

Received May 9, 2020, accepted May 21, 2020, date of publication May 29, 2020, date of current version June 11, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2998539

An Improved Partheno-Genetic Algorithm With Reproduction Mechanism for the Multiple Traveling Salesperson Problem

ZIJIAN WANG^{ID}¹, XI FANG^{ID}¹, HAO LI^{ID}², AND HONGBIN JIN²

¹School of Science, Wuhan University of Technology, Wuhan 430070, China

²People's Liberation Army Air Force Early Warning Academy, Wuhan 430070, China

Corresponding author: Xi Fang (fangxi@whut.edu.cn)

This work was supported in part by the Equipment Pre-Research Fund under Grant JZX7Y20190253036101, in part by the Equipment Pre-Research Ministry of Education Joint Fund under Grant 6141A02033703, and in part by the Hubei Natural Science Foundation under Grant 019CFC897.

ABSTRACT This paper considers the problem of the multiple traveling salesperson problem (MTSP) with multiple depots and closed path. We analyzed the advantages and disadvantages of Partheno-Genetic algorithm (PGA) in solving the MTSP. By integrating the reproduction mechanism in the invasive weed algorithm (IWO), we have solved the defect that the local information of individuals in the PGA population may be missing, and obtained the improved Partheno-Genetic algorithm with reproduction mechanism (RIPGA), which greatly improves the solution performance. By comparing the RIPGA with other GAs and other intelligent algorithms on a large number of traveling salesperson problem (TSP) test functions, we have well verified the solution performance of the RIPGA on the optimal solution, indicating that the algorithm can effectively avoid local convergence. At the same time, the algorithm is less affected by parameters and has good stability.

INDEX TERMS Multiple traveling salesperson problem (MTSP), partheno-genetic algorithm (PGA), Invasive weed algorithm (IWO), reproduction mechanism.

I. INTRODUCTION

Traveling salesperson problem (TSP) is a classic NP-hard problem encountered in combinatorial optimization [1]. The objective is to find the salesperson's route with the minimum cost, under the condition that the salesperson will visit the given locations only once, and at the end, return to the starting location. Many realistic problems can be modeled as TSP after transformation, such as path planning [2], production scheduling [3], emergency management [4], *et al.*

However, classic TSP can't be modeled in some special situations. For example, suppose that a company has multiple salespersons living in different cities, and the company hopes that they visit cities in the number of not less than a certain amount as to meet their minimum wage. Therefore, to address this problem, a multiple traveling salesperson problem (MTSP) was introduced [5]. By changing the single salesperson condition into a problem involving several

The associate editor coordinating the review of this manuscript and approving it for publication was Vivek Kumar Sehgal .

salespersons, MTSP adds some additional conditions to satisfy more realistic requirements. In this sense, the MTSP is more suitable for realistic applications [6].

We know that there are two approaches to solve MTSP problems: exact direct approach and heuristic approach. The exact algorithm is an early algorithm based on strict mathematical theory, such as the branch-and-bound method for solving large-scale symmetric MTSP for the first time [7]. Although the exact algorithms have strict mathematical foundations, their ability to solve problems depends entirely on the size of the problem. When the size is increased, it may not be solved within an acceptable time, or even solved.

In order to overcome the computational difficulties of the exact algorithms after the scale of the problem is expanded, the research of heuristic algorithms have gradually become the mainstream direction. In 2015, an evolutionary programming algorithm [8] was applied to solve the maintenance network problem in the logistics industry. Afterwards, a modified two-part wolf search algorithm [9] was proposed to enhance the global search capability of the original wolf pack

algorithm. In the reference [10], a super-heuristic algorithm based on artificial bee colony algorithm was proposed to solve the k-interconnected multi-depot multi-traveling salesperson problem. It reduced the search space and the calculation time. Then Chen *et al.* [11] proposed an ant colony optimization based memetic algorithm to solve bi-objective multiple traveling salespersons problem for multi-robot systems, while optimizing the maximum travel distance and total travel distance.

The MTSP is an extension of TSP, where more than one salesperson is present to visit the cities though each city must be visited exactly once by only one salesperson. Given a set of n cities to be visited by a salesperson, the TSP seeks the shortest possible tour for the salesperson that visits each city exactly once and return to the starting city. However, in case of the MTSP, there are m salespersons instead of one to visit $n > m$ cities and we have to find the tours for all m salespersons. The starting and ending cities are called depots, and the remaining cities are called intermediate cities. There are several situations of the MTSP depending on the number of depots [10].

- Single depot case, all the m salespersons have to start and end at a given single depot.
- m depots case, every salesperson has to start and end at their own depot.
- $2m$ depots case, every salesperson has to start and end at their own 2 depots, i.e., starting and ending depot are different for every salesperson.
- >1 and $< 2m$ depots case, it is hybridization of above specified cases.

This paper mainly considers the second case, which solves the MTSP with multiple start depots and closed path.

Genetic algorithm (GA) is a random search method proposed by Professor Holland in the United States inspired by the laws of biological evolution [12], which has a good global search ability and inherent hidden parallelism, so it is widely used in combination optimization, information number processing and database query [13], [14]. Tang *et al.* [15] proposed a genetic algorithm with one-chromosome representation for MTSP problem to solve the hot rolling production scheduling problem. Malmborg [16] and Park [17] used a two-chromosome representation in their genetic algorithm for the vehicle scheduling problem (VSP) which can also be adapted for MTSP. Carter and Ragsdale [18] proposed a genetic algorithm using a new two-part chromosome representation which greatly reduces redundant solutions. Chandran *et al.* proposed a clustering approach for the MTSP with the criterion to balance workloads amongst salespersons [19]. Singh *et al.* developed a new steady-state grouping GA (GGA-SS) [20] which was a chromosome representation scheme with the least possible redundancy. Then Yuan *et al.* [21] proposed a new crossover operator called two-part chromosome crossover (TCX) for solving the MTSP using a genetic algorithm and showed its superiority over the genetic algorithm of [18].

Invasive Weed Optimization (IWO) is a random search algorithm based on the evolutionary process of weeds in nature [22]. This algorithm uses the outstanding individuals in the group to guide the evolution of the entire group. It simulates the four processes of weed seed space diffusion, growth, reproduction, and competitive extinction during the invasion to generate the optimal individual. It has strong self-adaptability and randomness. After IWO was proposed, it has been tried to be fused with other intelligent algorithms to solve the realistic problems. Hajimirsadeghi and Lucas [23] proposed a hybrid algorithm (IWO / PSO) in 2009, which reflects the intelligent use of weeds by the population information, and treats the generated weeds as particles to perceive the optimal weeds in the group, avoiding the algorithm falling into a local optimal situation. In order to overcome the precocity of the algorithm and improve the global exploration ability of the algorithm, Zhang *et al.* [24] introduced the crossover operator in the genetic algorithm when weeding offspring, and proposed an improved IWO algorithm (MIWO).

This paper mainly introduces the improved Partheno-Genetic algorithm (IPGA) [25] of fusion weed algorithm breeding mechanism for solving MTSP with multiple depots and closed paths.

Reference [25] has proved the superiority of IPGA and compared it with the IWO algorithm. Since the PGA randomly selects k individuals during the genetic operation, and then puts the one with the best fitness in a temporary array contain in k individuals, the remaining $k - 1$ individuals that are selected are eliminated, and the remaining $k - 1$ individuals in the temporary array are obtained by performing different genetic recombination operations on an existing individual. Although this operation can obtain better results when solving the MTSP with multiple depots and closed paths it has to a certain extent abandoned the use of useful information provided by slightly worse individuals. In order to solve this shortcoming, we will introduce reproduction mechanism in IPGA. The reproduction mechanism in the invasive weed optimization algorithm retains the best individuals to the greatest extent possible, and also preserves the useful information of the poor individuals as much as possible. When solving the MTSP with multiple depots and closed paths, this method can save some excellent sub-paths of the traveling salesperson to achieve a better global solution, although it will increase the acceptable computing cost based on the original algorithm.

The paper is organized as follows: Section II presents the concept of the MTSP, Partheno-Genetic algorithm (PGA) and Invasive Weed Optimization (IWO). A new improved PGA with reproduction mechanism (RIPGA) is introduced for solving the MTSP in section III. Then section IV shows the computational results of different algorithms and the analysis. Finally, we summarize the paper and puts forward some ideas for future studies in section V.

II. BASIC THEORY

As mentioned earlier, this paper mainly deals with MTSP in the case of multiple start depots and closed paths. Considering the shortcomings of Reference [25], we introduced a reproduction mechanism to improve the solution performance.

A. MTSP WITH MULTIPLE DEPOTS AND CLOSED PATHS

The MTSP we considered can be briefly described as follows:

Give an undirected graph $G = (V, A)$, which is an ordered pair $G = (V, A)$ comprising a set V of vertices, together with a set A of arcs. Let m represent the total number of salespersons. The objective function is to partition V into m nonempty subsets $\{S_i\}_{i=1}^m$, and find a minimum cost circuit passing through each vertex of each subset S_i exactly once.

The MTSP objective function is as follows:

$$\text{Minimize } \sum_{i=1}^m (x_{n^i, 1}^i + \sum_{j=1}^{n^i-1} x_{j, j+1}^i) \quad (1)$$

The first sum represents cycling through the m salespersons. The second sum represents cycling through the total cities that the i th salesperson has visited (the index of the first city that the i th salesperson has visited is 1, and the last city index is n^i). $x_{j, j+1}^i$ indicates the distance between city j and $j+1$ that the i th salesperson visited. $x_{n^i, 1}^i$ indicates the distance between the last city n^i and the first city that the i th salesperson visited. The value of n^i should not be less than the specified minimum number of cities for each salesperson. $x_{j, j+1}^i$ is equal to $x_{j+1, j}^i$.

B. PGA FOR THE MTSP

It is very important to design a better chromosome representation method based on the problem in genetic algorithm. A great genetic algorithm chromosome design should be able to minimize or eliminate redundant solutions from candidate solutions. The emergence of redundant solutions not only enlarges the understanding space, but also reduces the search efficiency.

When using genetic algorithm to solve MTSP, there are two common forms of chromosome design, called “one-chromosome” [15] and “two-chromosome” [16], [17]. We set n as the number of cities and m as the number of traveling salespersons and encode n cities to integers from 1 to n , then their forms are as follows:

The first part of Fig. 1 is a one-chromosome with 12 cities and 4 travelers. The design of a one-chromosome is to add $m - 1$ virtual points, such as $-1, -2, -3$ above. By the calculation, we can see that the solution space of one-chromosome is $(n + m - 1)!$, but there are many redundant solutions, such as the same solution expressed by the path of any two travelers. The second part of Fig. 1 is a two-chromosome with the same number of cities and travelers. Two-chromosome is designed to represent the traveling salesperson’s path with one chromosome and the corresponding city-affiliated traveling salesperson with another chromosome. The calculation shows that the solution space is $n!m^n$. The redundant solution

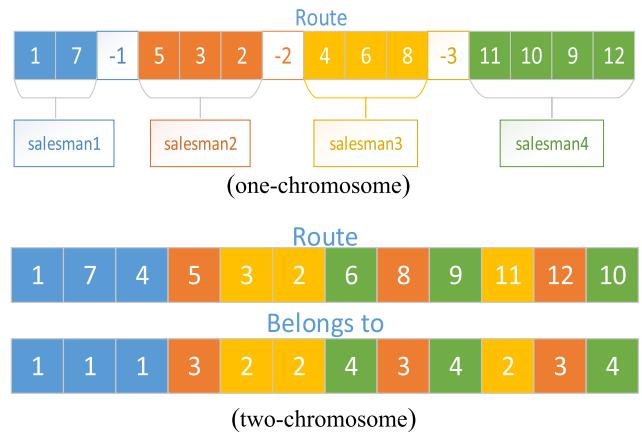


FIGURE 1. Two traditional encoding methods.

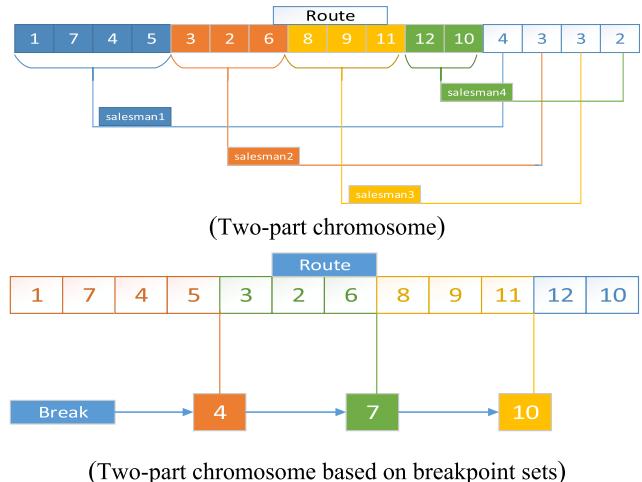


FIGURE 2. New coding method and its improvement.

and solution space of two chromosome were significantly larger than one-chromosome design.

In order to reduce the size of solution space and eliminate or reduce the number of redundant solutions, now we generally use a two-part chromosome coding based on breakpoint sets [18], as shown follow:

The first part of Fig. 2 is a two-part chromosome. The first part is the path of n cities, and the second part is the number of m cities each traveler passes through. The calculation shows that the solution space is $n!C_{n-1}^{m-1}$. The order of magnitude of the two-part chromosome solution space is significantly smaller than the traditional chromosome, and the number of redundant solutions is also less. Now we often use this way of coding base on breakpoint sets as shown in the second part of Fig. 2. with better solution performance.

In terms of genetic operators, the traditional genetic algorithm (TGA) mainly breeds offspring through the crossover operator. When the two individuals acting on the crossover operator are the same, new individuals cannot be generated. Therefore, TGA requires that the initial population has wide diversity. PGA eliminates the crossover operator of TGA, and

replaces it with a gene translocation operator that operates on only one chromosome. Iterations can be performed even if the individuals in the population are the same, so there is no need for the initial population to have wide diversity and there is no “premature convergence” problem. The paper will not repeat the content of the crossover operator, but the mutation operator and selection operator will be introduced later when IPGA is involved.

C. IWO FOR THE MTSP

Invasive Weed Optimization (IWO) is a random search algorithm based on the evolution of weeds in nature. The basic steps of IWO are as follows:

1) INITIALIZE A POPULATION

A certain number of weeds are given and distributed in the search space in a random manner. The number of weeds can be adjusted according to actual problems.

2) REPRODUCTION

Parents distributed throughout the search space produce the next generation according to the reproductive ability. Among them, a good fitness produces multiple individuals, and a poor fitness also produces a small number of individuals to ensure the diversity of the population. The formula for the number of seeds is shown in (2).

$$\text{weed}_n = \frac{f - f_{\min}}{f_{\max} - f_{\min}} (s_{\max} - s_{\min}) + s_{\min} \quad (2)$$

weed_n is the number of seeds produced by the individual of n , f is the fitness value of the current weed, f_{\max} and f_{\min} are the maximum and minimum fitness values of the weed in the population under the current iteration, and s_{\max} , s_{\min} are the maximum and minimum number of seeds that weeds can produce.

3) SPATIAL DISPERSAL

The resulting seeds are spread around the weeds in a normal distribution with the weeds' fitness as the mean. Since the algorithm in this paper only introduces the reproduction mechanism, this part will not be introduced in depth.

4) COMPETITIVE EXCLUSION

After multiple generations of breeding, the amounts of weeds after seed growth will make the environmental resources unbearable. Therefore, the offspring and the parents are sorted according to the set fitness value, and the individuals with good fitness are continued to breed, and the poor individuals to reduce environmental carrying capacity.

III. IMPROVED PARTHENO-GENETIC ALGORITHM WITH REPRODUCTION MECHANISM

A. ENCODING

We choose the two-part chromosome encoding based on the breakpoint sets mentioned in Section II. Take Fig. 2 as an example. Fig. 2 shows the MTSP problem with the number

Parent	Route												Breakpoint set		
	1	3	4	11	5	7	6	2	12	10	9	8	3	7	9
Weed 1	1	3	4	11	5	7	6	2	12	10	9	8	5	8	10
Weed 2	1	3	4	11	5	7	6	2	12	10	9	8	4	6	10
Weed 3	1	3	4	11	5	7	6	2	12	10	9	8	3	5	9

FIGURE 3. How seeds are produced by reproduction mechanism.

of cities being 12 and the traveling salesperson being 4 (the number of breakpoints + 1). The path can be expressed as:

Salesperson 1: city 1 → 7 → 4 → 5

Salesperson 2: city 3 → 2 → 6

Salesperson 3: city 8 → 9 → 11

Salesperson 4: city 12 → 10

where every travelling salesperson's path is a closed loop.

B. FITNESS FUNCTION

The population is evaluated by the fitness function (1). In most cases, the greater individual's fitness value indicates that the individual is more adaptive to the environment. Here, the fitness value is the total distance. The smaller total distance suggests a better route.

C. REPRODUCTION MECHANISM

The specific operation is as follows: first we calculate the fitness of the initial population, then we can achieve each individual generates the corresponding number of children according to (2), as shown in Fig. 3, where the city order of each child is the same as the parent, and the breakpoints in the second part are randomly set and different from the parent.

D. GENETIC OPERATION

The genetic operation process of RIPGA is almost the same as IPGA, except that the offspring which will be operated becomes the offspring generated in Fig. 3 according to the reproduction mechanism.

The process of genetic operation of RIPGA proceeds as follows:

1. Randomly select 8 individuals that have not been selected from the contemporary population.
2. Find the best individual that has the best fitness in 8 individuals just selected.
3. Create a temporary population that consists of 8 individuals. All of the 8 individuals are assigned to the best individual found in procedure 2.
4. Generate 2 random mutation segment selection points I and J, and the mutation segment insertion location P.

This way I call him group evolution which the optimal individual selected in the temporary population produces offspring through 7 other mutation methods in Fig. 4.

Reference [25] has confirmed the superiority of this method. The above group evolutionary processes are

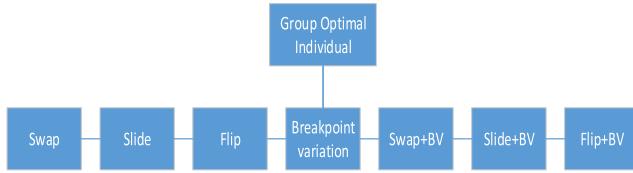


FIGURE 4. Group evolution, that is, the temporary population changes in multiple directions according to the group's optimal individual.

Input: Population set P , Population size m , fitness function f , largest generations g_{\max} .

- 1: Initialize P, g, i ,
- 2: **for** $g = 1$ to g_{\max}
- 3: **for** $i = 1$ to m
- 4: Calculate f_i ;
- 5: Generate offspring population ff_i of each individual according to reproduction mechanism.
- 6: Sort the ff_i to obtain the offspring population sff ;
- 7: Use IPGA to do the genetic recombination of sff ;
- 8: Get new population set P_{next} ;
- 9: Use a mixed selection operator on P_{next} to get P_{new} ;
- 10: Finding the best individual P_{best} and its fitness f_{best} ;
- 11: **end for**
- 12: **if** $f_{best} >$ actual need **then**
- 13: **end for**
- 14: **else if** $g = g + 1$;
- 15: **end for**

Output: the best individual P_{best} , best fitness f_{best} .

FIGURE 5. Algorithm of RIPGA.

as follows:

$$\begin{aligned}
 Swap : [x_1 x_2 x_3 x_4 x_5 x_6] &\xrightarrow{x_1 \text{ and } x_4} [x_4 x_2 x_3 x_1 x_5 x_6] \\
 Slide1unit : [x_1 x_2 x_3 x_4 x_5 x_6] &\xrightarrow{x_1 x_2 x_3} [x_3 x_1 x_2 x_4 x_5 x_6] \\
 Flip : [x_1 x_2 x_3 x_4 x_5 x_6] &\xrightarrow{x_3 x_4 x_5 x_6} [x_1 x_6 x_5 x_4 x_3 x_2] \\
 Breakpointmutation : [x_1 x_2 x_3] &\xrightarrow{x_3 \text{ to } x_4} [x_1 x_2 x_4]
 \end{aligned} \quad (3)$$

In order to prevent falling into a local optimum at the later stage of the iteration, we will adopt the best individual retention strategy according to a certain ratio, which we call the mixed selection operator.

Our RIPGA first puts the parents and offspring together according to fitness, selects 3/4 individuals in the population

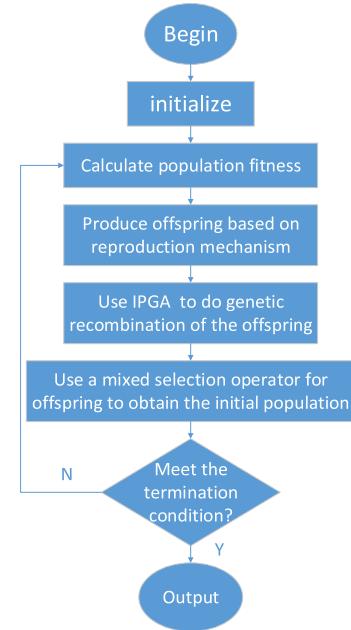


FIGURE 6. Flow chart of RIPGA.

TABLE 1. Results and time for the algorithms.

Algorithm	Total distance	Running time
IPSO	34004.8100	173.57s
IPGA	31314.7271	55.12s
RIPGA	25083.4809	68.75s

directly into the next generation according to the best individual retention strategy, and the remaining 1/4 individuals according to the original population generation method produce. The mixed selection operator can not only keep the optimal solution in each iteration, but also prevent the algorithm from falling into premature convergence to a certain extent.

According to the previous introduction, we can obtain the pseudocode of the improved algorithm. It is given in Fig. 5 and the flow chart of the RIPGA is shown in the Fig. 6 We record the shortest path length of each generation and the distance of each traveler to visually describe the effect of load balancing, that is, the workload balancing of each travelling salesperson.

IV. EXPERIMENTS AND ANALYSIS

The simulation experiment is divided into two parts. Firstly, the traditional algorithm is compared with the improved algorithm in the horizontal and vertical direction. Secondly, the stability of the improved algorithm is verified in a large number of numerical examples. All the simulation platforms in this paper is Intel (R) Core (TM) i7-7700HQ CPU, 8.0GB RAM, and the experimental environment is MATLAB 2016a.

In RIPGA, we set the maximum and minimum number of seeds to $N_{\max} = 5$ and $N_{\min} = 1$. In order to ensure the fairness of the experiment, we keep the parameters as consistent as possible when comparing, and the comparison algorithm is consistent with the parameters of the reference [26].

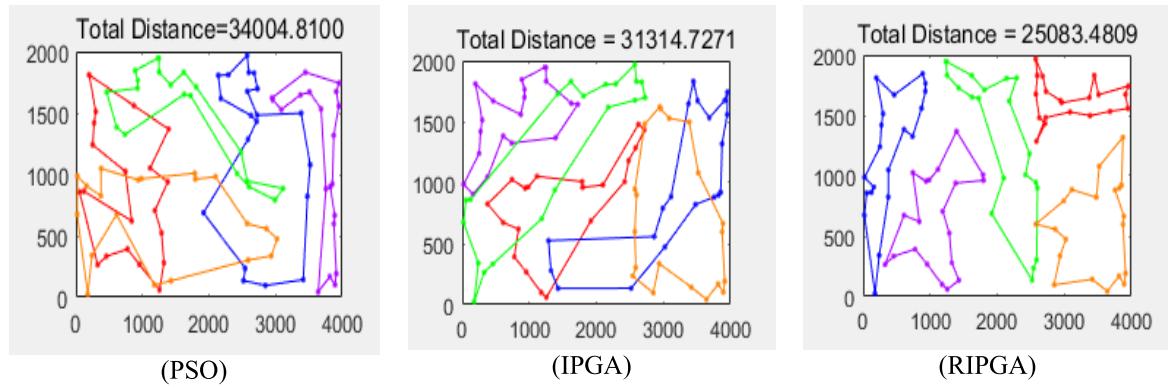


FIGURE 7. Solutions of IPSO, IPGA and RIPGA.

TABLE 2. Comparison of RIPGA and improved GAs.

n	m	IPGA			AC-PGA			RIPGA		
		mean	best	var	mean	best	var	mean	best	var
51	3	481	468	13.6	486	467	9.5	509	481	21.4
	5	509	489	16.3	497	488	10.5	492	468	13.2
	8	549	526	16.2	530	521	11.5	499	483	10.7
	10	542	526	13.6	546	532	8.9	502	482	12.3
150	3	80065	74819	3206	36041	35053	545.2	37498	34066	2081
	5	87556	82802	3469	37679	37102	542.9	38211	36891	3164
	8	90951	82824	4060	40942	38862	1288.2	41140	37363	3537
	10	101981	91221	4432	42036	40396	1296.6	42669	38855	2076
280	3	11482	11016	496	3753	3664	49.9	3083	3006	46.9
	5	12589	12135	517	3983	3890	68.1	3080	3012	48.5
	8	13752	12840	596	4381	4283	63.4	3117	3039	68.2
	10	14678	13259	732	4435	4357	66.1	3200	3120	36.1

A. COMPARISON OF PSO, IPGA AND RIPGA

In this part, we test IPSO, IPGA, and RIPGA against a multiple-travel salesperson problem in 100 cities with 5 salespersons. The main purpose is to compare RIPGA with improved GA and other improved intelligent algorithms such as IPSO in [21]. We will conduct more experimental comparisons in the next part.

Regardless of the path results from **Fig. 7** or the iterative curves from **Fig. 8**, the results of RIPGA are obviously better than the other two algorithms. From the running time of **Tab. 1**, it is consistent with the theory that the running time of RIPGA is slightly higher than IPGA due to the reproduction mechanism, and we think this time cost is worth being spent.

B. COMPARISON OF OTHER ALGORITHMS

In this section, we first compare RIPGA with other improved GAs, and then compare it with other intelligent algorithms. We explore how the number of cities can influence the solution and compare the test results of the six algorithms.

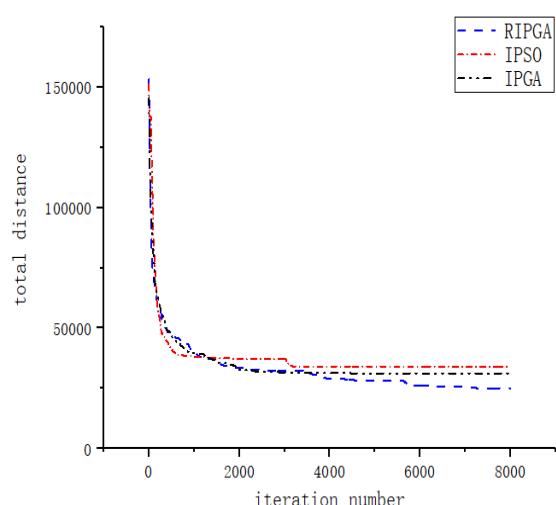
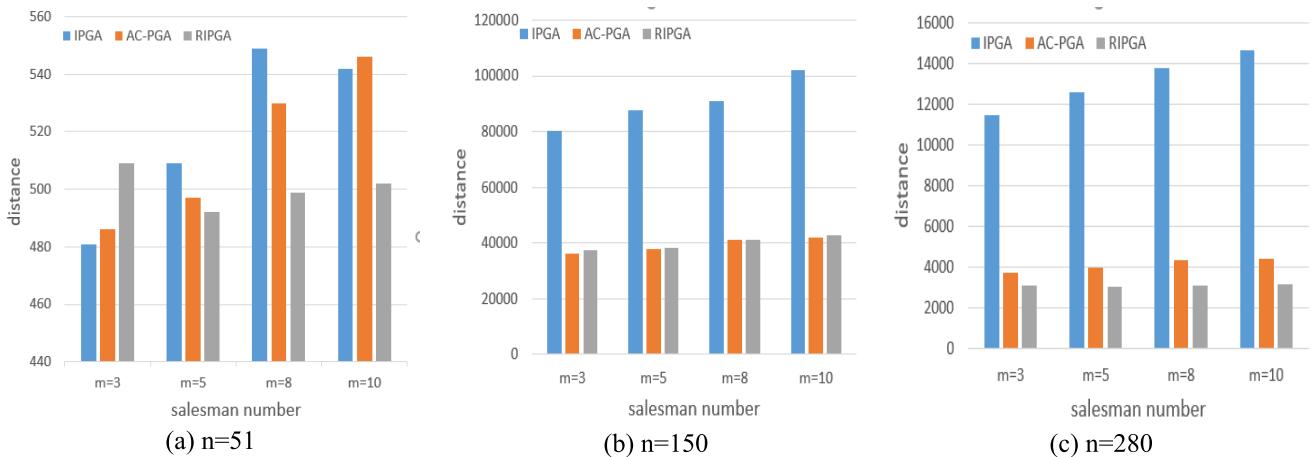
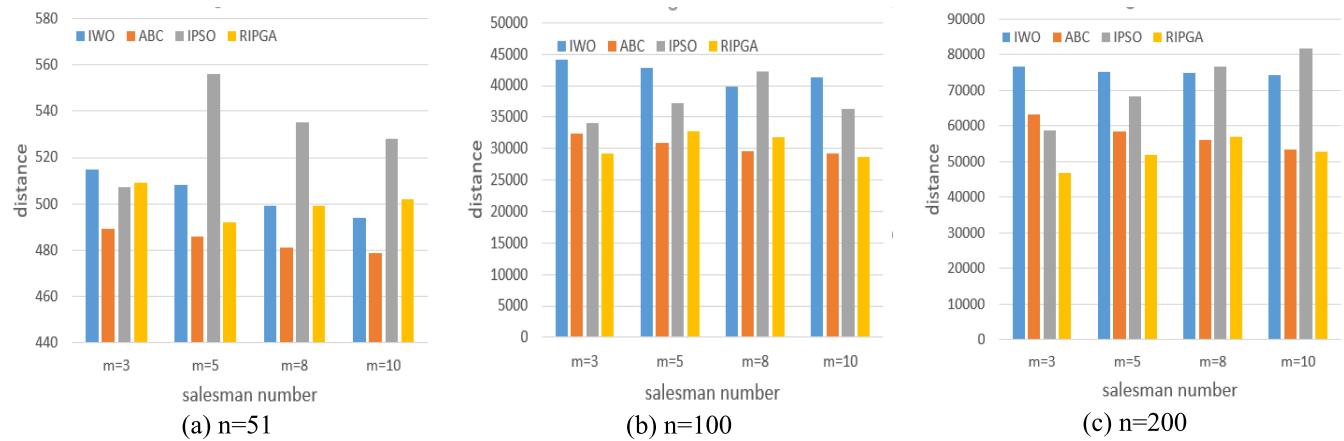


FIGURE 8. Solutions of IPSO, IPGA and RIPGA.

From **Tab. 2**, we can see that RIPGA has the best performance in solving optimal solutions. When the number of cities is small, it may be slightly inferior to AC-PGA, but

**FIGURE 9.** Solutions of different GAs when $n = 51, 150, 280$.**FIGURE 10.** Solutions of different intelligent algorithms when $n = 51, 100, 200$.**TABLE 3.** Comparison of RIPGA and other intelligent algorithms.

n	m	IWO			ABC			IPSO			RIPGA		
		mean	best	var									
51	3	515	495	11.4	489	475	9.7	507	499	22.2	509	482	21.4
	5	508	485	11.2	486	474	11.9	556	511	32.8	492	468	13.2
	8	499	481	11.0	481	470	12.1	535	505	23.8	499	483	10.7
	10	494	474	11.7	479	472	11.1	528	481	23.0	502	482	12.3
100	3	44034	41834	1056	32387	30137	940	34111	32020	2223	29189	26370	1952
	5	42877	41255	902	30882	29772	733	37280	34576	2328	32773	27129	3773
	8	39891	37251	1623	29536	28626	945	42344	39560	3510	31859	27758	3362
	10	41239	38511	1215	29156	28966	921	36365	35728	3963	28745	27651	836
200	3	76681	74103	1520	63257	61612	872	58677	56384	3480	46963	42736	3535
	5	75321	74876	925	58358	56437	1277	68374	63917	3692	51882	48683	1792
	8	74817	72436	1400	56088	54464	1408	76826	75362	2443	57109	52582	3066
	10	74204	72458	1341	53519	50630	1564	81719	80704	2789	52713	49832	2959

TABLE 4. Effect of parameters on results.

size	m=3		m=10		m=20	
	mean	time	mean	time	mean	time
20	3245	8.2s	3429	10.7s	3654	12.6s
60	3119	27.6s	3359	35.7s	3565	41.9s
100	3126	59.2s	3280	68.2s	3477	80.4s
160	3078	292.5s	-	-	-	-

when the number of cities increases, its performance is the best.

Then we compare RIPGA with other types of intelligent algorithms to test its performance. We can also see in Tab. 3 that RIPGA's solution performance will gradually improve as the number of cities increases.

We display the data of Tab. 2 and Tab. 3 in Fig. 9 and Fig. 10. We can also clearly see that the optimal solution of RIPGA is better than other algorithms in different situations. And then we test the running time and optimal solution of RIPGA with different numbers of travelling salespersons under different size of population.

Taking $n = 280$ as an example, we test the running time and optimal solution when the size of population is 20, 60, 100 or 160, $N_{\min} = 1$ and $N_{\max} = 5$. The results are shown in the Tab. 4.

In the experiment, we find that the computational cost was too high when the population size reached 160, so we discard this situation in the second half of this experiment. As can be seen from the data in the Tab. 4, the running time of the algorithm is greatly affected by the parameters, but the optimal solution is less affected by the parameters, which indicates that its performance is steady and can adapt to various situations (even if the population size is not large, it can achieve better results). Taken together, it is more appropriate to select a population size of 60-100 in most cases.

V. CONCLUSION AND FUTURE RESEARCH PROSPECT

In this paper, an improved algorithm combining PGA with IWO called RIPGA, is designed for solving the MTSP with multiple depots and closed path. We know that in today's PGA and its improved algorithms, there is an unavoidable defect: the lack of communication between individuals may lead to the loss of better local information among poor individuals. In order to solve this defect, we introduce the reproduction mechanism in the IWO algorithm, where individuals produce a certain number of seeds to save their excellent local information, and then use a hybrid selection operator to generate a good initial population.

In the comparison experiments, we compared the RIPGA and other GAs and other intelligent algorithms in this paper by a large number of MTSP standard test functions. Then, through the parameter changes in the same instance, test

whether the running time and optimal solution performance of the algorithm are greatly affected by the parameters. The experimental results show that RIPGA has better solution performance than other algorithms, and it can better show its advantages under more complex examples. The results of the parameter tests indicate that the calculation results and time cost of RIPGA are less affected. Simultaneously, the algorithm in this paper is not only suitable for solving the MTSP with multi start depots and closed paths, but also suitable for other conditions of MTSP such as single-start point and single-end point. In this regard, we only need to modify the fitness function directly and the encoding method in details.

The algorithm in this paper only solved the shortcomings of the lack of communication among individuals in the PGA population from another perspective, and didn't solve the problem in essence. How to increase the communication between individuals in the PGA population will be explored in future research. Considering that the MTSP with multiple depots and closed path is a group optimization problem, there are two cases of inter-group and intra-group optimization. Our next research direction is based on a two-step algorithm: first through classification or clustering algorithms, the MTSP is transformed into a TSP, and then the TSP in the group is solved by a kind of intelligent algorithm.

REFERENCES

- [1] M. Held, A. J. Hoffman, E. L. Johnson, and P. Wolfe, "Aspects of the traveling salesman problem," *IBM J. Res. Develop.*, vol. 28, no. 4, pp. 476–486, Jul. 1984, doi: [10.1147/rd.284.0476](https://doi.org/10.1147/rd.284.0476).
- [2] M. A. P. Muniandy, L. K. Mee, and L. K. Ooi, "Efficient route planning for travelling salesman problem," in *Proc. IEEE Conf. Open Syst. (ICOS)*, Subang, Malaysia, Oct. 2014, pp. 24–29, doi: [10.1109/ICOS.2014.7042404](https://doi.org/10.1109/ICOS.2014.7042404).
- [3] T. P. Bagchi, J. N. D. Gupta, and C. Sriskandarajah, "A review of TSP based approaches for flowshop scheduling," *Eur. J. Oper. Res.*, vol. 169, no. 3, pp. 816–854, Mar. 2006.
- [4] X. Fang, W. Wang, L. He, Z. Huang, Y. Liu, and L. Zhang, "Research on Improved NSGA-II Algorithm and Its Application in Emergency Management," *Math. Problems Eng.*, vol. 2018, Jan. 2018, Art. no. 1306341.
- [5] T. Bektas, "The multiple traveling salesman problem: An overview of formulations and solution procedures," *Omega*, vol. 34, no. 3, pp. 209–219, 2006.
- [6] C. Okonjo-Adigwe, "An effective method of balancing the workload amongst salesmen," *Omega*, vol. 16, no. 2, pp. 159–163, 1988.
- [7] Gavish, Bezalel, and K. Srikanth, "An optimal solution method for large-scale multiple traveling salesmen problems," *Oper. Res.*, vol. 34, no. 5, pp. 698–717, 1986.
- [8] L. Kota and K. Jarmai, "Mathematical modeling of multiple tour multiple traveling salesman problem using evolutionary programming," *Appl. Math. Model.*, vol. 39, no. 12, pp. 3410–3433, 2015.
- [9] Y. Chen, Z. Jia, X. Ai, D. Yang, and J. Yu, "A modified two-part wolf pack search algorithm for the multiple traveling salesmen problem," *Appl. Soft Comput.*, vol. 61, pp. 714–725, Dec. 2017.
- [10] P. Venkatesh and A. Singh, "Two Metaheuristic approaches for the multiple traveling salesperson problem," *Appl. Soft Comput.*, vol. 26, pp. 74–89, Jan. 2015.
- [11] X. Chen, P. Zhang, G. Du, and F. Li, "Ant colony optimization based memetic algorithm to solve bi-objective multiple traveling salesmen problem for multi-robot systems," *IEEE Access*, vol. 6, pp. 21745–21757, 2018.
- [12] J. H. Holland, "Genetic algorithms and classifier systems: Foundations and future directions," in *Proc. 2th Int. Conf. Genetic Algorithms Appl.*, New Jersey, NJ, USA: Lawrence Erlbaum Associates Publishers, 1987, pp. 82–89.

- [13] A. Senoussi, S. Dauzère-Pérès, N. Brahimi, B. Penz, and N. K. Mouss, "Heuristics based on genetic algorithms for the capacitated multi vehicle production distribution problem," *Comput. Oper. Res.*, vol. 96, pp. 108–119, Aug. 2018.
- [14] E. Sevinc and A. Cosar, "An evolutionary genetic algorithm for optimization of distributed database queries," *Comput. J.*, vol. 54, no. 5, pp. 717–725, May 2011.
- [15] Tang, Lixin, "A multiple traveling salesman problem model for hot rolling scheduling in Shanghai Baoshan Iron & Steel Complex," *Eur. J. Oper. Res.*, vol. 124, no. 2, pp. 267–282, 2000.
- [16] C. J. Malmborg, "A genetic algorithm for service level based vehicle scheduling," *Eur. J. Oper. Res.*, vol. 93, no. 1, pp. 121–134, Aug. 1996.
- [17] Y.-B. Park, "A hybrid genetic algorithm for the vehicle scheduling problem with due times and time deadlines," *Int. J. Prod. Econ.*, vol. 73, no. 2, pp. 175–188, Sep. 2001.
- [18] A. E. Carter and C. T. Ragsdale, "A new approach to solving the multiple traveling salesperson problem using genetic algorithms," *Eur. J. Oper. Res.*, vol. 175, no. 1, pp. 246–257, Nov. 2006.
- [19] Chandran, Nishanth, T. T. Narendran, and K. Ganesh, "A clustering approach to solve the multiple travelling salesmen problem," *Int. J. Ind. Syst. Eng.*, vol. 1, no. 3, pp. 372–387, 2006.
- [20] A. Singh and A. S. Baghel, "A new grouping genetic algorithm approach to the multiple traveling salesperson problem," *Soft Comput.*, vol. 13, no. 1, pp. 95–101, Jan. 2009.
- [21] Yuan, Shuai, "A new crossover approach for solving the multiple travelling salesmen problem using genetic algorithms," *Eur. J. Oper. Res.*, vol. 228, no. 1, pp. 72–82, 2013.
- [22] A. R. Mehrabian and C. Lucas, "A novel numerical optimization algorithm inspired from weed colonization," *Ecol. Informat.*, vol. 1, no. 4, pp. 355–366, Dec. 2006.
- [23] H. Hajimirsadeghi and C. Lucas, "A hybrid IWO/PSO algorithm for fast and global optimization," in *Proc. IEEE EUROCON*, St.-Petersburg, Russia, May 2009, pp. 1964–1971.
- [24] X. Zhang *et al.*, "A modified invasive weed optimization with crossover operation," in *Proc. IEEE 8th World Congr. Intell. Control Automat. (WCICA)*, Jinan, China, Jul. 2010, pp. 11–14.
- [25] H. Zhou, M. Song, and W. Pedrycz, "A comparative study of improved GA and PSO in solving multiple traveling salesmen problem," *Appl. Soft Comput.*, vol. 64, pp. 564–580, Mar. 2018.
- [26] C. Jiang and Z. Z. Wan Peng, "A new efficient hybrid algorithm for large scale multiple traveling salesman problems," *Expert Syst. Appl.*, vol. 139, pp. 112–867, Jan. 2020.

• • •