{ij}^2$  denote the  $i$ th pair's two  $j$ th integers in this sample. Therefore, smaller HDI indicates less difference between elites, suggesting the propagation among subpopulations is faster.

Besides, we use the SR to assess the performance of MPGA, better performance with higher SR. This measure is defined as follows:

$$\text{SR} = \frac{N_s}{N_t}, \quad (12)$$

where  $N_s$  denotes the number of times the algorithm finds the optimal value and  $N_t$  denotes the total number of times it runs. The optimal value, mentioned in this paper, is defined as the best value found by all current algorithms by now, rather than the best value found by the currently running algorithm.

Also, due to the stochastic nature of MPGA, we have to run the proposed algorithm repeatedly, and statistical features are often utilized to analyze simulation results [38–40]. Therefore, the AOV is introduced and calculated as follows:

$$\text{AOV} = \frac{1}{N_t} \sum_i^{N_t} \text{OV}_i, \quad (13)$$

in which  $\text{OV}_i$  is the optimal value obtained by the  $i$ th run of the algorithm. The smaller the AOV is, the better the performance will be.

### 3. Experiments and Results' Analysis

In this section, we design experiments to evaluate how the parameters of interaction structures of subpopulations, i.e.,

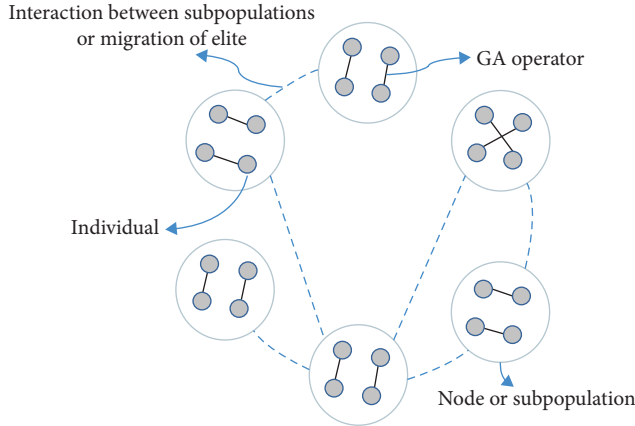


FIGURE 6: Schematic diagram of the MPGA with networks.

ring-shaped networks and small-world networks, affect the performance of MPGAs.

We take two benchmark problems proposed by Kacem et al. [41] as examples. One possesses 8 jobs and 8 machines, labeled as P1; the other possesses 10 jobs and 10 machines, labeled as P2. For the basic parameters of MPGA, we set the number of subpopulations  $N_s$  as 100, the size of a subpopulation  $S$  as 40, the mutation possibility  $P_m$  as 0.08, and the total number of iterations  $iter_{max}$  as 400.

**3.1. MPGAs with Ring-Shaped Networks.** For ring-shaped networks, the parameter  $K$  is changed constantly. The value of  $K$  slides from 1 to 24, and the interval is 1. Due to the stochastic nature of evolutionary algorithms, for each value of  $K$ , the MPGA with a certain network is run 50 times.

Figure 7 illustrates the curves of the HDI over iterations for different  $K$ . The  $X$ -axis represents the values of the HDI, and the  $Y$ -axis represents the times of iteration. As shown in the following, the HDI decreases faster as  $K$  becomes larger, which indicates the propagation rate of advantageous genes becomes larger as  $K$  increases.

To further reveal the relationship between the propagation rate and the structural parameter of networks, Figure 8 shows the curves of the HDI at 100th iteration, 200th iteration, and 300th iteration along with APL for different  $K$ , in which the  $X$ -axis represents the value of  $K$ , and the  $Y$ -axis (left) and  $Y$ -axis (right) represent the value of HDI and APL, respectively. It can be seen that, as  $K$  becomes larger, the HDI at the same iteration decreases in accordance with the trend of APL shown by cyan lines, namely, the smaller the APL is, the faster the propagation of advantageous genes will be. Also, the variation of parameter  $K$  can change the APL to affect the propagation rate of advantageous genes. In this way, the performance of MPGAs can be influenced by different values of  $K$ .

To directly show the influence of variation of  $K$  on the performance of MPGAs, curves of SR and AOV over  $K$  are shown in Figure 9. For both benchmark problems, when  $K$  increases from 1 to 25, the performance of MPGAs increases rapidly at first and then decreases with fluctuation. Concretely, in Figure 9(a), the SR of P1 begins with 24% and

increases to the peak, i.e., 90%, when  $K = 4$ . Then, SR decreases slowly with fluctuation. In contrast, the AOV of P1 begins with 14.9 and decreases to 14.12. Then, AOV rises slowly with fluctuation. For P2, shown in Figure 9(b), the trends of SR and AOV are similar to those in P1. The peak of the SR or the lowest value of AOV can also be found at  $K = 4$ . For the worst performance of P2, the lowest SR is 4% at  $K = 24$ , and the highest AOV is 8.42 at  $K = 23$ .

Therefore, the variation of  $K$  can influence the propagation rate of advantageous genes, consequently affecting the performance of algorithms. In detail, as  $K$  is very small, the APL is large, and the advantageous genes can hardly propagate to distant subpopulation in the network. Each subpopulation evolves with little communication and cannot benefit from elites of other subpopulations. However, as  $K$  is too large, the APL is very small, and the advantageous genes can be propagated rapidly. Therefore, MPGA is more likely to fall into local optimal, leading to premature convergence. For our benchmark problems, the moderate value of  $K$  is 4, so the corresponding propagation rate is neither too large nor too small, and the MPGA with networks can get the best performance.

Additionally, because of the stochastic nature of metaheuristics, to show significant differences between MPGAs with different structural parameters  $K$  of the network, the nonparametric Wilcoxon signed-rank test of the 50 independent runs is conducted [42]. The null hypothesis, termed as  $H_0$ , is set as "there exists no difference between MPGAs with different structural parameters of networks." Accordingly, the alternative hypothesis  $H_1$  is "the MPGAs with two different values of the structural parameter are statistically different." A significance level 0.05 is employed, and a  $p$  value less than 0.05 indicates that there exist significant differences between the two samples [43]. As shown in Table 4, we take  $K = 4$  vs. other values of  $K$  as examples, and the row titled with "total" indicates how many times  $h = 1$  within our two benchmark problems. Basically, the larger the difference between two values of  $K$  is, more likely  $h = 1$ , namely, there exist differences between MPGAs with different  $K$ , especially, as the difference of  $K$  is large. The results further indicate parameter  $K$  can influence the performance of MPGAs.

**3.2. MPGAs with Small-World Networks.** For small-world networks, the structural parameters are  $K$  and  $p$  as mentioned in Section 2.1.3. Since the number of subpopulations (100) is very small, causing the scale of the corresponding network to be small, the rewiring possibility  $p$  can bring significant uncertainty, namely, the number of rewired edges fluctuates with the same  $p$ . Thus, we directly change the number of rewired edges  $P$  as the structural parameter to implement simulation. We suppose the value of  $P$  slides from 0 to 120 and the interval becomes larger as  $P$  increases. When  $P = 0$ , the corresponding network is a ring-shaped network. It is noted that, with the same  $P$  ( $P \neq 0$ ), there still exists uncertainty because different edges can be selected to rewire to different locations. To limit this, we randomly generate three different small-world networks with the same

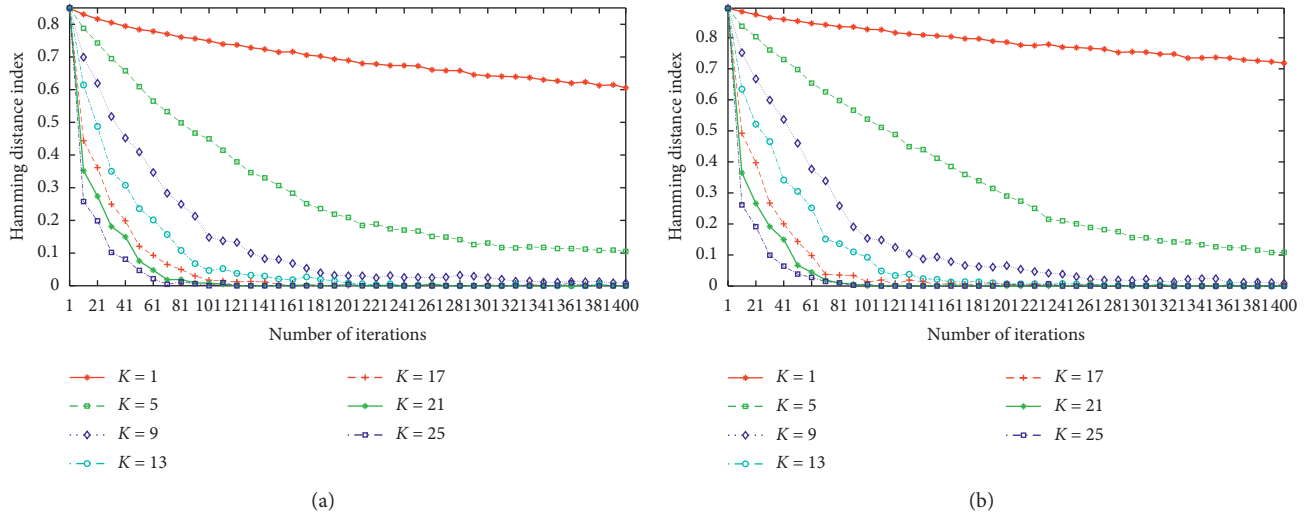


FIGURE 7: HDI over iterations with ring-shaped networks. (a) P1. (b) P2.

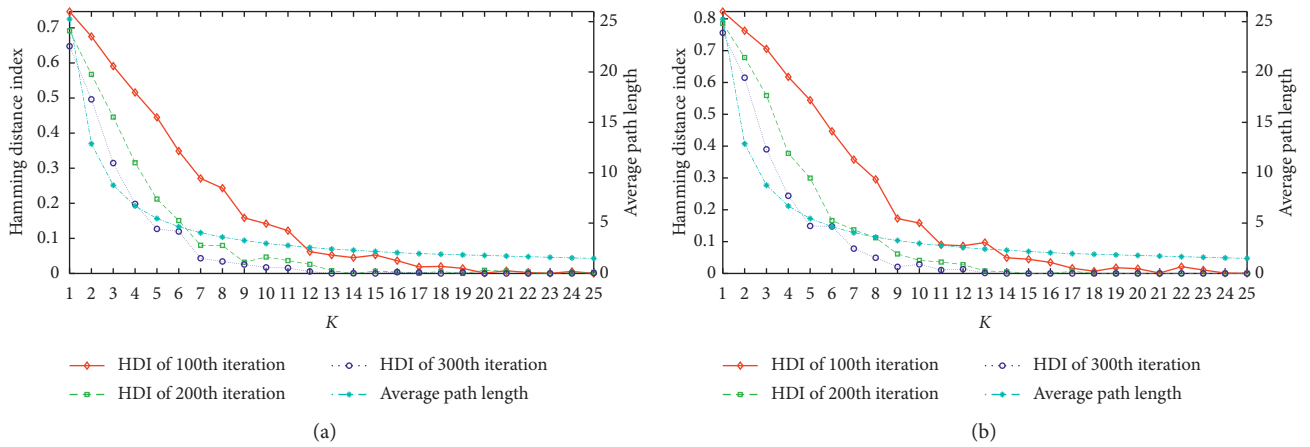


FIGURE 8: HDI over  $K$  with ring-shaped networks. (a) P1. (b) P2.

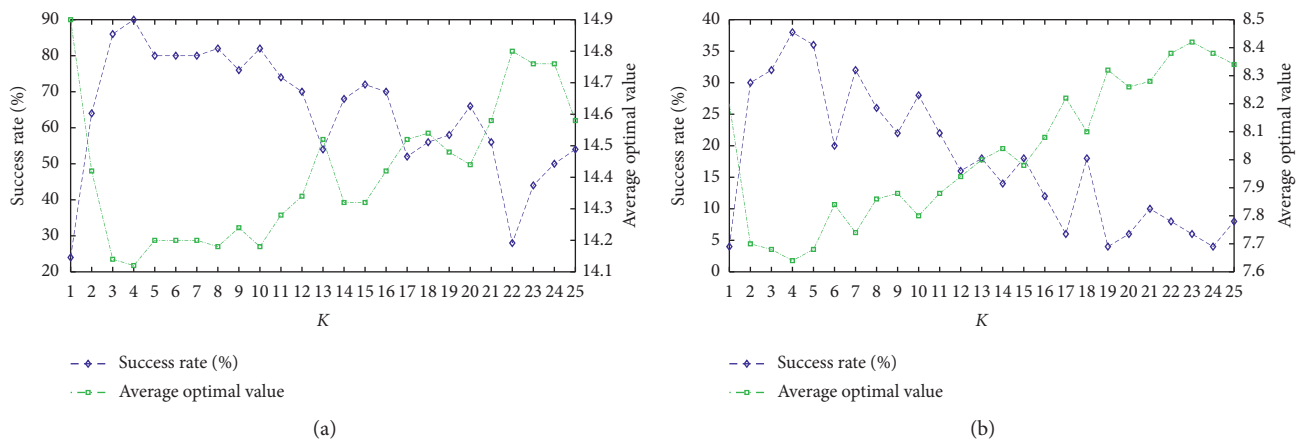


FIGURE 9: SR and AOV over  $K$  with ring-shaped networks. (a) P1. (b) P2.



```

Input:
  FJSP: the FJSP to solve
   $N_s$ : the number of subpopulations;
   $S$ : the size of each subpopulation;
   $P_m$ : mutation probability;
   $iter_{max}$ : the maximum generation to run;
Output:
   $R^*$ : the best fitness or makespan;
   $g^*$ : the best individual;
 $G = \{G^1, G^2, \dots, G^M\} \leftarrow \text{encoding}(\text{FJSP}, N_s, S)$ ;
While (generation  $\leq iter_{max}$ ) do
  For  $i_{sub} = 1: N_s$  do
     $R \leftarrow \text{decoding}(G^{i_{sub}})$ ;
     $G_s^{i_{sub}} \leftarrow \text{selection}(G^{i_{sub}})$  based on  $R$ ;
     $G_c^{i_{sub}} \leftarrow \text{crossing}(G_s^{i_{sub}})$ ;
     $G^{i_{sub}} \leftarrow \text{mutation}(G_c^{i_{sub}}, P_m)$ ;
  End for
   $R \leftarrow \text{decoding}(G)$ ;
  Select the elites of each subpopulation to compose elite set  $E$ ;
   $G \leftarrow \text{migration}(G, E)$ ;
End while
Find the best individual  $g^*$  from  $E$  and the corresponding fitness  $R^*$ .

```

ALGORITHM 1: MPGA with networks.

$P$ , and MPGA is run 20 times with each network, namely, a total of 60 times for each value of  $P$ . At last, for P1, the value of  $K$  is set as 2. For P2, the value of  $K$  is set as 4.

Figure 10 illustrates curves of the HDI over iterations for different  $P$ . It can be noted that the HDI decreases faster as  $P$  increases, indicating the propagation rate of advantageous genes becomes larger with the increase of  $P$ .

To intuitively show the influence of variation of  $P$  on the performance of MPGAs, curves of the HDI at 100th, 200th, and 300th iterations over  $P$  are shown in Figure 11, along with APL. The  $X$ -axis represents the value of  $P$ , and the  $Y$ -axis (left) and  $Y$ -axis (right) represent the value of HDI and APL, respectively. As shown by red, green, and blue lines, with the increase of  $P$ , HDI of the same iteration decreases in accordance with the trend of APL shown by cyan lines. It can be concluded that the variation of  $P$  can change the APL, thus affecting the propagation rate of advantageous genes, by which the performance of MPGAs is influenced. Compared with the ring-shaped network in Figure 8, the decrease of APL caused by the increase of  $P$  is gentler than that caused by the increase of  $K$ , resulting in a more gradual change of HDI in Figure 11.

Figure 12 directly shows how the performance, measured by SR and AOV, is affected by  $P$ , in which the  $X$ -axis represents the value of  $P$ , and the  $Y$ -axis (left) and  $Y$ -axis (right) represent the value of SR and AOV, respectively. For P1, shown in Figure 12(a), the SR of P1 starting from 64.44% increases to the peak of 86.67% as  $P = 4$  and then decreases with fluctuation. Besides, the AOV starting from 14.41 decreases to the lowest value 14.12 as  $P = 5$  and then increases with fluctuation.

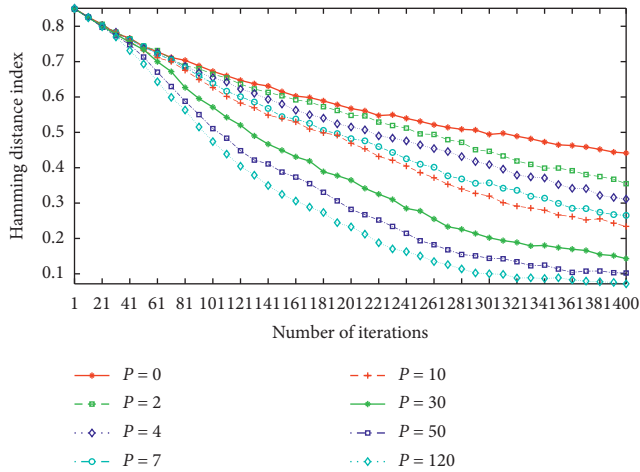
In Figure 12(b), because P2 is more sophisticated than P1, the success rate is already low. We choose to set  $K$  as 4, at which, as shown in Figure 9, MPGA with the ring-shaped network gets

the best performance, i.e., SR = 38% and AOV = 7.64. Therefore, the improvement of performance is limited and subject to fluctuation due to the stochastic nature of evolutionary algorithms. However, the best performance can be found at  $P = 6$  at which the peak of the SR is 48.33%, and the lowest value of AOV is 7.5. Besides, for the worst performance, the SR is 18.33%, and AOV is 7.86.

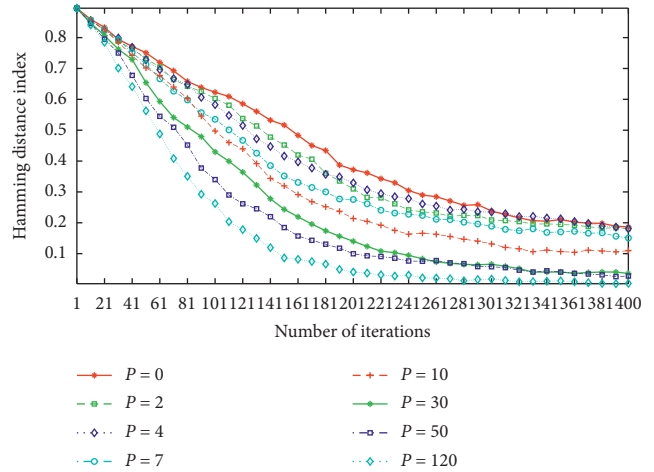
Therefore, a similar conclusion can be drawn that the variation of structural parameter  $P$  can influence the propagation rate of advantageous genes, thus affecting the performance of MPGAs with small-world networks. Besides, it is noteworthy that the edge number and node number are not changed as  $P$  varies, so the change in performance indicates that the MPGA can be improved by choosing a more proper interaction structure of subpopulations as other conditions remain unchanged.

Besides, the Wilcoxon signed-rank test of the 60 independent runs is implemented again. As shown in Table 5, we take  $P = 6$  vs. other values of  $P$  as examples. Although there are fewer times  $h = 1$ , the results still indicate that parameter  $P$  can influence the performance of MPGAs with small-world networks.

**3.3. Comparison between Ring-Shaped Networks and Small-World Networks.** We can conclude that both parameter  $K$  of ring-shaped networks and unique parameter  $P$  of small-world networks can influence the performance of MPGAs, but the degree of influence is different. This is because due to the variation of  $K$  and  $P$ , the APL is changed to a different extent. For small-world networks, as  $K = 2$  and  $P$  slides from 0 to 120, APL varies from 12.88 to 3.53; as  $K = 4$  and  $P$  slides from 0 to 120, APL varies from 7.00 to 2.57. However, for ring-shaped networks, as  $K$  slides from 1 to 25, APL varies

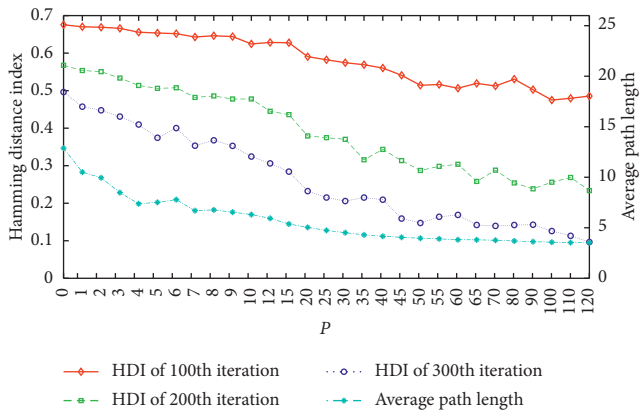


(a)

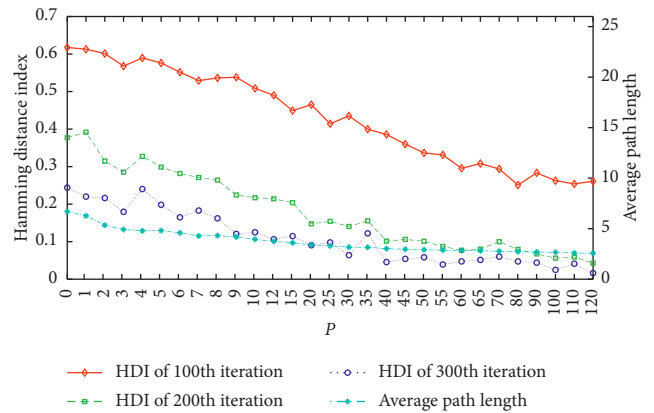


(b)

FIGURE 10: HDI over iterations with small-world networks. (a) P1. (b) P2.

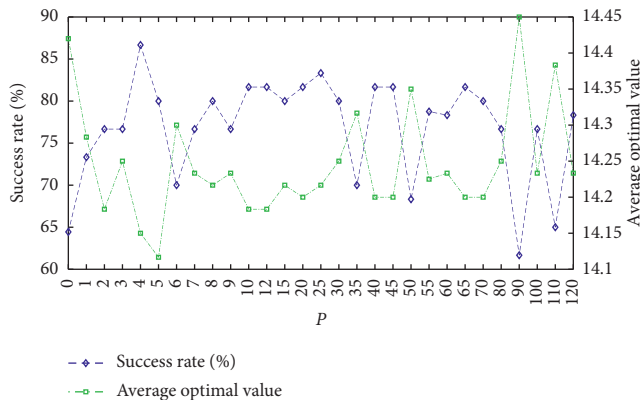


(a)

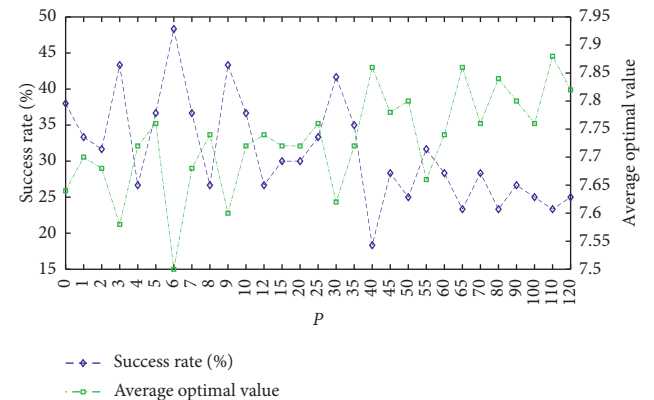


(b)

FIGURE 11: HDI over  $P$  with small-world networks. (a) P1. (b) P2.



(a)



(b)

FIGURE 12: SR and AOV over  $P$  with small-world networks. (a) P1. (b) P2.

TABLE 4: Wilcoxon signed-rank test results for ring-shaped networks.

Parameter pairs	$K = 4$	$K = 4$	$K = 4$	$K = 4$	$K = 4$	$K = 4$	$K = 4$	$K = 4$	$K = 4$	$K = 4$	
	vs. $K = 1$	vs. $K = 2$	vs. $K = 6$	vs. $K = 9$	vs. $K = 11$	vs. $K = 13$	vs. $K = 15$	vs. $K = 18$	vs. $K = 21$	vs. $K = 25$	
$P_1$	$p$ value	$1.2e-07$	0.0104	0.4850	0.2407	0.0920	$7.2e-04$	0.0540	0.0017	0.0011	$5.4e-04$
	$h$	1	1	0	0	0	1	0	1	1	1
$P_2$	$p$ value	$1.2e-05$	0.5316	0.0709	0.0339	0.0282	0.0051	0.0076	$3.3e-04$	$3.9e-05$	$1.5e-05$
	$h$	1	0	0	1	1	1	1	1	1	1
Total		2	1	0	1	1	2	1	2	2	2

TABLE 5: Wilcoxon signed-rank test results for small-world networks.

Parameter pairs	$P = 6$	$P = 6$	$P = 6$	$P = 6$	$P = 6$	$P = 6$	$P = 6$	$P = 6$	$P = 6$	$P = 6$	
	vs. $P = 1$	vs. $P = 2$	vs. $P = 3$	vs. $P = 4$	vs. $P = 5$	vs. $P = 8$	vs. $P = 12$	vs. $P = 20$	vs. $P = 55$	vs. $P = 90$	
$P_1$	$p$ value	0.8348	0.1444	0.5485	0.0389	0.0164	0.2513	0.1444	0.0495	0.3359	0.0762
	$h$	0	0	0	1	1	0	0	1	0	0
$P_2$	$p$ value	0.048	0.0771	0.4191	0.0133	0.0946	0.0106	0.0133	0.0285	0.041	0.0077
	$h$	1	0	0	1	0	1	1	1	1	1
Total		1	0	0	2	1	1	1	2	1	1

TABLE 6: Comparison with other algorithms.

Problems	Optimal value						
	AIA	HHS	M2	M4	MILP	HA	This article
SFJS01	66	66	66	66	66	66	66
SFJS02	107	107	107	107	107	107	107
SFJS03	221	221	221	221	221	221	221
SFJS04	355	355	355	355	355	355	355
SFJS05	119	119	119	119	119	119	119
SFJS06	320	320	320	320	320	320	320
SFJS07	397	397	397	397	397	397	397
SFJS08	253	253	253	253	253	253	253
SFJS09	210	210	210	210	210	210	210
SFJS10	516	516	516	516	516	516	516
MFJS01	468	468	468	468	468	468	468
MFJS02	448	446	446	466	446	446	446
MFJS03	468	466	466	466	446	466	466
MFJS04	554	554	564	590	554	554	554
MFJS05	527	514	514	546	514	514	514
MFJS06	635	634	634	666	634	634	634
MFJS07	879	879	928	1990	879	879	879
MFJS08	884	884	—	—	—	884	884

from 25.25 to 1.50. It can be seen that the decrease of APL is dramatic as  $K$  increases, bringing significant influence on the performance, while for  $P$  of small-world networks, the variation of  $P$  is gentle, causing the influence to be slighter. Therefore, to adjust the APL and the propagation rate of advantageous genes, we can first change parameter  $K$ , and the variation of  $P$  can be a supplementary method to adjust

APL in a smaller step. In this way, we can get a more proper interaction structure to improve the performance of MPGA.

3.4. Efficacy of the MPGA with Networks to Solve the FJSP. To further demonstrate the efficacy of MPGA with networks to solve the FJSP, our algorithm is employed to solve small-

size FJSPs (SFJ01–10), and larger-size FJSPs (MFJS01–08) [44]. Based on the above analysis, we set  $K = 4$  and  $P = 3$ . Other parameters for MPGA are the same as above. Previous studies, such as AIA [45], HHS [46], M2 and M4 [47], MILP [48], and HA [49], are also experimented on the aforementioned instances, and their results are cited to compare with our algorithm.

As shown in Table 6, regarding MFJS02 and MFJS03, the optimal value of our algorithm is 446 and 466 (better than 448 and 468 found by AIA). Regarding MFJS04, the optimal value is 554 (better than 564 found by M2 and 590 found by M4). Regarding MFJS05, the optimal value is 514 (better than 527 and 546 found by AIA and M4, respectively). Regarding MFJS06, the optimal value is 634 (better than 635 and 666 found by AIA and M4, respectively). Regarding MFJS07, the optimal value is 879 (better than M2 and M4). At last, for MFJS08, the optimal value is 884, which is not given by M2, M4, and MILP.

#### 4. Conclusion and Future Scope

For further analyzing the influence of interaction structures on the performance of MPGAs, MPGA based on complex networks, such as ring-shaped networks and small-world networks, has been presented in this work. Then, how structural parameters  $K$  and  $P$  of these two types of networks affect the performance of MPGAs is discussed by solving the FJSP.

First, the HDI is utilized to quantitatively measure the differences of elites between subpopulations. The smaller the HDI is, the higher the propagation rate of advantageous genes will be. The simulation results indicate that the HDI has a positive correlation with the APL. This means the smaller the APL is, the higher the propagation rate is.

Next, as the structural parameters  $K$  and  $P$  increase, the curves of the HDI over iterations decrease faster, namely, the propagation rate becomes faster with the increases of  $K$  and  $P$ , resulting in the variation of the performance of MPGAs captured by SR and AOV. For instance, the highest and lowest SR of P2 are 48.33% and 4.00%. Since the propagation rate of advantageous genes can neither be too high nor too low, we should choose proper  $K$  and  $P$  for MPGAs with networks. Based on experiments, the proper intervals for  $K$  can be [2, 5], and the choice of  $P$  is dependent on the value of  $K$  because the variation of  $K$  can bring a significant change of APL, and the variation of  $P$  can be a supplementary method to adjust APL in a smaller step.

At last, the edge number and node number of small-world networks are not changed as  $P$  varies, so the change in performance indicates that MPGA can be improved by choosing a more proper interaction structure of subpopulations as other conditions remain unchanged.

Because of the limitation of computing capability, the number of nodes is not big enough compared with classic network science, but this study can still illustrate the importance of interaction structures and the meaning of introducing complex networks as a study tool. In the future,

more complex or compound networks, such as multilayer networks, can be introduced into the research.

#### Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

#### Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

#### Acknowledgments

This research was supported by the National Natural Science Foundation of China (Grant no. 51805059) and the National Green Manufacturing System Integration Support Project [2017] 327.

#### References

- [1] D. Whitley, "A genetic algorithm tutorial," *Statistics and Computing*, vol. 4, no. 2, 1994.
- [2] X.-L. Zheng and L. Wang, "A multi-agent optimization algorithm for resource constrained project scheduling problem," *Expert Systems with Applications*, vol. 42, no. 15-16, p. 6039, 2015.
- [3] Y. Li and X. Zeng, "Multi-population co-genetic algorithm with double chain-like agents structure for parallel global numerical optimization," *Applied Intelligence*, vol. 32, no. 3, pp. 292–310, 2010.
- [4] E. Cantú-Paz, "A survey of parallel genetic algorithms," *Calculation of Paralleles Reseaux System Report*, vol. 7, no. 1, 1998.
- [5] A. Leitão, F. B. Pereira, and P. Machado, "Island models for cluster geometry optimization: how design options impact effectiveness and diversity," *Journal of Global Optimization*, vol. 63, no. 4, pp. 677–707, 2015.
- [6] Y.-J. Gong, W.-N. Chen, Z.-H. Zhan et al., "Distributed evolutionary algorithms and their models: a survey of the state-of-the-art," *Applied Soft Computing*, vol. 34, p. 286, 2015.
- [7] W. Du, M. Zhang, W. Ying et al., "The networked evolutionary algorithm: a network science perspective," *Applied Mathematics and Computation*, vol. 338, pp. 33–43, 2018.
- [8] J. L. Payne, M. Giacobini, and J. H. Moore, "Complex and dynamic population structures: synthesis, open questions, and future directions," *Soft Computing*, vol. 17, no. 7, pp. 1109–1120, 2013.
- [9] C. Lu, L. Gao, and J. Yi, "Grey wolf optimizer with cellular topological structure," *Expert Systems with Applications*, vol. 107, pp. 89–114, 2018.
- [10] C. Lu, L. Gao, Q. Pan, X. Li, and J. Zheng, "A multi-objective cellular grey wolf optimizer for hybrid flowshop scheduling problem considering noise pollution," *Applied Soft Computing*, vol. 75, pp. 728–749, 2019.
- [11] Q. Zhang and H. Li, "MOEA/D: a multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, 2007.
- [12] Y. Fu, H. Wang, M. Huang, and J. Wang, "A decomposition based multiobjective genetic algorithm with adaptive multi-population strategy for flowshop scheduling problem," *Natural Computing*, vol. 18, no. 4, p. 757, 2019.

- [13] D. S. Knysh and V. M. Kureichik, "Parallel genetic algorithms: a survey and problem state of the art," *Journal of Computer and Systems Sciences International*, vol. 49, no. 4, p. 579, 2010.
- [14] F. Yang, K. Gao, I. W. Simon, Y. Zhu, and R. Su, "Decomposition methods for manufacturing system scheduling: a survey," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 2, p. 389, 2018.
- [15] Y. Hou, N. Wu, M. Zhou, and Z. Li, "Pareto-optimization for scheduling of crude oil operations in refinery via genetic algorithm," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 3, p. 517, 2017.
- [16] A. Kimms, C. F. M. Toledo, P. M. França, R. Morabito, and A. Kimms, "Multi-population genetic algorithm to solve the synchronized and integrated two-level lot sizing and scheduling problem," *International Journal of Production Research*, vol. 47, no. 11, 2009.
- [17] M. Zandieh and N. Karimi, "An adaptive multi-population genetic algorithm to solve the multi-objective group scheduling problem in hybrid flexible flowshop with sequence-dependent setup times," *Journal of Intelligent Manufacturing*, vol. 22, no. 6, pp. 979–989, 2011.
- [18] P. Brucker and R. Schlie, "Job-shop scheduling with multi-purpose machines," *Computing*, vol. 45, no. 4, pp. 369–375, 1990.
- [19] P. Pongchairerks and V. Kachitvichyanukul, "A particle swarm optimization algorithm on job-shop scheduling problems with multi-purpose machines," *Asia-Pacific Journal of Operational Research*, vol. 26, no. 2, pp. 161–184, 2009.
- [20] J. Xiong, X. Tan, K.-W. Yang, L.-N. Xing, and Y.-W. Chen, "A hybrid multiobjective evolutionary approach for flexible job-shop scheduling problems," *Mathematical Problems in Engineering*, vol. 2012, p. 1, Article ID 478981, 2012.
- [21] K. Gao, Z. Cao, L. Zhang, Z. Chen, Y. Han, and Q. Pan, "A review on swarm intelligence and evolutionary algorithms for solving flexible job shop scheduling problems," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 4, p. 904, 2019.
- [22] A. A. R. Hosseinabadi, H. Siar, S. Shamshirband, M. Shojafar, and M. H. N. M. Nasir, "Using the gravitational emulation local search algorithm to solve the multi-objective flexible dynamic job shop scheduling problem in small and medium enterprises," *Annals of Operations Research*, vol. 229, no. 1, pp. 451–474, 2015.
- [23] W. Zhang, J. B. Wen, Y. C. Zhu, and Y. Hu, "Multi-objective scheduling simulation of flexible job-shop based on multi-population genetic algorithm," *International Journal of Simulation Modelling*, vol. 16, no. 2, pp. 313–321, 2017.
- [24] X. Shi, W. Long, Y. Li, D. Deng, and Y. Wei, "Research on the performance of multi-population genetic algorithms with different complex network structures," *Soft Computing*, vol. 24, no. 17, p. 13441, 2020.
- [25] X. Shi, W. Long, Y. Li, and D. Deng, "Multi-population genetic algorithm with ER network for solving flexible job shop scheduling problems," *PLoS One*, vol. 15, no. 5, Article ID e0233759, 2020.
- [26] J. W. Rivkin and N. Siggelkow, "Patterned interactions in complex systems: implications for exploration," *Management Science*, vol. 53, no. 7, p. 1068, 2007.
- [27] C. Guan-Rong, W. Xiao-Fan, and L. Xiang, "Introduction to complex networks: models structures and dynamics," *Nature*, 2012.
- [28] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.
- [29] M. E. J. Newman and D. J. Watts, "Renormalization group analysis of the small-world network model," *Physics Letters, Section A: General, Atomic and Solid State*, vol. 14, 1999.
- [30] X.-Q. Shi, W. Long, Y.-Y. Li, Y.-L. Wei, and D.-S. Deng, "Different performances of different intelligent algorithms for solving FJSP: a perspective of structure," *Computational Intelligence and Neuroscience*, vol. 2018, Article ID 4617816, 1 page, 2018.
- [31] C. Zhang, "Bilevel genetic algorithm for the flexible job-shop scheduling problem," *Chinese Journal of Mechanical Engineering*, vol. 43, no. 4, pp. 119–124, 2007.
- [32] W. Changjun and X. Yugeng, "Performance analysis of active schedules in identical parallel machine," *Journal of Control Theory and Applications*, vol. 5, no. 3, pp. 239–243, 2007.
- [33] J. Hutchison and Y.-L. Chang, "Optimal nondelay job shop schedules," *International Journal of Production Research*, vol. 28, no. 2, pp. 245–257, 1990.
- [34] H.-C. Chang and T.-K. Liu, "Optimisation of distributed manufacturing flexible job shop scheduling by using hybrid genetic algorithms," *Journal of Intelligent Manufacturing*, vol. 28, no. 8, p. 1973, 2017.
- [35] A. Lara-Caballero, S. G. De-Los-Cobos-Silva, R. A. Mora-Gutiérrez, E. A. Rincón-García, M. Á. Gutiérrez-Andrade, and P. Lara-Velázquez, "Multiobjective genetic algorithms for reinforcing equal population in congressional districts," *Mathematical Problems in Engineering*, vol. 2019, p. 1, 2019.
- [36] T. Qiu, J. Liu, W. Si, and D. O. Wu, "Robustness optimization scheme with multi-population co-evolution for scale-free wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 27, no. 3, p. 1028, 2019.
- [37] S. Tiwari, K. Kaur, Y. Pathak, S. Shivani, and K. Kaur, "Computed tomography reconstruction on distributed storage using hybrid regularization approach," *Modern Physics Letters B*, vol. 33, no. 6, Article ID 1950063, 2019.
- [38] L. Li and R. Mo, "A comprehensive decision-making approach based on hierarchical attribute model for information fusion algorithms' performance evaluation," *Mathematical Problems in Engineering*, vol. 2014, p. 1, Article ID 124156, 2014.
- [39] V. P. Singh, R. Srivastava, Y. Pathak, S. Tiwari, and K. Kaur, "Content-based image retrieval based on supervised learning and statistical-based moments," *Modern Physics Letters B*, vol. 33, no. 19, Article ID 1950213, 2019.
- [40] Y. Pathak, K. V. Arya, and S. Tiwari, "An efficient low-dose CT reconstruction technique using partial derivatives based guided image filter," *Multimedia Tools and Applications*, vol. 78, no. 11, Article ID 14733, 2019.
- [41] I. Kacem, S. Hammadi, and P. Borne, "Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems," *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, vol. 32, no. 1, p. 1, 2002.
- [42] E. L. Yu and P. N. Suganthan, "Ensemble of niching algorithms," *Information Sciences*, vol. 180, no. 15, p. 2815, 2010.
- [43] F. Kang, J. Li, and Z. Ma, "Rosenbrock artificial bee colony algorithm for accurate global optimization of numerical functions," *Information Sciences*, vol. 181, no. 16, p. 3508, 2011.
- [44] P. Fattahi, M. Saidi Mehrabad, and F. Jolai, "Mathematical modeling and heuristic approaches to flexible job shop scheduling problems," *Journal of Intelligent Manufacturing*, vol. 18, no. 3, p. 331, 2007.
- [45] A. Bagheri, M. Zandieh, I. Mahdavi, and M. Yazdani, "An artificial immune algorithm for the flexible job-shop

- scheduling problem,” *Future Generation Computer Systems*, vol. 26, no. 4, p. 533, 2010.
- [46] Y. Yuan, H. Xu, and J. Yang, “A hybrid harmony search algorithm for the flexible job shop scheduling problem,” *Applied Soft Computing*, vol. 13, no. 7, p. 3259, 2013.
- [47] Y. Demir and S. Kürşat İşleyen, “Evaluation of mathematical models for flexible job-shop scheduling problems,” *Applied Mathematical Modelling*, vol. 37, no. 3, p. 977, 2013.
- [48] E. G. Birgin, P. Feofiloff, C. G. Fernandes, E. L. de Melo, M. T. I. Oshiro, and D. P. Ronconi, “A MILP model for an extended version of the flexible job shop problem,” *Optimization Letters*, vol. 8, no. 4, p. 1417, 2014.
- [49] X. Li and L. Gao, “An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem,” *International Journal of Production Economics*, vol. 174, p. 93, 2016.