IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 3, 2021, accepted August 27, 2021, date of publication September 1, 2021, date of current version September 15, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3109298

Path Optimization of Gluing Robot Based on Improved Genetic Algorithm

YUHANG ZHANG[1,2], ZILING SONG[1], JING YUAN[2], ZHIYUN DENG[3], HAN DU[3], AND LIDAN LI[4]

[1]College of Mining, Liaoning Technical University, Fuxin 123000, China
[2]College of Applied Technology and Economic Management, Liaoning Technical University, Fuxin 123000, China
[3]State Key Laboratory of Hydroscience and Engineering, Department of Hydraulic Engineering, Tsinghua University, Beijing 100084, China
[4]College of Science, Liaoning Technical University, Fuxin 123000, China

Corresponding author: Han Du (duh@mail.tsinghua.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 51474119, in part by the Youth Project in Liaoning Province Department of Education under Grant LJ2020QNL008, and in part by the Discipline Innovation Team of Liaoning Technical University under Grant LNTU20TD-07.

ABSTRACT The steel cord conveyor belt surface is prone to damage in mining. The worn belt surface has acceleration characteristics, so timely and rapid repair is very necessary. To quickly and automatically repair the worn belt surface is a core design objective of the gluing robot (GR). Based on this objective, a new variant Traveling Salesman Problem (TSP) is put forward: after the worn segments are divided according to the worn information and GR's workspace, path optimization of the gluing robot (POGR) problem is presented at a certain worn segment; then the POGR is simplified into a ''double vertices'' TSP problem by Hamilton graph, and the mathematical model is built. An improved genetic algorithm (IGA) is proposed to handle the POGR problem, which is called IGA-POGR. The main benefit of the proposed IGA-POGR is the ability to solve POGR of different scales in different ways. The performance of the IGA-POGR is illustrated on four well-known TSP problems. Numerical results show that IGA-POGR does not give any deviation (0%) from the optimal solution. Compared with discrete particle swarm optimization (DPSO), IGA-POGR has better performance in terms of the solving quality and time consumption when solving four idealized POGR problems.

INDEX TERMS Mining steel cord conveyor belt, gluing robot (GR), path optimization of gluing robot (POGR), improved genetic algorithm (IGA).

ABBREVIATION AND NOMENCLATURE

Non-Destructive Testing	NDT.
Gluing Robot	GR.
Total Repair Time	TRT.
Vulcanization Time	VT.
Natural Cooling Time	NCT.
Repair Time	RT.
Path optimization of gluing robot	POGR.
Traveling Salesman Problem	TSP.
Non-deterministic Polynomial	NP.
Improved Genetic Algorithm	IGA.
Path Optimization of Gluing Robot	POGR.
Discrete Particle Swarm Optimization	DPSO.
Simulated Annealing	SA.
Ant Colony System	ACS.
Partially-Mapped Crossover	PMX.
Exchange Mutation	EM.
Rank-based Roulette Wheel Selection	RB-RWS.
Exchange Order of Multipoint	EOM.
Tournament Selection	TS.
Exchange Order of Single-point	EOS.
Rank-Based Roulette Wheel Selection	RB-RWS.

The associate editor coordinating the review of this manuscript and approving it for publication was Hongwei Du.

I. INTRODUCTION

China has abundant coal resources, but the coal mining environment is harsh. With the development of the intelligent mine [1]–[4], new intelligent inspective devices and coal mine robots are more widely used to improve working efficiency and ensure the safety of production [5]–[10]. The belt is mainly used for coal collection and transportation, and it plays an important role in mining operations. Because of complicated condition of the raw coal, relatively fixed

point of falling materials, fatigue and other force majeure, numerous damages of the belt such as belt aging, the worn surface, corrosion, broken wire core, belt deviation, skidding and fracture could occur, and even cause casualties [11]–[13]. Factually, though accidental downtime led by belt damages is short, the coal mine enterprise may suffer from enormous economic losses. The causes of these losses include the need of a new custom-made belt, the interruption of mining production and a halt to the whole technological process [14]. Hence, these conveyor belts demand an appropriate attention. In order to ensure its safe and efficient work, it is necessary to check the belt regularly to reduce the uncertainty in the production process and eliminate hidden dangers.

Belt detection is an issue of great importance, a great majority of literatures focus on intelligent inspection of the belt and presenting the results of belt tests. The ability, to get the real-time data about the belt damages and to interpret the information appropriately, allows to control unexpected changes or to take measures before breakdown. As the mainstream method, non-destructive testing (NDT) gets special attention these years. Among the methods of NDT, X-ray and magnetic sensors detection methods are widely used in practice. Cui *et al.* designed a monitoring system based on X-ray. The system can realize on-line and full speed detection and provide X-ray transmission image at any location of the conveyor belt. It can locate the areas of wire damage, and give an alarm automatically [15]. In 2012, a fault automatic detection method, basing on the statistical features and the idea of regularity, was put forward to monitor the conditions of the belt. It can distinguish the fault region from the fault-free region more clearly [16]. The magnetic image analysis method was proposed by the Australian A.Harrison in the late 1970s [17]. It is mainly used to inspect broken wire core, deformation, corrosion and joint damage. In 2018 [14], owing to the high resolution of the magnetic probe, DiagBelt System can detect the cuts of individual cord, the broken wire in the cords, and even the corrosion of cords in the core. Also, in this work, magnetic probe can detect the location of all splices in the belt loop. In addition, all changes of the belt segments and splices can be visualized as a 2D image. What's more important is the estimation of the worn belt surface, the quantity and the total area. Maintenance work becomes easy to do. Based on the data gathered with the use of the DiagBelt mobile system, A Kirjanów-Błażej *et al.* suggests that the already existing defects of the surface areas grow faster than the new defects, since new defects usually have small size. Hence, the greatest increase in the surface area is caused by the already existing and growing defects, not by the new ones. Early detection of the microdamage of the belt surface allows the user to plan maintenance more precisely [18].

The worn belt surface has acceleration characteristics, and, more importantly, the worn belt surface should be repaired before it gets worse. This verdict was based upon both the literature study [18] and on-the-spot investigation. Therefore, the worn belt surface must be repaired in time. However, there are few studies on automatic repair of the worn belt surface. At present, the defects of traditional manual repair are high labor intensity and low efficiency. Depended on present work of the belt detection and motivated by a real-life application in mining, Conveyor Belt Research Team of Liaoning Technical University has begun to design the gluing robot (GR) [19]. GR has the ability to repair the worn belt surface automatically. Meanwhile, GR will effectively prevent the acceleration of worn belt surface. Application of GR can reduce the probability to stop production due to conveyor belt failure, improve repair efficiency, as well as saving maintenance costs.

Research shows that, when repairing the worn belt surface, the total repair time is usually determined, about 12 hours. The total repair time consists of repair time, vulcanization time and natural cooling time, which is shown in Eq (1).

$$TRT = VT + NCT + RT \tag{1}$$

where TRT represents the nomenclature of Total Repair Time parameter; VT represents the nomenclature of the Vulcanization Time parameter; NCT represents the nomenclature of the Natural Cooling Time parameter; RT is the Repair Time index.

In order to ensure the quality of the belt, vulcanization time and natural cooling time have strict regulations in automatic repair. Repairing time is limited, so improving repair efficiency is an important design objective of the GR. In the limited repair time, optimizing the path of GR will repair more worn surfaces, decrease time loss and improve repair efficiency.

The main purpose of this article is to determine the path optimization of the GR (POGR) problem and solve it. According to the characteristics of the POGR, it is fairly easy to convert into a ''double vertices'' Traveling Salesman Problem (TSP). TSP problem is an Non-deterministic Polynomial (NP) complete problem. There is no solution to solve it in polynomial time so far. As an intelligent algorithm, genetic algorithm (GA) allows us to find optimal or nearly optimal solutions of various problems in a very efficient way [19]. Therefore, since the basic GA appears, scholars all around the world began to use GA to solve TSP in the last decades and proposed many improvements [20]–[24]. Considering the uncertainty number of every POGR, an improved genetic algorithm (IGA) is proposed to handle the TSP problem, which is called IGA-POGR in this paper. The main benefit of the proposed IGA-POGR is the ability to solve POGR of different scales in different ways. It uses the same genetic operators but different methods and multiplicative parameters to solve the POGR. When the optimal path is got by IGA-POGR, GR will walk on the planned path and achieve automatic repair at the same time.

The rest of the paper is organized as follows: In Section II, related work will be reviewed and give a brief introduction of the proposed IGA-POGR algorithm. In section III, firstly, repairing the belt by GR is divided into four steps (as shown in Fig.1 1∼4). Then the concept of path optimization of

gluing robot (POGR) is introduced. In Section IV, a new variant TSP problem (''double vertices'' TSP) is put forward. The POGR problem is simplified into a ''double vertices'' TSP problem by Hamilton graph. Then build the mathematical modeling. The novelty and improvement of the IGA-POGR are presented in Section V. In Section VI, The performance of the IGA-POGR is illustrated on four well-known TSP problems. A comparative experimental study between IGA-POGR and discrete particle swarm optimization (DPSO) will be covered which is followed by the experimental conclusion. The comparative experiment is based on four idealized POGR problems. Finally, conclusions and future work are given in Section VII.

II. RELATED WORK

POGR is a ''double vertices'' TSP. TSP is a typical NP-hard problem. In literature search, there are many evolutionary computations used to solve TSP, such as particle swarm optimization (PSO) [25], simulated annealing (SA) [26], ant colony system (ACS) [27] and some others. Furthermore, Goldberg first applied GA for TSP, and achieved a short tour [28].

GA is optimization algorithm. The main idea of GA is to mimic biological evolution [29]. Studies on the application of GA to optimization problems and the effect of operators on the behavior of GA are presented [30]–[32]. In literature, many GAs have been proposed and improved to solve TSP [20]–[24]. The path representation is probably the most natural representation of a tour [33]. Hui [34] used path representation to encode the path of the TSP problem and found the global optimum. Yifei *et al.* [35] used the natural number coding genetic algorithms for welding robot path optimization. Through consulting from the related optimization literature [36], [37], when meeting the maximization problems, the roulette wheel method is more preferred. When meeting the minimization problems, the tournament selection method is more efficient. Also, Stern *et al.* gave a same suggestion about the three popular selection operators: proportional selection, ranking selection [38] and tournament selection [39], [40]. The partially-mapped crossover (PMX) operator passes on ordering and value information from the parent generation to the offspring generation [34]. As the literature [41], [42] described, PMX tries to preserve the absolute positions of the integers. The exchange mutation (EM) operator randomly selects two cities in the tour and exchanges them [43]. So does the partheno-genetic operator [44], [45].

The POGR problem has uncertainty number of worn blocks in each worn segment. Through the investigation of the worn belt surface in the coal mines of Shanxi, China, combined with GR's workspace, the POGR problem is divided into small-scale POGR and large-scale POGR. The number 20 is the dividing line. In most studies, when GA is used to solve TSP, a fixed representation scheme and a fixed set of operators are chosen. Due to the complexity of the POGR, it is important to choose a suitable solution. Thus, IGA-POGR is proposed. The main purpose of the IGA-POGR is to use the same genetic operators but different methods and different parameters to solve the POGR. The genetic operators of IGA-POGR are selection operator, crossover operator, and partheno-genetic operator. If the scale is greater than or equal to 20, selection operator selects Liner Rank-based Roulette Wheel Selection (RB-RWS), crossover operator selects the PMX and partheno-genetic operator selects Exchange Order of Multipoint (EOM) [44], [45]; if the scale is less than 20, selection operator selects Tournament Selection (TS), crossover operator selects PMX and partheno-genetic operator selects Exchange Order of Single-point (EOS) [44], [45].

III. PROBLEM DESCRIPTION

Path optimization steps of GR to repair the worn belt surface are shown in Fig. 1.

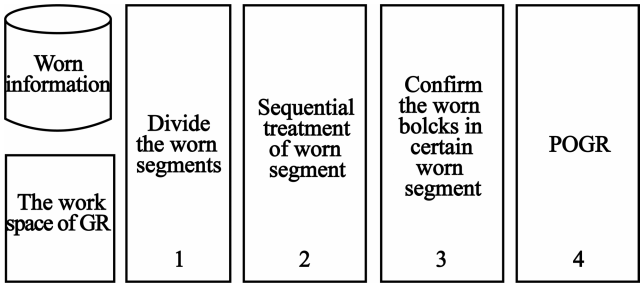


FIGURE 1. Steps of GR to repair the worn belt surface.

As the main conveying tools, the length of the belt is generally more than 2000 m. The worn location of the belt is random. It may concentrate on a certain area or disperse in different positions of the belt. After determining the worn position, according to certain conditions, such as the worn information [14], [18] and the GR's workspace, the worn segments are divided in order to repair the worn belt surface efficiently. Then, on the basis of the forward direction of the belt, the GR sequentially handles the worn segments and achieves automatic repair.

There may be N worn blocks in each worn segment. After dividing the belt, schematic diagram of the worn blocks in worn segments is shown in Fig. 2. Fig. 2a is a schematic diagram of the worn segments. Fig. 2b is a series of pictures relating to N worn blocks in a certain worn segment. Path optimization of gluing robot (POGR) is used to determine the walking path among N worn blocks in a certain segment. (The worn segments in Fig. 2 have different number of worn blocks, each of which is a real picture of the worn belt).

The lines in Fig. 3 show the feasible walking path of the GR in worn segment 1. The repairing process of GR starts with one worn block. The process will not stop until all worn blocks are repaired in the same worn segment. GR will go back to the starting point at last. The blue solid line is the walking path of the GR in the worn blocks, and the red dotted line is the walking path of the GR among the worn blocks. The blue solid lines are seen as length invariant and not considered in the POGR. The red dotted lines are the object of the POGR.

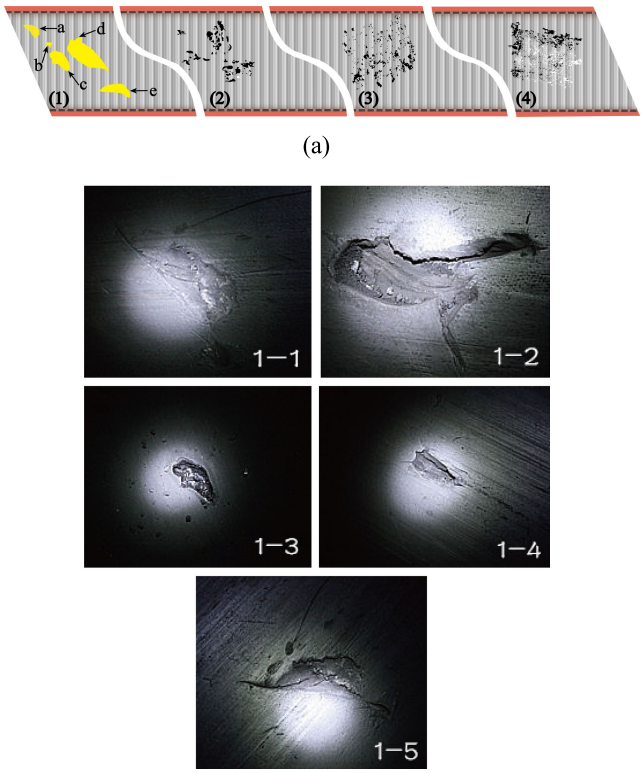


FIGURE 2. a Schematic diagram of the worn segments after dividing the belt; b five worn blocks in worn segment 1.

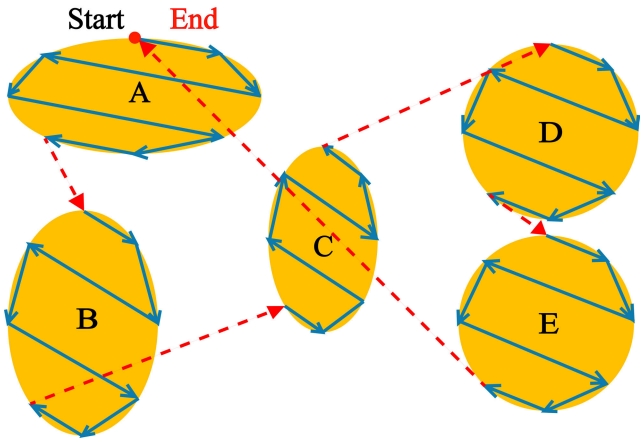


FIGURE 3. A feasible path of GR in worn segment 1.

Completing the gluing work in the shortest path will improve work efficiency of GR. The essence of POGR is to get the shortest path which GR walks among worn blocks in the certain worn segment. The red dotted line in Fig. 3 is a feasible optimization path.

IV. MODELING

POGR is very similar to RPP problem. RPP is to find a loop with the shortest distance, when a postman passes through all edges and each edge is passed only once in the graph [46]. It belongs to the combinatorial optimization problems. According to the principle of graph theory, in an undirected graph $G(V, E, \omega)$, where $V = \{V_1, V_2, \ldots V_n\}$ represents a collection of points, E represents the collection of edges in the graph, ω represents the weight of the edges (the actual length of GR's walking path, including the length in each worn block and the length among worn blocks).

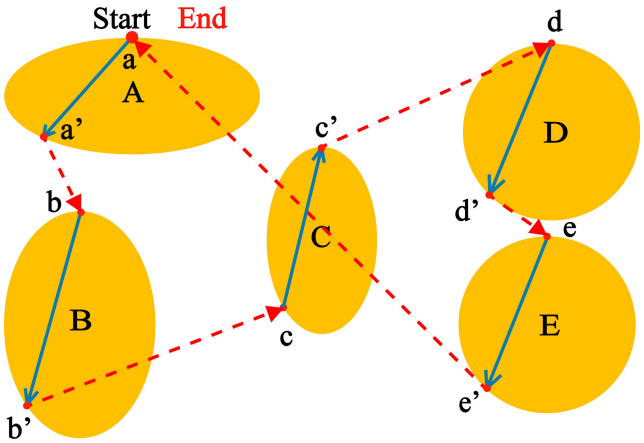


FIGURE 4. The schematic diagram of a feasible path.

Because of the characteristic of POGR, the path in every worn block can be seen as a fixed value. To simplify the problem, in Fig. 3, consider the path in each worn block as an edge, as shown in the blue solid line in Fig. 4. For example, the walking path of worn block A is represented by aa', the walking path of worn block B is represented by bb', the walking path of worn block C is represented by cc', the walking path of worn block D is represented by dd', and the walking path of worn block E is represented by ee'. The paths among worn blocks are shown in the red dotted-line.

As shown in Fig. 4, the shortest distance $D = \{d_{ij}\}$ $(i = 1, 2, \ldots n; j = 1, 2 \ldots n)$, between vertex i and vertex j, d_{ij} is calculated according to the Floyd. Set D is composed of the shortest distance between vertex i and vertex j that are not on one edge. In practice, the distance between two points is calculated by the Euclidean distance formula as (2). Before executing, these distances of all edges have calculated and stored as a distance matrix.

$$d_{ij} = \left((x_i - x_j)^2 + (y_i - y_j)^2\right)^{1/2} \tag{2}$$

The Hamilton graph proposed by Kang *et al.* [47] represents the RPP problem. So does POGR. As shown in Fig. 5, a vertex is used to represent a solid-line edge of a worn block in Fig. 4. Also, the edge sets the initial direction. That is, point 1 represents aa' (worn block A), point 2 represents bb' (worn block B), point 3 represents cc' (worn block C), point 4 represents dd' (worn block D), and point 5 represents ee' (worn block E). At the same time, the POGR is transformed into a "double vertices" TSP problem. For a TSP problem,

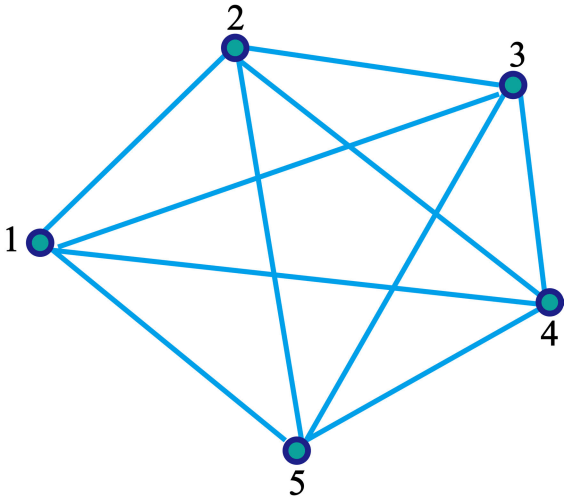


FIGURE 5. The Hamilton graph of the POGR.

there are several paths to traverse the undirected graph from one of the cities and go back to the starting city. How to find the shortest one from these paths is the TSP problem [20]. So does POGR. POGR problem in Fig. 5 is to find the shortest path that passes through five vertices and each vertex passes only once.

POGR is a new variant of the TSP. POGR is a ''double vertex'' TSP— In Fig. 5, each vertex represents a worn block shown in Fig. 4. Each worn block has a starting point and an ending point. Hence, each vertex in Fig. 5 has two points inside it. So POGR is a ''double vertex'' TSP. When repairing each worn block, the starting point and the ending point are interchangeable. Take the worn block *A* in Fig. 4 as an example. The path *aa*' is the GR's initial path. *a* is the starting point and *a*' is the ending point. However, when the GR works, the actual path may be *a*'*a*, *a*' is the starting point and a is the ending point. The starting point and the ending point are interchanged in the actual repairing, which provides the possibility for the path optimization. For example, from worn block *A* to worn block *B*, worn block *A* has two points *a* and *a*', worn block *B* has two points *b* and *b*'. There are four possible paths between the two worn blocks. In Fig. 5, the weight of each edge is not fixed, and its value is determined by the selection of the starting point of each worn block in Fig. 4. Actually, POGR is also a symmetrical TSP—the distances between the two points do not rely on the trajectory direction. For example, d_{ab} represents the distance between a and b, and d_{ba} represents the distance between b and a. When travelling from *a* to *b*, that is d_{ab}. When travelling from *b* to *a*, that is d_{ba}. If $d_{\mathrm{ab}} = d_{\mathrm{ba}}$, it is symmetrical. And in POGR, $d_{\mathrm{ab}} = d_{\mathrm{ba}}$, so POGR is symmetrical TSP.

Assuming in the Hamiltonian *G*' (*V*', *E*', *ω*'), where *V*' represents the set of vertices, *E*' represents the set of edges, *ω*' represents the weight of the edges and *ω*'∈ *D*. In *G*', the order of the vertex set *V*' is

$$T = (t_1, t_2, \ldots t_n), \quad ti \in V'. \tag{3}$$

Definition 1: In the Hamilton graph *G*', starting from any vertex, all the other vertices are visited and each vertex is visited only once. And then go back to the starting point. The sequence of vertices formed is defined as *ab*. *dist*, which is a feasible approach to the POGR problem.

Definition 2: In the Hamilton graph *G*', starting from any vertex, visiting all the other vertices, each vertex is visited only once. And then go back to the starting point. The vertex sequence of the shortest path is defined as *ag*. *dist*, which is the optimal approach of POGR. That is $minT_d$.

$$\min T_d = \sum_{i=1}^{n} \omega_i + \sum_{j=1}^{n} \omega_j' \tag{4}$$

Among them, ω_{i} represents the length of path in each worn block that the GR passed, which is considered to be a fixed value in the POGR; ω_{j}' represents the length of path among worn blocks that GR passed. POGR can be simplified to:

$$\min T_d = \sum_{j=1}^{n} \omega_j' \tag{5}$$

V. IGA-POGR ALGORITHM

Since the TSP is an NP-hard problem, exact programming algorithms can merely find an optimal solution for the instances due to limited computational resource. GA has shown its superiority in convenient modeling and easy implementation, meanwhile, found a satisfactory solution [48]. In the light of the characteristics of the POGR, a novel algorithm called IGA-POGR is proposed. Considering the uncertainty of the number of worn blocks in the worn segment, the operator in the IGA-POGR selects different methods according to the scale of the problem. It uses the same genetic operators but different methods and different parameters to solve the POGR. The genetic operators of IGA-POGR are selection operator, crossover operator, and partheno-genetic operator. If the scale is greater than or equal to 20, selection operator selects Liner Rank-based Roulette Wheel Selection (RB-RWS), crossover operator selects the PMX and partheno-genetic operator selects Exchange Order of Multipoint(EOM) [44], [45]; if the scale is less than 20, selection operator selects Tournament Selection(TS), crossover operator selects PMX and partheno-genetic operator selects Exchange Order of Single-point(EOS) [45], [46]. In this way, the efficiency of the IGA-POGR and the ability to find the global optimal approach are greatly enhanced.

A. ENCODING

Whenever, we apply GA for TSP, it runs over thousands of chromosome, each represents a solution. The chromosome needs encoding, when the IGA solves POGR. Since the POGR is a ''double vertices'' TSP problem, two sets of chromosome encodings are used.

In the Hamilton graph, which represents a POGR problem, each vertex represents an edge. The edge which represents a certain worn block includes the starting point and the ending point. Both the starting point and the ending point are initial settings. In order to express not only the vertex information but also the direction of the edge that represented by the

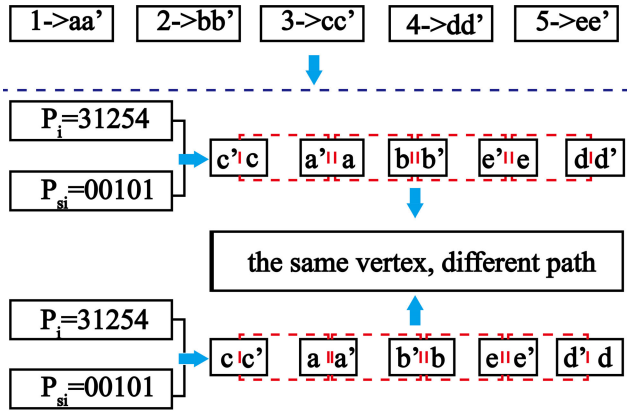


FIGURE 6. The same vertex, different path.

vertex, two sets of chromosome encodings are proposed. P_i as a real number encoding represents the vertex information [20] (path representation), in addition, P_{si} as a binary number encoding represents the direction of each edge. The length of the chromosome encoding is $|V'|$.For example, when $|V'| = 5$, two individual encoding $P_i = 31254$, $P_{si} = 00101$, and encoding $P_i = 31254$, $P_{si} = 11010$ are randomly generated. In P_{si}, 1 means the direction of the edge, which is the same as the initial setting; 0 means the direction of the edge, which is opposite to the initial setting. As we can see from Fig. 6, everything seemed same-same vertex, same path, because of the same P_i (31254). But actually, because of the different P_{si} (00101,11010), there are different paths: $c'c \rightarrow a'a \rightarrow bb' \rightarrow e'e \rightarrow dd'$ *and* $cc' \rightarrow aa' \rightarrow b'b \rightarrow ee' \rightarrow d'd$.

B. DETERMINE THE FITNESS FUNCTION

Fitness function measures the matching degree between feasible approach and optimal approach. The value of fitness function indicates the distance between the two. In this paper, the fitness function adopts a distance-based matching idea [49].

$$f_{dist}(ab, ag) = {}^{1}/_{(ab.dist - ag.dist)} \tag{6}$$

$$T = (\sum_{i=1}^{n} \sum_{j=1}^{n} d(i,j))/2n \tag{7}$$

Since *ag.dist* is unknown, a parameter T is given according to POGR, which represents the optimal approach. In (6), *d(i, j)* is the distance between vertex *i* and vertex *j*. The value of *d(i, j)* is obtained from set D or (2). Obviously, $T \leq$ ag.dist. When *ab.dist* is closer to T, it must be closer to *ag.dist*.

Formula of fitness function:

$$f_{dist}(ab, ag) = {}^{1}/_{(ab.dist - T)} \tag{8}$$

C. GENETIC OPERATOR

The genetic operators of IGA-POGR are selection operator, crossover operator, and partheno-genetic operator. POGR has the characteristics of the uncertainty number of worn blocks in each worn segment. Through the investigation of the worn belt surface in the coal mines of Shanxi, China, combined with GR's workspace, the POGR problem is divided into small-scale POGR and large-scale POGR. The number 20 is the dividing line. If the scale N is greater than or equal to 20, the POGR is the large-scale POGR. If the scale N is less than 20, the POGR is the small-scale POGR. For solving POGR problem efficiently and obtaining the global optimal approach, genetic operators select different methods according to the scale of POGR. As shown in Fig. 7, if the scale is greater than or equal to 20, selection operator selects Liner Rank-based Roulette Wheel Selection (RB-RWS), crossover operator selects the partially matched crossover (PMX) [41], [42] and partheno-genetic operator selects Exchange Order of Multipoint(EOM) [44], [45]; if the scale is less than 20, selection operator selects Tournament Selection (TS), crossover operator selects PMX and partheno-genetic operator selects Exchange Order of Single-point(EOS) [44], [45].

D. THE FLOWCHART OF IGA-POGR

The flow chart of the IGA-POGR is shown in Fig. 7.

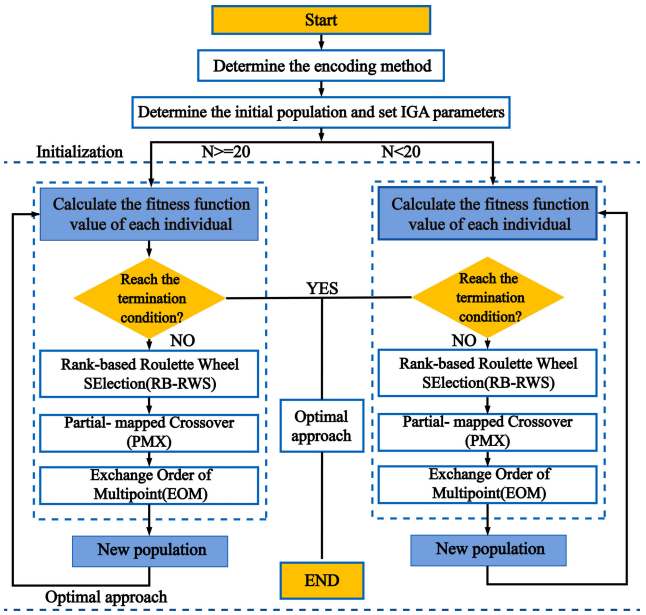


FIGURE 7. The flowchart of IGA- POGR.

The steps of IGA-POGR:

Step 1: The initial population $S1$ is generated. The larger the problem is, the bigger the initial population $S1$ becomes. It's well known that large population size needs more computation search time in finding an optimal or near optimal solution, but compared with small population, it has larger search space and good solutions can be easily got. So it is essential to choose different number of population according to the scales of the problem.

Step 2: According to the fitness function, calculate the fitness function value of each chromosome;

Step 3: Determine whether the loop termination condition is satisfied. That is, whether the iteration meets the pre-set value, or when the highest fitness chromosome no longer grows or grows slowly, the loop ends. Otherwise, go to Step 4;

Step 4: Selection operator: generate population *S2*. If $N \geq 20$, select the method of RB-RWS. If $N < 20$, select the method of TS. As can be seen from the optimization works [36], [37], the roulette wheel method is more preferred for maximization problems. In the minimization problems, the tournament selection method is more efficient method;

Step 5: Crossover operator: The operators which define the child production process are called the crossover operator. Crossover should increase the average quality of the population. According to certain rate of crossover (P_c) and crossover method, new chromosomes are generated; IGA-POGR selects the method of PMX;

Step 6: Partheno-genetic operator: It is needed to explore new states and helps the algorithm to avoid local optima. According to certain rate of mutation (P_{m}) and mutation method, new chromosomes are generated; If $N \geq 20$, P_{si} which is number encoding select the method of EOM; If $N < 20$, P_i which is binary number select the method of EOS;

Step 7: Generate a new population *S3* by crossover and partheno-genetic operators, and return to Step 2. The performance of IGA-POGR largely depends on the crossover and partheno-genetic operators. Hence, adapting the suitable value of P_c and P_m is very important because it guides the search process and maintains the diversity in the population.

E. METHOD OF THE GENETIC OPERATOR

1) THE SCALE GREATER THAN OR EQUAL TO 20

a: SELECTION OPERATOR

RB-RWS is shown in Fig. 8.

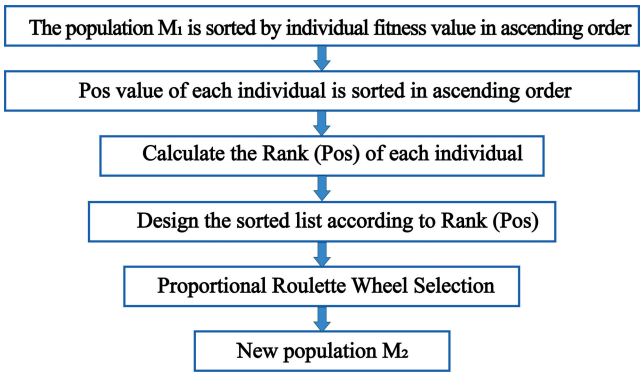


FIGURE 8. RB-RWS.

In the rank-based roulette wheel selection, the probability of a chromosome being selected is based upon its fitness rank relative to the entire population. RB-RWS is the selection strategy where the biasness could be under control through the selective pressure SP. It can avoid premature convergence and eliminate the need to scale fitness values. In RB-RWS, Rank (Pos) is defined as follow:

$$Rank(Pos) = 2 - SP + (2 \cdot (SP - 1) \cdot (Pos - 1)/(n - 1)) \tag{9}$$

In (9), n is the sum of individuals in the population, *Pos* is the position of an individual in the population (if $Pos = 1$, it represents the least fit individual; if $Pos = n$, it represents the fittest individual), SP is the selective pressure.

b: CROSSOVER OPERATOR

Usually, one-point crossover is selected. First, randomly choose one crossover point and then recombine the gene fragment of a pair of chromosomes to form two new chromosomes. The one-point crossover is suitable for the random keys encoding, but in order to preserve the permutation when dealing with natural encoding, another special crossover operators must be used.

As the literature [41], [42] says, PMX tries to preserve the absolute positions of the integers. The PMX needs two crossover points. *A* and *B* as two parent chromosomes, offspring chromosome *A* will inherit the sub-sequence outside these two points from *A* and offspring chromosome *B* will inherit the respective sub-sequence from *B*. Between the two points, the genes of *A* and *B* are copied from the other parent chromosome. The PMX operator tries to preserve their position as seen in Fig. 9.

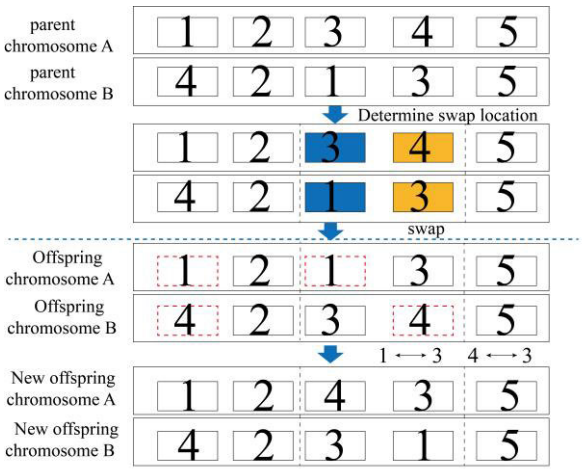


FIGURE 9. PMX.

After the yielding of offspring chromosome *A* and offspring chromosome B, for those genes between the two crossover points, if the values already present in the sequence, components are assigned through a procedure based on the mapping $1 \leftrightarrow 3$, $4 \leftrightarrow 3$. As for the third position of *A*, the value 1 (from *B*) cannot be assigned to it, because it is already present in the place of 1, but 3 is already present in the place of 4, which is not yet present ($1 \rightarrow 3 \rightarrow 4$). The number 4 can therefore be assigned to position 3 of *A*. Then, new offspring chromosome *A* (12 |43 |5), new offspring chromosome *B* (42 |31 |5). The crossover method of P_{i} selects PMX, P_{si} doesn't change.

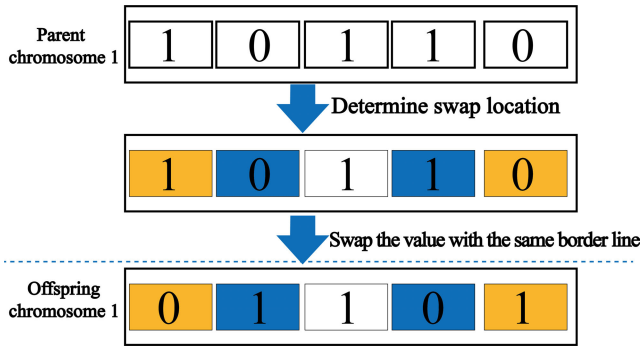


FIGURE 10. Schematic diagram of EOM.

c: PARTHENO-GENETIC OPERATOR

Partheno-genetic operators are used to implement the function of mutation, and the operators are executed on one chromosome. The partheno-genetic operators used in POGR only change the order of the chromosome but never change the numbers of 0 and 1. EOM implementation process is as follows in Fig. 10, chromosome (10110), random positions are 1 and 5, 2 and 4. In swap operator, the new chromosome is (01101). From the example we can see that the operator just changes the initial direction of some edges in the POGR. P_{si} selects EOM, P_i doesn't change.

2) THE SCALE LESS THAN 20

a: SELECTION OPERATOR

The algorithm of TS is shown in Fig. 11. TS is the selection strategy where r chromosomes are randomly chosen from the population and copy the best one from this group into the intermediate population, and repeat N times.

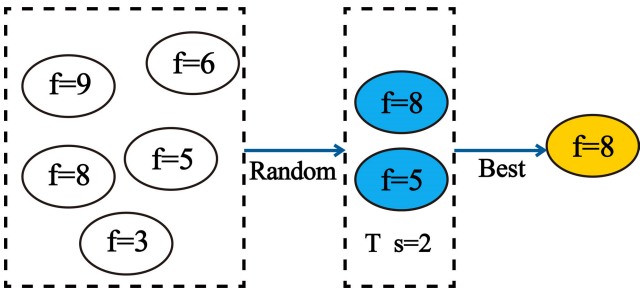


FIGURE 11. Schematic diagram of TS.

b: CROSSOVER OPERATOR

The crossover method of P_i selects PMX, which is shown in Fig. 9. P_{si} doesn't change.

c: PARTHENO-GENETIC OPERATOR

Select two genes from the chromosome randomly, and swap the position of the two genes to generate a new chromosome. P_{si} selects EOS, which is shown in Fig. 12. P_i doesn't change.

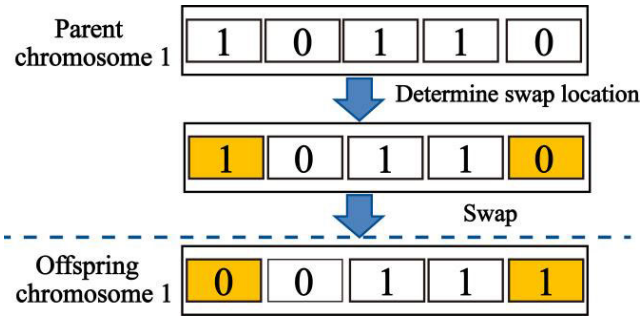


FIGURE 12. Schematic diagram of EOS.

VI. EXPERIMENTAL RESULTS AND DISCUSSIONS

The POGR problem is divided into small-scale POGR and large-scale POGR. The number 20 is the dividing line. IGA-POGR algorithm chooses different operators according to the scale of the POGR. In this Section, a number of well-known TSP instances are selected to evaluate the performance of the proposed IGA-POGR algorithm. For the verification of the performance of the IGA-POGR algorithm, well-known TSP instances are employed from TPSLIB library [50]. Burma14 and ulysses16 were chosen as examples, the scales of them are less than 20; ulysses22 and dantzig42 were chosen, because the scales of them are more than 20.

Next, IGA-POGR algorithm is used to the POGR problem. The collection of POGR consists of four idealized worn segments: 5 worn blocks (including ten coordinate points), 8 worn blocks (including sixteen coordinate points), 11 worn blocks (including twenty-two coordinate points), 14 worn blocks (including twenty-eight coordinate points). This collection has been solved by IGA-POGR algorithm effectively.

At last, comparison between IGA-POGR and DPSO are made in the essay. The experiments showed that both IGA-POGR and DPSO method are good when dealing with the small-scale POGR. The obtained results also show that the IGA-POGR has better performance when meeting large-scale POGR.

A. EXPERIMENTAL SET-UP

The IGA-POGR algorithm is coded in Python Version 3.8. One objective of the experiment is to investigate the performance of IGA-POGR with different methods in terms of number of generations to come out with the optimal approach for several well-known TSP instances. Another objective of the experiment is to obtain the optimal approach of the POGR problem by IGA-POGR algorithm. Also, comparison between IGA-POGR and DPSO are made in the essay.

There are two kinds of instances. One is the well-known optimal approach TSP instances: burma14, ulysses16, ulysses22 and dantzig42. The other is idealized four worn segments which respectively consist of 5 worn blocks, 8 worn blocks, 11 worn blocks and 14 worn blocks. The four worn segments with different worn blocks are the POGR problem. Whether the well-known TSP problem or the POGR problem, if the scale is less than 20, the selection operator

of IGA-POGR algorithm selects TS, the crossover operator selects PMX and the partheno-genetic operator selects EOS. In the TS, the tournament size is set to 2. If the scale is greater than or equal to 20, the selection operator of IGA-POGR algorithm selects RB-RWS, the crossover operator selects PMX and the partheno-genetic operator selects EOM. While in the RB-RWS, $SP = 1.1$ [36].

It is difficult to choose suitable values for parameters such as population size, rate of crossover (P_c), and rate of mutation (P_m) in building a practical IGA-POGR. For solving the POGR problems, in order to converge effectively and achieve global optima, firstly, we obey De Jong's suggestion [51] and Goldberg's idea [52]. De Jong first systematically studied the effects of different combinations of parameters. These parameters have been used in many GA implementations [53]. Goldberg suggested that GA may work well with a large crossover rate and with a small mutation rate. Grefenstette [54] provided more details and suggested that when meeting the large population size, high crossover rates with low mutation rates was good; however, in the small population size, the mutation rates become important in order to provide diversity and to increase the search quality. Also, Lawler [55] and Johnson [56] described how a population of medium quality can be created.

In the experiments presented here the following parameters have been established: If the scale is greater than or equal to 20, the size of population (800), the rate of mutation ($P_m = 0.03$) and the rate of crossover ($P_c = 0.8$); If the scale is less than 20, the size of population (=300), the mutation rate ($P_m = 0.1$) and the crossover rate ($P_c = 0.5$). Any change in the value of these parameters (increasing or decreasing) affects the result of GA negatively or positively [57]. Choosing the right parameters is a nontrivial task. In each experiment, the IGA-POGR algorithm was run thirty times, and the lowest distance as a final result is chosen. In all experiments, when the condition is met, the loop ends.

TABLE 1. Results of the best approach for known instances.

Instances	The number of cities	IGA	Known optimal approach
burma14	14	30.8785	30.8785
ulysses16	16	6859	6859
ulysses22	22	7013	7013
dantzig42	42	679	679

B. RESULTS AND DISCUSSION

1) IGA-POGR ALGORITHM RESULTS FOR THE WELL-KNOW TSP INSTANCES

The best approaches obtained for the well-known TSP instances with IGA-POGR algorithm are showed in Table 1. It clearly shows that IGA-POGR can solve well-known TSP problems—whether the scale is greater than 20 or less than 20, an optimal approach can be obtained. Numerical results in Table 1 show that the proposed IGA-POGR does not give any deviation (0%) from the optimal solution for the four instances: burma14, ulysses16, ulysses 22 and dantzig 42.

In Fig. 13, the performance curves show the minimum distance obtained by the IGA-POGR in each generation. As we can see from the curves, along with the generation increasing, the distance reduced towards optimal approach and finally converged at a certain generation. For example, in burma14 and dantzig 42, the IGA-POGR converged at generation 21 and 145 respectively, where there is no more improvement made after this generation. It proves that IGA-POGR is feasible and valid.

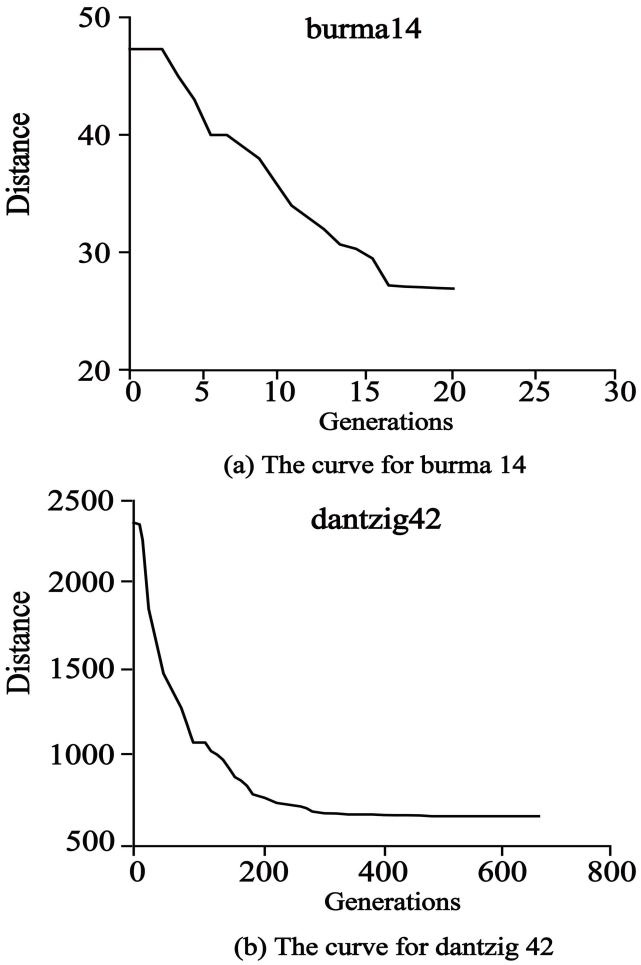


FIGURE 13. Performance curves for instances showing number of generations to converge.

For the well-known TSP instances which are mentioned above, when mutation rate and crossover rate are fixed, the size of population is exchanged. Result shows that large population size need more time in finding an optimal solution. With small population, it can easily fall into the local optimal solution. Improving mutation rate can prevent premature convergence.

2) IGA-POGR ALGORITHM RESULTS FOR THE POGR PROBLEM

Taking a belt with a 1400cm width for instance, Fig. 14 is the schematic diagram of the worn belt. It contains four different worn segments. The X axis represents the width of the belt. Obviously, the width is 1400cm in this experiment. And the Y axis represents the length of GR's workspace, the length is designed to be 3000 cm. The blue rectangular block diagram represents each worn block. The edges of the blue rectangular block are the edges of the worn block. The size of the blue rectangular block is the size of the worn block. In each worn block, there are a starting point and an ending point. The starting point and ending point represented by the red boxes are the initial setting. For the POGR problem, the starting point and the ending point are interchangeable.

In order to obtain the optimal path among worn blocks, first, the worn blocks and the starting and ending points are marked, as shown in Fig. 15. Each worn block has its own number tag on it. The starting point and the ending point corresponding to each worn block have a number tag starting with the number tag of the worn block. Take "Five Worn Blocks" as an example, 1,2,3,4 and 5 are respectively as the number tags for each worn block. For the worn block 1, 11 is the starting point of the initial setting and 12 is the ending point. For the worn block 2, 21 is the starting point of the initial setting and 22 is the ending point. For the worn block 3, 31 is the starting point of the initial setting and 32 is the ending point. For the worn block 4, 41 is the starting point of the initial setting and 42 is the ending point. For the worn block 5, 51 is the starting point of the initial setting and 52 is the ending point. Thus, for the POGR problem with 5 worn blocks, it is equivalent to the TSP problem with a scale of 10. Then IGA-POGR is executed. Based on the IGA-POGR algorithm, if the scale is less than 20(5 worn blocks, 10 points), the selection operator of the IGA-POGR algorithm selects TS, crossover operator selects PMX and partheno-genetic operator selects EOS. The optimization path of the "Five Worn Blocks" that calculated by the IGA-POGR algorithm can be seen in Table 2.

All the optimization paths of the POGR problems are shown in Table 2. As we can see from Table 2, after the optimization calculation of the IGA-POGR algorithm, not only the paths among the worn blocks are changed, but also the starting point and the ending point of some worn blocks are exchanged. Take "Five Worn Blocks" as an example, the initial setting path among the worn blocks is 1→2→3→4→5.Actually, the path GR walks among the worn blocks is 1→3→2→5→4. The initial setting path including the starting point and the ending point among the worn blocks is 11→12→21→22→31→32→41→42→51→52.Actually, the path GR walks among the worn blocks including the starting point and the ending point is 11→12→31→32→22→21→41→42→52→51. Based on the results of the computational experiments about the POGR problem, a conclusion may be formulated that IGA can solve

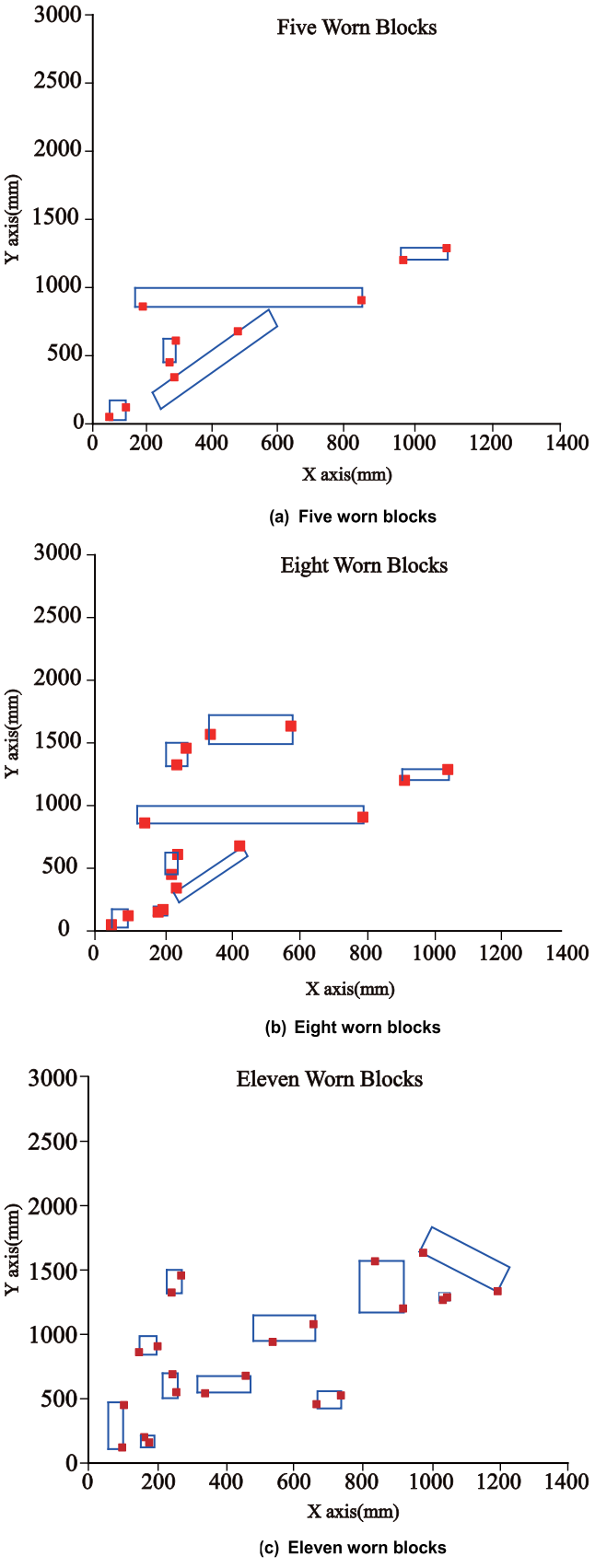
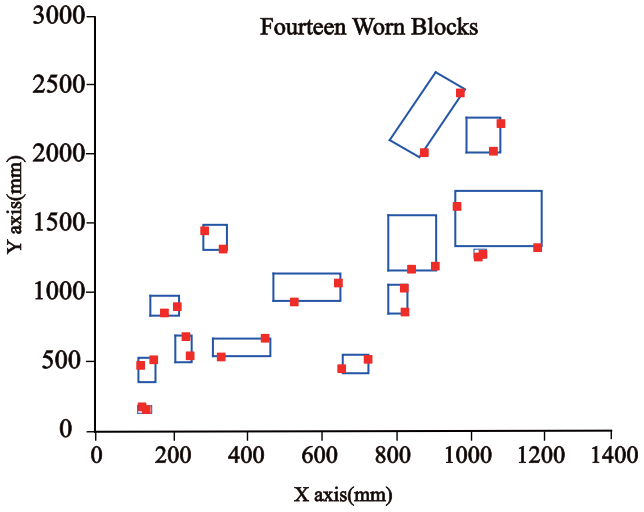


FIGURE 14. The worn segments of 1400cm wide belt.



(d) Fourteen worn blocks

FIGURE 14. *(Continued.)* **The worn segments of 1400cm wide belt.**

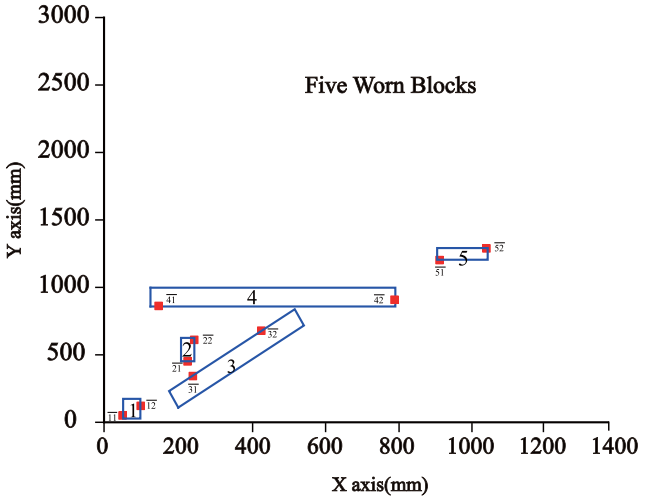
TABLE 2. **The optimal path of the GR.**

The number of worn blocks	The optimal path of the GR	The optimal path of the GR including the starting and ending point
5	1->3->2->4->5	[11,12,31,32,22,21, 41,42, 52,51]
8	5->6->7->4->2->3->8->1	[52,51,62,61,72,71, 41,42,22,21,31,32, 82,81,12,11]
11	1->2->3->11->10->4->6->5->7->8->9	[11,12,22,21,31,32, 11A,11B,10A,10B, 41,42,61,62,51,52, 71,72,81,82,92,91]
14	1->2->3->4->5->8->7->6->9->10->14->11->13->12	[11,12,21,22,31,32, 42,41,51,52,82,81, 71,72,61,62,91,92, 10A,10B,14A,14B, 11B,11A,13A,13B,12B,12A]

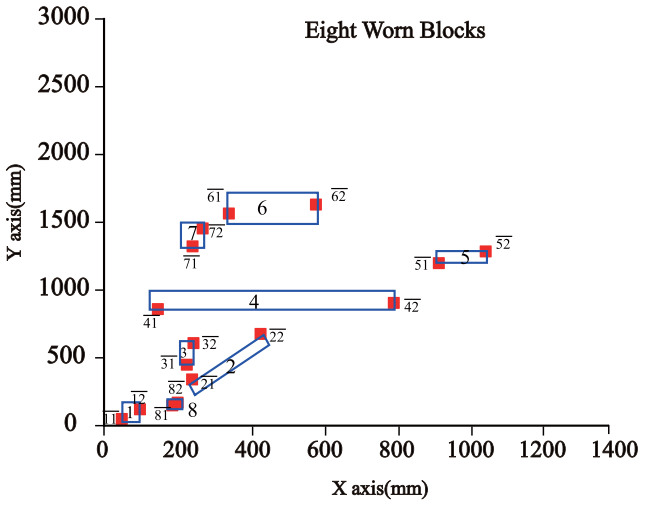
the POGR problem, find the optimal path for the GR, shorten the repair time, and improve the working efficiency. In the process of the repairing automatically with GR, every second saved is invaluable. GR will repair more worn belt surface in the limited time. This is certainly significant throughout the maintenance.

3) COMPARISON BETWEEN IGA-POGR AND DPSO

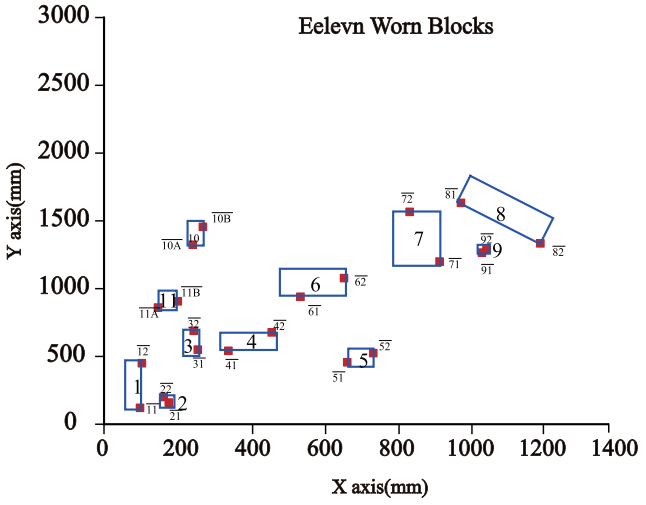
In recent years, particle swarm optimization (PSO) has attracted the interest of researchers due to their simplicity, effectiveness and efficiency in solving complex optimization problems [58]. There have been some reported works focusing on TSP problem. Discrete particle swarm optimization (DPSO) is proposed by Clerc for solving TSP problem



(a)Five worn blocks with the starting and the ending point

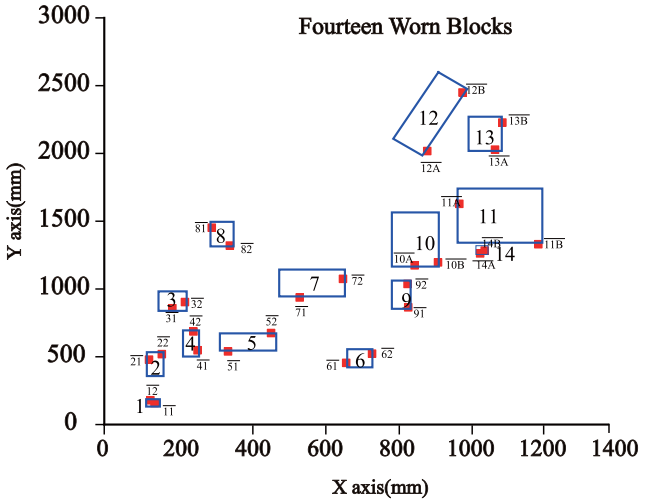


(b)Eight worn blocks with the starting and the ending point



(c)Eleven worn blocks with the starting and the ending point

FIGURE 15. **Marked graphs of worn blocks with the starting and ending Points.**



(d)Fourteen worn blocks with the starting and the ending point

FIGURE 15. *(Continued.)* **Marked graphs of worn blocks with the starting and ending Points.**

TABLE 3. Comparison between IGA-POGR and DPSO.

The number of worn blocks	Best length(cm)		Execution Time(sec)	
	IGA-POGR	DPSO	IGA-POGR	DPSO
5	2740	2740	0.5	0.51
8	4097	4097	0.67	0.69
10	5930	6057	0.9	1
14	6508	6528	1.3	1.5

with the definition of velocity as a ''swap sequence'' as well as other variables and rules, and it achieved good results [59]. We compare the proposed IGA-POGR with DPSO, and the results are showed in Table 3. The configurations of IGA-POGR and DPSO are the same.

According to the characteristic of the POGR, it is a ''double vertices'' issue. It means that the number of the scale is twice as much as the number of worn blocks. Thus when the number of worn blocks is 5 and 8, it is the small-scale POGR, the size of population is 300; when the number of worn blocks is 10 and 14, it is the large-scale POGR, the size of population is 800.

The experiments showed that both IGA-POGR and DPSO method can be used to solve the POGR effectively when dealing with the small-scale POGR. The numerical results in Table 3 also show that the IGA-POGR has better performance in terms of the solving quality and time consumption when using large-scale POGR. According to the results, the proposed IGA-POGR algorithm is better in comparison with DPSO in solving the complex POGR problem.

VII. CONCLUSION AND FUTURE WORK

The worn belt surface has acceleration characteristics, so to repair the worn belt surface quickly and automatically is the core design objective of the gluing robot (GR).Based on this objective, a new variant Traveling Salesman Problem (''double vertices'' TSP) is put forward: which is called POGR (path optimization of the gluing robot). And the mathematical model is built. IGA(improved genetic algorithm) is used to solve the POGR (IGA-POGR). Selection operator, crossover operator and partheno-genetic operator are applied in IGA-POGR algorithm. The genetic operator selects different methods for different scales of the POGR. Four well-known TSP instances and four idealized POGR problems are used to verify the performance of IGA-POGR. Numerical results show that IGA-POGR does not give any deviation (0%) from the known optimal solution. Compared with DPSO (discrete particle swarm optimization), IGA-POGR has better performance in terms of the quality of solutions and time consumption when solving four idealized POGR problems.

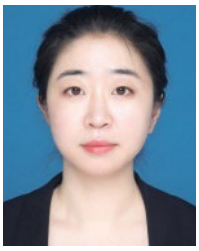
Moreover, the further research of GR, for example, is that GR starts from the specific point, and then returns to the very same point. It is very important to solve the POGR better with other algorithms as well. Furthermore, to determine the threshold of the worn blocks according to the user's demand and calculate the volume of the worn blocks accurately, are necessary for GR.

In China, steel cord conveyor belt constitute an increasing share of the conveyor belts operated in both underground and open pit mines. If mines or conveyor belt service companies were interested, GR could be widely implemented in mining industry. Using the GR to repair the worn belt surface can improve the repair efficiency, reduces the number of unexpected failures, eliminate the hidden trouble of the steel core exposed and corroded, prolong the service life of the belt and ensure the safety of production. Thus GR contributes to improve conveyor belt management, and push on the development of the transportation system automation of intelligent mine.

REFERENCES

[1] G. Wang, Y. Xu, and H. Ren, ''Intelligent and ecological coal mining as well as clean utilization technology in China: Review and prospects,'' *Int. J. Mining Sci. Technol.*, vol. 29, no. 2, pp. 161–169, Mar. 2019, doi: 10.1016/j.ijmst.2018.06.005.
[2] X. Ge, ''Smart mine construction based on knowledge engineering and Internet of Things,'' *Int. J. Performability Eng.*, vol. 14, no. 5, pp. 1060–1068, 2019, doi: 10.23940/ijpe.18.05.p25.10601068.
[3] L. Ma, ''Study on intelligent mine based on the application of 5G wireless communication system,'' *IOP Conf. Ser., Earth Environ. Sci.*, vol. 558, no. 3, 2020, Art. no. 032050.
[4] J. Wang and Z. Huang, ''The recent technological development of intelligent mining in China,'' *Engineering*, vol. 3, no. 4, pp. 439–444, 2017.
[5] Y. Wang, P. Tian, Y. Zhou, and Q. Chen, ''The encountered problems and solutions in the development of coal mine rescue robot,'' *J. Robot.*, vol. 2018, pp. 1–11, Oct. 2018.
[6] W. Wang, W. Dong, Y. Su, D. Wu, and Z. Du, ''Development of search-and-rescue robots for underground coal mine applications,'' *J. Field Robot.*, vol. 31, no. 3, pp. 386–407, May 2014.
[7] J. Szrek, J. Wodecki, R. Błażej, and R. Zimroz, ''An inspection robot for belt conveyor maintenance in underground mine—Infrared thermography for overheated idlers detection,'' *Appl. Sci.*, vol. 10, no. 14, p. 4984, Jul. 2020.
[8] Y. Li and H. Zhu, ''A simple optimization method for the design of a lightweight, explosion-proof housing for a coal mine rescue robot,'' *J. Brazilian Soc. Mech. Sci. Eng.*, vol. 40, no. 7, pp. 1–10, Jul. 2018.
[9] M. Li, H. Zhu, S. You, and C. Tang, ''UWB-based localization system aided with inertial sensor for underground coal mine applications,'' *IEEE Sensors J.*, vol. 20, no. 12, pp. 6652–6669, Jun. 2020.

[10] Y. Yang, C. Miao, X. Li, and X. Mei, "On-line conveyor belts inspection based on machine vision," *Optik*, vol. 125, no. 19, pp. 5803–5807, Oct. 2014.
[11] G. Fedorko, V. Molnar, D. Marasova, A. Grincova, M. Dovica, J. Zivcak, T. Toth, and N. Husakova, "Failure analysis of belt conveyor damage caused by the falling material. Part I: Experimental measurements and regression models," *Eng. Failure Anal.*, vol. 36, pp. 30–38, Jan. 2014.
[12] G. Fedorko, V. Molnár, Ž. Ferková, P. Peterka, J. Krešák, and M. Tomašková, "Possibilities of failure analysis for steel cord conveyor belts using knowledge obtained from non-destructive testing of steel ropes," *Eng. Failure Anal.*, vol. 67, pp. 33–45, Sep. 2016.
[13] M. Bajda and M. Hardygóra, "The influence of natural ageing processes on the strength parameters of steel cord conveyor belts," *Int. J. Mining, Reclamation Environ.*, vol. 32, no. 6, pp. 430–439, Aug. 2018.
[14] R. Błażej, L. Jurdziak, T. Kozłowski, and A. Kirjanów, "The use of magnetic sensors in monitoring the condition of the core in steel cord conveyor belts—Tests of the measuring probe and the design of the DiagBelt system," *Measurement*, vol. 123, pp. 48–53, Jul. 2018.
[15] G.-X. Cui and H.-W. Zhang, "Design of monitoring system of conveyor belt of steel rope core based on X-ray," *Ind. Mine Autom.*, vol. 38, no. 4, pp. 70–72, 2012.
[16] X.-G. Li, C.-Y. Miao, Y. Zhang, and W. WANG, "Automatic fault detection for steel cord conveyor belt based on statistical features," *J. China Coal Soc.*, vol. 37, no. 7, pp. 1233–1238, 2012.
[17] A. Harrison, "New technique for measuring loss of adhesion in conveyor-belt splices," *Aust. J. Coal Min. Technol. Res., (Aust.)*, vol. 4, no. EDB-84-114310, pp. 27–34, 1983.
[18] A. Kirjanów-Błażej, L. Jurdziak, R. Burduk, and R. Błażej, "Forecast of the remaining lifetime of steel cord conveyor belts based on regression methods in damage analysis identified by subsequent DiagBelt scans," *Eng. Failure Anal.*, vol. 100, pp. 119–126, Jun. 2019.
[19] J. J. Grefenstette, "Genetic algorithms and machine learning," in *Proc. 6th Annu. Conf. Comput. Learn. Theory (COLT)*, 1993, pp. 3–4.
[20] F. Liu and G. Zeng, "Study of genetic algorithm with reinforcement learning to solve the TSP," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 6995–7001, Apr. 2009.
[21] J. Grefenstette, R. Gopal, B. Rosmaita, and D. Van Gucht, "Genetic algorithms for the traveling salesman problem," in *Proc. 1st Int. Conf. Genetic Algorithms Their Appl.*, vol. 160, no. 168: Lawrence Erlbaum, 1985, pp. 160–168.
[22] M. Albayrak and N. Allahverd, "Development a new mutation operator to solve the traveling salesman problem by aid of genetic algorithms," *Expert. Syst. Appl.*, vol. 38, no. 3, pp. 1313–1320, Mar. 2011.
[23] Y. Deng, Y. Liu, and D. Zhou, "An improved genetic algorithm with initial population strategy for symmetric TSP," *Math. Problems Eng.*, vol. 2015, pp. 1–6, Oct. 2015.
[24] P.-C. Chang, W.-H. Huang, and C.-J. Ting, "Dynamic diversity control in genetic algorithm for mining unsearched solution space in TSP problems," *Expert Syst. Appl.*, vol. 37, no. 3, pp. 1863–1878, Mar. 2010.
[25] K.-P. Wang, L. Huang, C.-G. Zhou, and W. Pang, "Particle swarm optimization for traveling salesman problem," in *Proc. Int. Conf. Mach. Learn. Cybern.*, vol. 3, Nov. 2003, pp. 1583–1585.
[26] T. Munakata and Y. Nakamura, "Temperature control for simulated annealing," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 64, no. 4, Sep. 2001, Art. no. 046127.
[27] L. Huang, C. Zhou, and K. Wang, "Hybrid ant colony algorithm for traveling salesman problem," *Prog. Natural Sci.*, vol. 13, no. 4, pp. 295–299, 2003.
[28] D. E. Goldberg and R. Lingle, "Alleles, loci, and the traveling salesman problem," in *Proc. Int. Conf. Genetic Algorithms Their Appl.*, vol. 154. Pittsburgh, PA, USA: Carnegie-Mellon Univ. Pittsburgh, 1985, pp. 154–159.
[29] H. M. Pandey, A. Chaudhary, and D. Mehrotra, "A comparative review of approaches to prevent premature convergence in GA," *Appl. Soft Comput.*, vol. 24, pp. 1047–1077, Nov. 2014.
[30] W. M. Jenkins, "Towards structural optimization via the genetic algorithm," *Comput. Struct.*, vol. 40, no. 5, pp. 1321–1327, Jan. 1991.
[31] W. Jenkins, "Structural optimization with the genetic algorithm," *Struct. Engineer*, vol. 69, no. 24, pp. 418–422, 1991.
[32] W. M. Jenkins, "Plane frame optimum design environment based on genetic algorithm," *J. Struct. Eng.*, vol. 118, no. 11, pp. 3103–3112, Nov. 1992.
[33] P. Larrañaga, C. M. H. Kuijpers, R. H. Murga, I. Inza, and S. Dizdarevic, "Genetic algorithms for the travelling salesman problem: A review of representations and operators," *Artif. Intell. Rev.*, vol. 13, no. 2, pp. 129–170, Apr. 1999.
[34] W. Hui, "Comparison of several intelligent algorithms for solving TSP problem in industrial engineering," *Syst. Eng. Procedia*, vol. 4, pp. 226–235, Jan. 2012.
[35] T. Yifei, Z. Meng, L. Jingwei, L. Dongbo, and W. Yulin, "Research on intelligent welding robot path optimization based on GA and PSO algorithms," *IEEE Access*, vol. 6, pp. 65397–65404, 2018.
[36] N. M. Razali and J. Geraghty, "Genetic algorithm performance with different selection strategies in solving TSP," in *Proc. World Congr. Eng.*, vol. 2, no. 1. Hong Kong: International Association of Engineers, 2011, pp. 1–6.
[37] H. Kılıç and U. Yüzgeç, "Improved antlion optimization algorithm via tournament selection and its application to parallel machine scheduling," *Comput. Ind. Eng.*, vol. 132, pp. 166–186, Jun. 2019.
[38] H. Stern, Y. Chassidim, and M. Zofi, "Multiagent visual area coverage using a new genetic algorithm selection scheme," *Eur. J. Oper. Res.*, vol. 175, no. 3, pp. 1890–1907, 2006.
[39] J. E. Baker, "Adaptive selection methods for genetic algorithms," in *Proc. Int. Conf. Genetic Algorithms Their Appl.*, vol. 1. Hillsdale, NJ, USA, 1985, pp. 101–111.
[40] B. L. Miller and D. E. Goldberg, "Genetic algorithms, tournament selection, and the effects of noise," *Complex Syst.*, vol. 9, no. 3, pp. 193–212, 1995.
[41] M. Moz and M. Vaz Pato, "A genetic algorithm approach to a nurse rerostering problem," *Comput. Oper. Res.*, vol. 34, no. 3, pp. 667–691, Mar. 2007.
[42] G. Üçoluk, "Genetic algorithm solution of the TSP avoiding special crossover and mutation," *Intell. Autom. Soft Comput.*, vol. 8, no. 3, pp. 265–272, Jan. 2002.
[43] W. Banzhaf, "The 'molecular' traveling salesman," *Biolog. Cybern.*, vol. 64, no. 1, pp. 7–14, 1990.
[44] L. Maojun and T. Tiaosheng, "A partheno-genetic algorithm and analysis on its global convergence," *Acta Automatica Sinica*, vol. 25, no. 1, pp. 68–72, 1999.
[45] F. Kang, J.-J. Li, and Q. Xu, "Virus coevolution partheno-genetic algorithms for optimal sensor placement," *Adv. Eng. Informat.*, vol. 22, no. 3, pp. 362–370, Jul. 2008.
[46] C. Cerrone, B. Dussault, X. Wang, B. Golden, and E. Wasil, "A two-stage solution approach for the directed rural postman problem with turn penalties," *Eur. J. Oper. Res.*, vol. 272, no. 2, pp. 754–765, Jan. 2019.
[47] M.-J. Kang and C.-G. Han, "Solving the rural postman problem using a genetic algorithm with a graph transformation," in *Proc. ACM Symp. Appl. Comput. (SAC)*, 1998, pp. 356–360.
[48] I. A. Abed, S. P. Koh, K. S. M. Sahari, P. Jagadeesh, and S. K. Tiong, "Optimization of the time of task scheduling for dual manipulators using a modified electromagnetism-like algorithm and genetic algorithm," *Arabian J. Sci. Eng.*, vol. 39, no. 8, pp. 6269–6285, Aug. 2014.
[49] N. K. Jerne, "Towards a network theory of the immune system," *Annu. Immunol.*, vol. 125, no. 3, pp. 373–389, 1974.
[50] G. Barksdale and W. D. B. Meyer, *TPS, a Text Processing System-Primer and Reference Manual*. School Monterey, CA, USA: Naval Postgraduate, 1976.
[51] K. A. De Jong and W. M. Spears, "Using genetic algorithms to solve NP-complete problems," in *Proc. ICGA*, 1989, pp. 124–132.
[52] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA, USA: Addison-Wesley, 1989.
[53] K. A. De Jong, *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*. Ann Arbor, MI, USA: Univ. Michigan, 1975.
[54] J. J. Grefenstette, "Optimization of control parameters for genetic algorithms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-16, no. 1, pp. 122–128, Jan. 1986.
[55] E. L. Lawler, "The traveling salesman problem: A guided tour of combinatorial optimization," in *Wiley-Interscience Series in Discrete Mathematics*. Hoboken, NJ, USA: Wiley, 1985.
[56] D. S. Johnson, "Local optimization and the traveling salesman problem," in *Proc. Int. Colloq. Automata, Lang., Program.* Berlin, Germany: Springer, 1990, pp. 446–461.
[57] L. Davis, *Handbook of Genetic Algorithms*. New York, NY, USA: Van Nostrand Reinhold, 1991.
[58] W. Elloumi, H. El Abed, A. Abraham, and A. M. Alimi, "A comparative study of the improvement of performance using a PSO modified by ACO applied to TSP," *Appl. Soft Comput.*, vol. 25, pp. 234–241, Dec. 2014.
[59] M. Clerc, "Discrete particle swarm optimization, illustrated by the traveling salesman problem," in *New Optimization Techniques in Engineering*. Berlin, Germany: Springer, 2004, pp. 219–239.



YUHANG ZHANG was born in Songyuan, Jilin, China, in 1981. She received the master's degree from Liaoning Technical University, in 2010. Her research interests include optimization algorithm and mathematical modeling. She is committed to research and solve the problems that may be encountered by the conveyor belts in industrial production.

ZILING SONG was born in Chifeng, Inner Mongolia, China, in 1965. He received the Ph.D. degree from Liaoning Technical University, in 2007. He is currently a Professor and a Ph.D. Supervisor. His research interests include mine environmental engineering and open-pin mine.

JING YUAN was born in Fuxin, Liaoning, China, in 1968. He received the Ph.D. degree in computer science and technology from Northeastern University, in 2008. He has been engaged in the research of computer graphics and image processing for many years. He did a profound and in-depth investigation with CHN Energy, Jinneng Holding Group, Yangmei Group, and the coal mines belonging to them. He summarized the problems existing in the operation of the mining steel cord conveyor belt and owned 15 authorized patents on the steel cord conveyor belt. At the same time, patent-backed products are also in production. As a member of the Coal Industry Coal Mine Safety Standardization Technology Committee, he has presided over two industry standards.

ZHIYUN DENG was born in Hubei, China, in 1990. He received the Ph.D. degree in geotechnical engineering from Chongqing University, Chongqing, China, in 2020. He is currently working with the State Key Laboratory of Hydroscience and Engineering, Tsinghua University, Beijing, China. His research interests include slope stability analysis, tunnel stability, pipe jacking engineering, and fiber reinforced concrete.

HAN DU was born in Liaoning, China, in 1990. He received the Ph.D. degree in mining engineering, in 2020. He is currently working with the State Key Laboratory of Hydroscience and Engineering, Tsinghua University, Beijing, China. His research interests include slope stability analysis, landslide early warning, and open pit mine underground reservoir construction.

LIDAN LI was born in Zhuanghe, Liaoning, China, in 1980. She received the Ph.D. degree from the School of Mathematical Sciences, Dalian University of Technology, in 2020. She is currently working with the College of Science, Liaoning Technical University, Fuxin, Liaoning. Her research interest includes inverse problems of continuous optimization.