

RESEARCH ARTICLE

# Multi-AGV path planning with double-path constraints by using an improved genetic algorithm

Zengliang Han<sup>1</sup>, Dongqing Wang<sup>1\*</sup>, Feng Liu<sup>2</sup>, Zhiyong Zhao<sup>1</sup>

**1** College of Automation and Electrical Engineering, Qingdao University, Qingdao, 266071, P.R. China, **2** Department of Industrial Engineering, University of Texas at Arlington, Arlington, TX 76019, United States of America

\* [dqwang64@163.com](mailto:dqwang64@163.com)



## Abstract

This paper investigates an improved genetic algorithm on multiple automated guided vehicle (multi-AGV) path planning. The innovations embody in two aspects. First, three-exchange crossover heuristic operators are used to produce more optimal offsprings for getting more information than with the traditional two-exchange crossover heuristic operators in the improved genetic algorithm. Second, double-path constraints of both minimizing the total path distance of all AGVs and minimizing single path distances of each AGV are exerted, gaining the optimal shortest total path distance. The simulation results show that the total path distance of all AGVs and the longest single AGV path distance are shortened by using the improved genetic algorithm.

## OPEN ACCESS

**Citation:** Han Z, Wang D, Liu F, Zhao Z (2017) Multi-AGV path planning with double-path constraints by using an improved genetic algorithm. PLoS ONE 12(7): e0181747. <https://doi.org/10.1371/journal.pone.0181747>

**Editor:** Xiaosong Hu, Chongqing University, CHINA

**Received:** May 14, 2017

**Accepted:** July 6, 2017

**Published:** July 26, 2017

**Copyright:** © 2017 Han et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

**Data Availability Statement:** All relevant data are within the paper and its Supporting Information files.

**Funding:** This research was supported by the National Natural Science Foundation of China (No. 61573205) and the Shandong Provincial Natural Science Foundation of China (No. ZR2015FM017).

**Competing interests:** The authors have declared that no competing interests exist.

## Introduction

Multi-objective optimization have been applied in many fields, including engineering [1–5], transportation [6–10] and logistics [11–15]. A multi-objective problem is to search for a solution that satisfies all objective functions between conflicting objectives. Multi-objective optimization methods include classical optimization algorithms (like weighted sum methods,  $\epsilon$ -constraint methods, interactive methods, Pareto-dominated methods) [16–19] and intelligent optimization methods (like evolution based algorithms and swarm based algorithms) [20–24].

Multiple automated guided vehicles (multi-AGVs), characterized by multi-objectives, are playing an increasingly important role in the area of distribution logistics due to their high efficiency for material handling among workstations. The applications of AGV systems face several important issues: AGVs number determining [25–27], path planning [28–30] and constraint exerting [31–33], etc.

Determining the optimal numbers of vehicles is the fundamental problem in the management of an AGV system. Several methodologies have been proposed to achieve this goal and their main objective is to attend all tasks on time with a sufficient numbers of vehicles [25–27, 34]. For example, Vivaldini et al. presented a new module to estimate the optimal numbers of AGVs for the execution of a set of tasks by integrating task assignment and routing [27]. Ji and

Xia built a new model for an AGV system and studied the minimum vehicle numbers by an approximately analytical method based on the binary search [34]. Koo et al. studied an AGV fleet size model where part waiting time is estimated for various vehicle dispatching rules to determine the proper AGV fleet size [26].

The multi-AGV path planning is most important in ensuring an efficient flow of materials during the production process. The path planning involves three issues in dispatching, scheduling and routing of tasks at the same time. The multi-AGV path planning problem [35–37] is similar to the traveling salesman problem (TSP) [38–40] in the aspect of finding the shortest tour/time which has extremely large search spaces and is very difficult to solve. Smolic-Rocak et al. used time windows in a vector form to solve the shortest path problem for multi-AGV systems [36]. Draganjac et al. implemented a shortest feasible path planning algorithm considering nonholonomic vehicle constraints for multi-AGV systems [35]. Wang et al. proposed a multi-offspring genetic algorithm for the TSP by producing excellent individuals [38]. Wang et al. investigated a novel memetic algorithm with a competitive capacity to maintain the total distance as short as possible for the TSP [39]. Jiang and Yan developed a discrete fruit fly optimization algorithm for the TSP [40].

There existing various constraints in multi-AGV path planning, e.g. collision-free constraints [41–43], time window constraints [36, 44], and time/distance constraints [35, 45]. This paper investigates an improve genetic algorithm for multi-AGV path planning by exerting double-path restrictions on both the total path distance of all AGVs and single path distances of each AGV, and by choosing three-exchange crossover heuristic operators for crossover operation.

Traditional genetic algorithms adopted two-exchange crossover operators for crossover operation, that is, using two parent individuals to produce a progeny chromosome [46–48]. Obviously, traditional genetic algorithms with two parent individuals other than more parent individuals would obtain less parent information, and reduce the diversity of offspring performance. In order to improve the diversity of progenies, we put forward the idea of three-exchange crossover operators for crossover operation, that is, using three parent individuals to produce a progeny chromosome.

The contributions of this paper lie in the follows:

- Unlike the traditional path constraint only exerting on the total path distance of all AGVs, this paper exerts double-path restrictions on both the total path distance of all AGVs and single path distances of each AGV. The strategy double shortened total/single AGV path distance and obtained the optimal result.
- By using three parent individuals other than the traditional method with two parent individuals, to produce a progeny chromosome, this method increases the information of producing the progeny chromosomes, earning the possibility of inheriting the excellent characteristics of the parents and accelerating the searching speed of the improved algorithm.
- Simulations on multi-AGV path diagrams and the iterative maps of the improved/traditional genetic algorithms verify that the improved genetic algorithm has the shorter total path distance of all AGVs than that of the traditional genetic algorithm.

The rest of the paper is organized as follows. Section 2 demonstrates the overview of facility layout. Section 3 investigates an improved genetic algorithm with double-path constraints by using the three-exchange crossover heuristic operators. Section 4 provides the feasibility of the algorithm by simulation. Finally, the concluding remarks are involved in Section 5.

## Overview of facility layout

### Facility layout

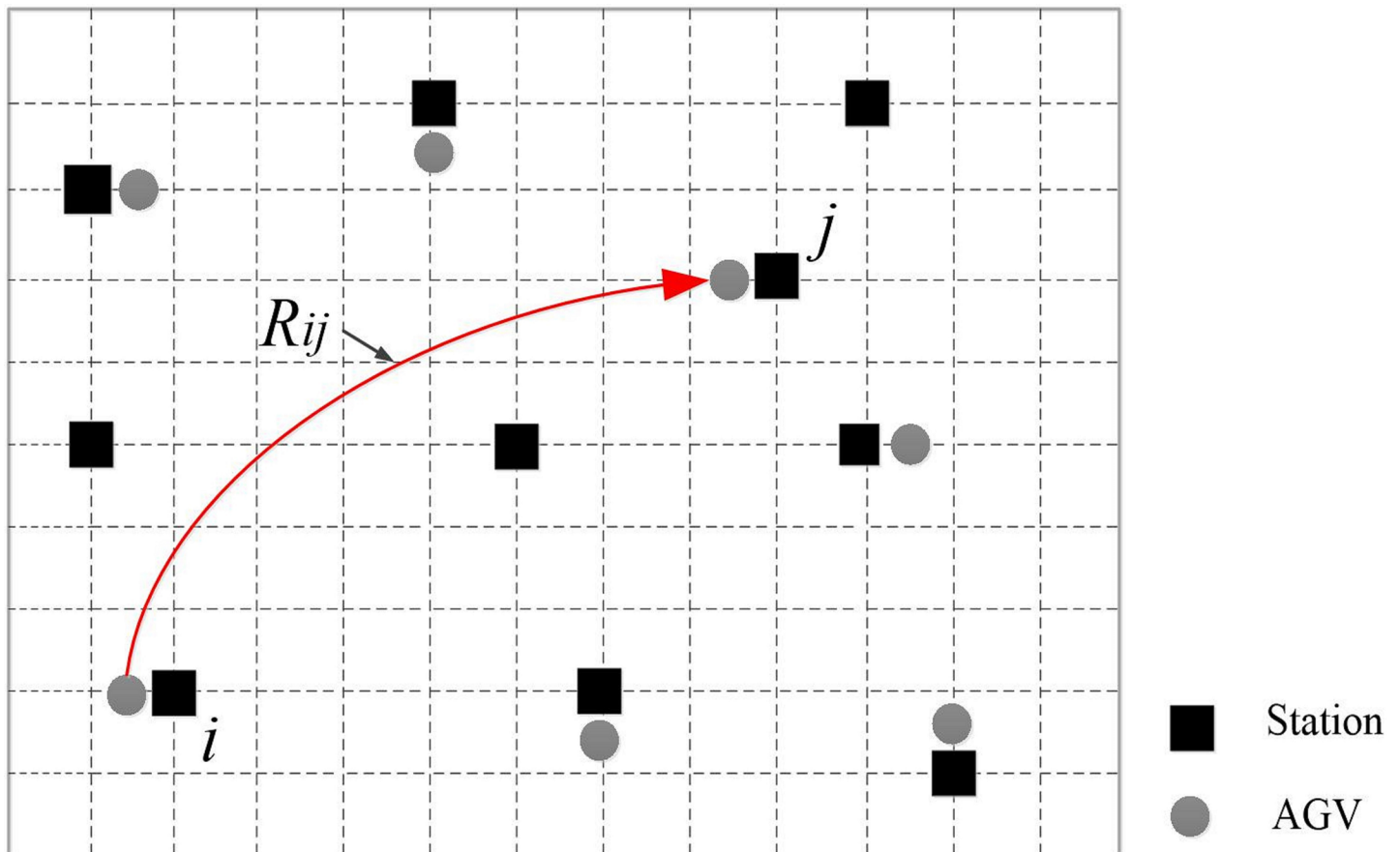
A jobshop manufacturing system with multiple AGVs performs material delivering. There are  $M$  AGVs traversing through  $N$  workstations ( $N > M$ ). For the workstation distribution, the following assumptions are considered for describing the details.

- $M$  AGVs traverse through  $N$  workstations ( $N > M$ ).
- Only one AGV passes through each workstation (except the starting point).
- Each AGV starts from the same starting point (workstation) and comes back to the starting point.
- Each AGV travels one route separately with the predefined path and the fixed speed.
- Two constraints are exerted:

The total path distance of all AGVs should be minimized;

Each single AGV path distance should be minimized.

The schematic diagram is shown in Fig 1.



**Fig 1. The work diagram of the AGV system.**

<https://doi.org/10.1371/journal.pone.0181747.g001>

### Mathematical formulation

Referring to Fig 1, we formulate the proposed problem, mathematically. The indexes, parameters and variables are introduced below.

1. Indexes
  - $i, j$ —Indexes for two workstations on the ends of an arc,  $i = 0, 1, 2 \dots, N, j = 0, 1, 2 \dots, N$ .
  - $k$ —Index for AGVs numbers,  $k = 1, 2 \dots, M$ .
  - $l$ —Index for the workstations requires AGVs to delivery,  $l = 1, 2 \dots, L_k, 0 < L_k < N$ .

2. Parameters
  - $R_{ij}$ —Arc between two workstations  $i$  and  $j$ .
  - $C_{ij}$ —Path through the corresponding arc segment  $R_{ij}$ .

3. Variables
  - For  $i = 0, 1, \dots, N, j = 0, 1, \dots, N, k = 1, \dots, M$ , define

$$X_{ijk} = \begin{cases} 1 & \text{the } k\text{th AGV passes the arc } R_{ij} \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

$$Y_{ki} = \begin{cases} 1 & \text{the } k\text{th AGV goes to the } i\text{th station} \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

$$z_k = \sum_{i=0}^{L_k} \sum_{j=0}^{L_k} C_{ij} X_{ijk}. \tag{3}$$

4. Objective function

$$J = \min \left( \sum_{k=1}^M z_k \right) \cap \min z_k \ (k = 1, 2, \dots, k), \tag{4}$$

5. Requirements

$$\sum_{k=1}^M Y_{ki} = \begin{cases} M, & i = 0 \\ 1, & i = 1, \dots, L_k, \end{cases} \tag{5}$$

$$\sum_{i=0}^{L_k} X_{ijk} = Y_{kj}, \ j = 0, \dots, L_k, \ k = 1, \dots, M, \tag{6}$$

$$\sum_{j=0}^{L_k} X_{ijk} = Y_{ki}, \ i = 0, \dots, L_k, \ k = 1, \dots, M, \tag{7}$$

where

Requirement (5) specifies that each AGV starts from the starting workstation 0, all workstations can only be accessed once by an AGV;

Requirement (6) shows that any arc starts from the starting workstation;  
Requirement (7) implies that any arc ends with the starting workstation.

### Algorithm design

The key problem of applying the genetic algorithm to the multi-AGV path planning is to adopt the effective coding and decoding methods. Genetic algorithms repeatedly select, crossover, and mutate the population to produce a new generation population that is more adaptable to the environment than its parents, until satisfying the desired requirements.

The step of the genetic algorithm includes: genetic coding, population selection, fitness function, selection action, crossover operation, and matrix decoding. The proposed new genetic algorithm minimizes the total path distance of all AGVs by selecting individuals with big fitness values, and minimizes each AGV path distances by the three-exchange heuristic crossover operator method.

### Genetic coding

Symbol 0 indicates the starting workstation (point); symbols 1, 2, . . . ,  $N$  mean the  $N$  workstations that need AGVs delivery. We add  $M - 1$  dummy symbols, denoting  $M - 1$  virtual sites, labeled  $N + 1, \dots, N + M - 1$ . They have the same coordinates as the starting site, meaning that every time a dummy symbol appears, the corresponding AGV returns to the starting point. Assume that a gene represents a path that an AGV travels; one chromosome contains all genes, i.e., all paths that all AGVs travel. To avoid frequent sub-paths, we assume that the path distance from the starting point 0 to the starting point 0 is infinite.

For example: there are 10 workstations, the code is 0-9, 5 AGVs to complete the task, a random chromosome coding is shown in Fig 2.

The paths of the five AGVs are as follows:

0 - 6 - -2 - -0  
 0 - 7 - -0  
 0 - 1 - -8 - -0  
 0 - 3 - -4 - -0  
 0 - 5 - -9 - -0

In the iterative process, there may be two kinds of dead solutions as the following Figs 3 and 4.

1. The virtual symbols are on one end of a chromosome

The paths of the five AGVs are as follows:

0 - 0 - -0  
 0 - 2 - -6 - -7 - -0  
 0 - 1 - -8 - -0  
 0 - 3 - -4 - -0  
 0 - 5 - -9 - -0

The 0 - 0 - -0 path means the distance is infinite, and can not meet the distance minimizing constraint, so this chromosome will be eliminated.

2. The virtual symbol appears continuously in a chromosome

The paths of the five AGVs are as follows:

0 - 6 - -2 - -0  
 0 - 7 - -8 - -1 - -0

0 - -0 - -0  
 0 - -3 - -4 - -0  
 0 - -5 - -9 - -0

Obviously, the 0 - -0 - -0 path is present on this chromosome, and can not meet the distance minimizing constraint, so this chromosome also will be eliminated.

### Population selection

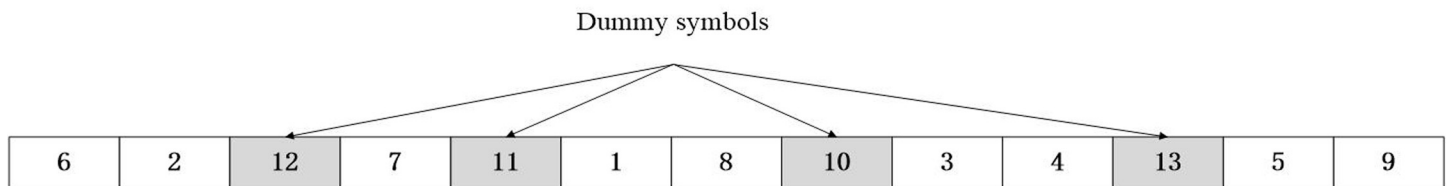
The appropriate population size is important for the convergence of the genetic algorithm. If the population size is too small, the genetic algorithm is easy to converge to the local optimal solution; on the contrary, if the population scale is too large, the computing speed of the genetic algorithm will be reduced. The size of the population is related to the variable  $N$ , and the appropriate population size should be controlled between  $4N$  and  $6N$  [49].

### Fitness function

In this paper, we use the exponential fitness functions according to [50]. The idea of this transformation method comes from the SA (simulated annealing) process [50]. Due to the advantages of exponentiation scale transformation, referring to [51], we choose a fitness function with exponentially transformation as:

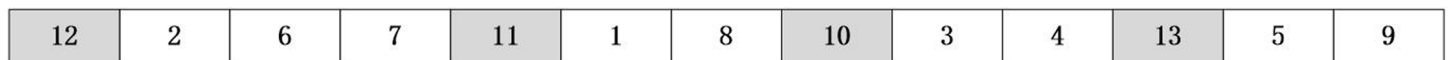
$$f = \alpha * \exp(\beta * Z), \tag{8}$$

where  $Z(= Z_1+Z_2+\dots+Z_k)$  is one of the parent individuals;  $\alpha$  and  $\beta$  are arithmetic constants;  $\alpha$  determines the coercion of replication, the smaller the value, the greater the replication intensity of the individual with the greatest fitness.



**Fig 2. A random chromosome.**

<https://doi.org/10.1371/journal.pone.0181747.g002>



**Fig 3. Situation 1 of dead solutions.**

<https://doi.org/10.1371/journal.pone.0181747.g003>



**Fig 4. Situation 2 of dead solutions.**

<https://doi.org/10.1371/journal.pone.0181747.g004>

### Selection operation

The selection operation is used to determine the recombination or crossover parent individuals and the number of offspring individuals generating by the candidate population. How to select an operator will directly affect the results of the genetic algorithm. An inappropriate operator will cause the evolution to stop or make the algorithm lose diversity, and produce premature problems [52].

In this paper, we use the Roulette Wheel Selection [53] to select the parent individuals, the probability of individual  $i$  is equal to the proportion of its fitness value and the sum of the individual population fitness [54], as the following,

$$P_i = f_i / \sum_{k=1}^Q f_k, \tag{9}$$

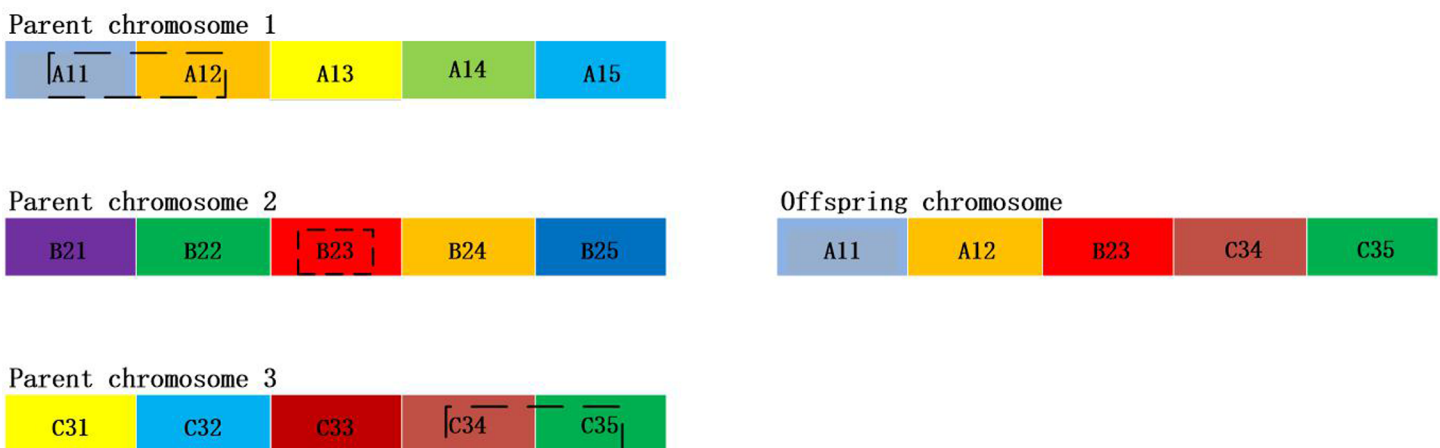
where  $f_i$  is the fitness value of the individual  $i$ .  $Q$  is the numbers of selected chromosomes or population size.

### Crossover operation

Traditionally, crossover refers to the process in which two chromosomes exchange some genes with each other in a certain way to form one new individuals. After crossover operation, a new generation is produced, and it inherits the father’s basic characteristics.

The idea of the three-exchange heuristic crossover operator method is to produce a progeny with three parent individuals. The proposed method increases the information of producing the progeny chromosomes, comparing with the traditional method with two parent individuals. The increased parent chromosomes improve the possibility of inheriting the excellent characteristics of the parents, and accelerate the searching speed of the algorithm. The explanation of the three-exchange heuristic crossover operator method is shown in Fig 5.

Taking a task including 10 workstations and 5 AGVs as an example, the process of three-crossover heuristic crossover operator method is described in detail as follows. The distance between the ten workstations is shown in Table 1.



**Fig 5. Three-exchange heuristic crossover operator method with 5 genes.**

<https://doi.org/10.1371/journal.pone.0181747.g005>

**Table 1. The distance between the ten workstations.**

| start | 0        | 1  | 2 | 3  | 4 | 5  | 6 | 7 | 8  | 9  |
|-------|----------|----|---|----|---|----|---|---|----|----|
| 0     | $\infty$ | 4  | 1 | 12 | 7 | 5  | 6 | 3 | 5  | 5  |
| 1     | 5        | 0  | 7 | 4  | 1 | 8  | 1 | 2 | 4  | 7  |
| 2     | 3        | 5  | 0 | 3  | 1 | 6  | 4 | 9 | 1  | 3  |
| 3     | 7        | 1  | 9 | 0  | 7 | 8  | 9 | 5 | 9  | 9  |
| 4     | 8        | 6  | 6 | 1  | 0 | 13 | 5 | 1 | 3  | 12 |
| 5     | 1        | 4  | 7 | 3  | 2 | 0  | 5 | 2 | 5  | 2  |
| 6     | 9        | 7  | 8 | 8  | 7 | 1  | 0 | 1 | 11 | 4  |
| 7     | 12       | 11 | 3 | 2  | 7 | 1  | 6 | 0 | 3  | 8  |
| 8     | 5        | 4  | 2 | 4  | 4 | 5  | 3 | 5 | 0  | 7  |
| 9     | 2        | 7  | 5 | 11 | 8 | 7  | 9 | 8 | 4  | 0  |

Note: 0 is the starting point.

<https://doi.org/10.1371/journal.pone.0181747.t001>

Three individuals were randomly selected as three-crossover heuristic crossover operators:

$A = 6\ 2\ 12\ 7\ 11\ 1\ 8\ 10\ 3\ 4\ 13\ 5\ 9$

$B = 3\ 2\ 12\ 7\ 9\ 11\ 1\ 4\ 13\ 5\ 10\ 6\ 8$

$C = 5\ 3\ 10\ 8\ 7\ 11\ 2\ 6\ 12\ 1\ 4\ 13\ 9$

The total distance of route  $A$  is 81, the largest distance of the single AGV path is 27;

The total distance of route  $B$  is 78, the largest distance of the single AGV path is 24;

The total distance of route  $C$  is 71, the largest distance of the single AGV path is 22.

Taking chromosome  $A$  as a reference, point 6 is the first position of chromosome  $A$ . From right to left, cyclically move genes in the chromosomes  $B$  and  $C$ , and stop when point 6 being the first position, then choose 6 as the first point of progeny  $S$ , the results are

$A = 6\ 2\ 12\ 7\ 11\ 1\ 8\ 10\ 3\ 4\ 13\ 5\ 9$

$B = 6\ 8\ 3\ 2\ 12\ 7\ 9\ 11\ 1\ 4\ 13\ 5\ 10$

$C = 6\ 12\ 1\ 4\ 13\ 9\ 5\ 3\ 10\ 8\ 7\ 11\ 2$

$S = 6\ *\ *\ *\ *\ *\ *\ *\ *\ *\ *\ *\ *$

From the [Table 1](#), we can get the distance around point 6 as follows,

$$d(6, 2) = 8,$$

$$d(6, 8) = 11,$$

$$d(6, 12) = d(6, 0) = 9.$$

Thus we get

$$d(6, 8) > d(6, 12) > d(6, 2).$$

To meet the constraint of minimizing single AGV path distance, we choose point 2 as the second point, the results are

$A = 6\ 2\ 12\ 7\ 11\ 1\ 8\ 10\ 3\ 4\ 13\ 5\ 9$

$B = 6\ 2\ 12\ 7\ 9\ 11\ 1\ 4\ 13\ 5\ 10\ 8\ 3$

$C = 6\ 2\ 12\ 1\ 4\ 13\ 9\ 5\ 3\ 10\ 8\ 7\ 11$

Similarly, we can determine the other genes of crossover progeny  $S$  in turn. Exerting with the single AGV path distance minimizing constraint, the crossover progeny  $S$  of the first crossover step is



|   |   |    |   |    |   |   |    |   |   |    |   |   |
|---|---|----|---|----|---|---|----|---|---|----|---|---|
| 6 | 2 | 12 | 7 | 11 | 1 | 8 | 10 | 3 | 4 | 13 | 5 | 9 |
|---|---|----|---|----|---|---|----|---|---|----|---|---|

Fig 6. A random chromosome.

<https://doi.org/10.1371/journal.pone.0181747.g006>

$$S = 6\ 2\ 12\ 7\ 9\ 11\ 1\ 4\ 13\ 5\ 10\ 8\ 3$$

The above obtained progeny chromosome indicates that the total path distance of all AGVs is 65 and the maximum path distance of a single AGV is 17. Obviously, the total path distance of all AGVs and the maximum single AGV path distance of the obtained *S* are less than those of original *A*, *B*, and *C*.

### Mutation operation

Mutation is to exchange genes within the same chromosome, resulting in a new individual. Mutation can determine the local search ability of the genetic algorithm, maintain the diversity of the group, prevent premature convergence of the genetic algorithm [55].

This paper adopts the exchanging mutation method [56]. The idea is to choose the ordinal numbers *a*, *b*, *c* ( $a < b < c$ ), and then insert the intermediate paths *a* and *b* (including *a b*) after *c* (*c* refers to the paths associated with it). For example, if the sequence numbers  $a = 2$ ,  $b = 5$  and  $c = 10$  are randomly generated, the corresponding individual transformations are as follows:

$$A = 6\ 2\ 12\ 7\ 11\ 1\ 8\ 10\ 3\ 4\ 13\ 5\ 9$$

After the mutation:

$$A = 6\ 1\ 8\ 10\ 3\ 4\ 2\ 12\ 7\ 11\ 13\ 5\ 9$$

### Matrix decoding

Set up a production scene with 10 workstations and 5 AGVs. A random chromosome is shown in Fig 6.

The paths of the five AGVs are:

$$0 - -6 - -2 - -0$$

$$0 - -7 - -0$$

$$0 - -1 - -8 - -0$$

$$0 - -3 - -4 - -0$$

$$0 - -5 - -9 - -0$$

According to Table 1, establish a 10 \* 10 workstation matrix *D*:

$$D = \begin{pmatrix} \infty & 4 & 1 & 12 & 7 & 5 & 6 & 3 & 5 & 5 \\ 5 & 0 & 7 & 4 & 1 & 8 & 1 & 2 & 4 & 7 \\ 3 & 5 & 0 & 3 & 1 & 6 & 4 & 9 & 1 & 3 \\ 7 & 1 & 9 & 0 & 7 & 8 & 9 & 5 & 9 & 9 \\ 8 & 6 & 6 & 1 & 0 & 13 & 5 & 1 & 3 & 12 \\ 1 & 4 & 7 & 3 & 2 & 0 & 5 & 2 & 5 & 2 \\ 9 & 7 & 8 & 8 & 7 & 1 & 0 & 1 & 11 & 4 \\ 12 & 11 & 3 & 3 & 7 & 1 & 6 & 0 & 3 & 8 \\ 5 & 4 & 2 & 4 & 4 & 5 & 3 & 5 & 0 & 7 \\ 2 & 7 & 5 & 11 & 8 & 7 & 9 & 8 & 4 & 0 \end{pmatrix}$$

The specific decoding steps are described as follows:

(1) Get the reachable matrices of AGVs

The travel path of AGV1 is 0 - -6 - -2 - -0, comparing the path with matrix D, value 1 represents the AGV1 through the station, otherwise represented with value 0. So the reachable matrix  $X_1$  of AGV1 can be obtained:

$$X_1 = \begin{pmatrix} \infty & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Similarly, we can get other AGVs reachable Matrices  $X_2, X_3, X_4, X_5$ .

(2) Get the path distance matrices of AGVs

Multiplying the matrix  $X_k$  ( $k = 1, 2, \dots, 5$ ) with  $D$  obtains the path distance matrix  $D_k$  ( $k = 1, 2, \dots, 5$ ) of each AGV.

For example, the path distance matrix for AGV1 is

$$D_1 = \begin{pmatrix} \infty & 0 & 0 & 0 & 0 & 0 & 6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

(3) Compute the path distance of AGVs

The path distance of AGV1 is

$$Z_1 = 6 + 3 + 8 = 17,$$

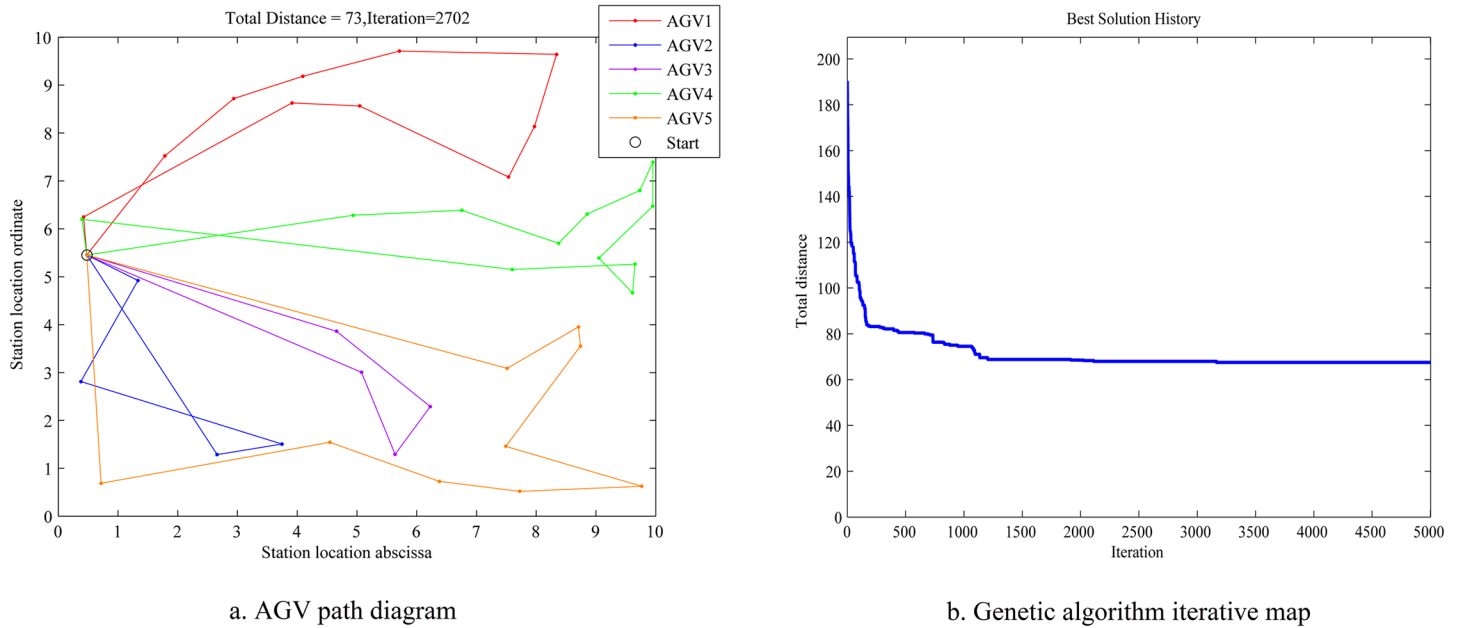
similarly,

$$Z_2 = 3 + 12 = 15,$$

$$Z_3 = 4 + 4 + 5 = 13,$$

$$Z_4 = 12 + 7 + 8 = 27,$$

$$Z_5 = 5 + 5 + 2 = 12,$$



**Fig 7. AGV diagram and map of the new genetic algorithm.**

<https://doi.org/10.1371/journal.pone.0181747.g007>

and the total path distance of all AGVs is

$$Z = Z_1 + Z_2 + Z_3 + Z_4 + Z_5 = 84.$$

### Simulation

In simulation, we set up the production scene with 5 AGVs and 50 workstations, and meet the requirements listed in the Section 2 and the double-path constraints. Set the population size is 200, applying the proposed new genetic algorithm to perform path optimization.

The simulation results for two genetic algorithms are drawn in Figs 7–10.

Figs 7 and 8 show that, at 3000 iterations, the total path distance is 72 for the new genetic algorithm, and is 86 for the traditional genetic algorithm.

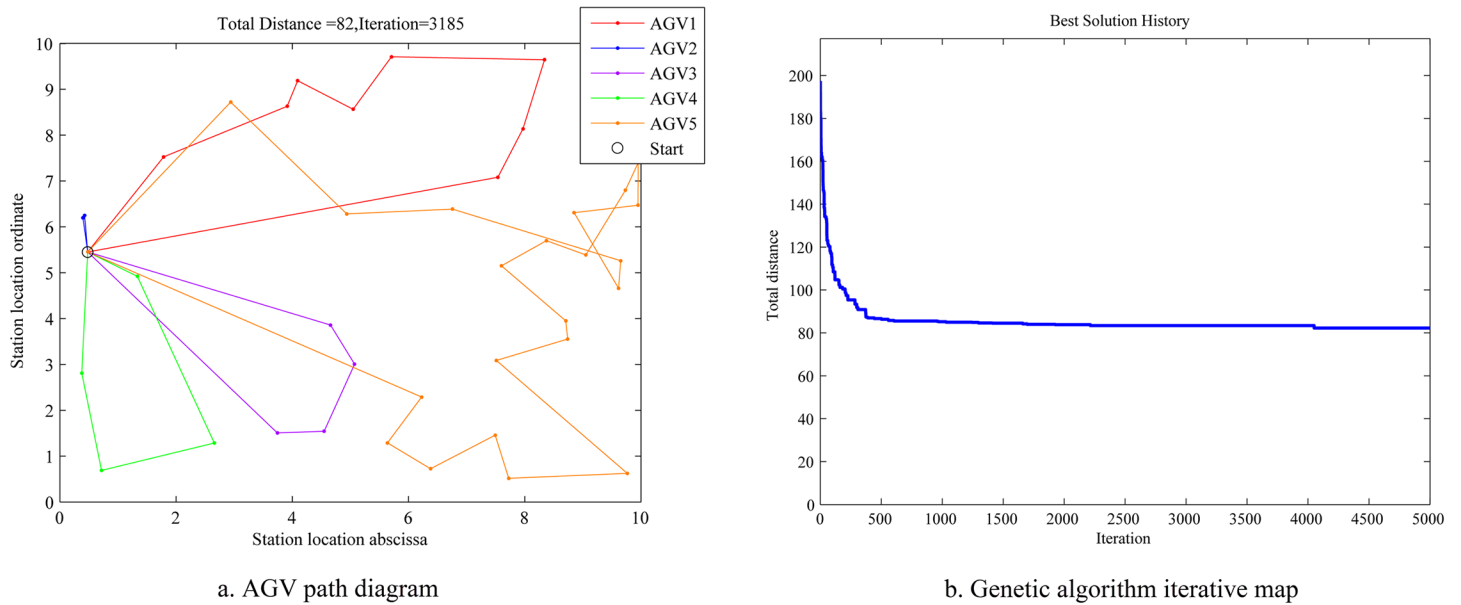
Fig 9 shows that, at 60 iterations, the maximum distance of single AGV is 34 for the traditional genetic algorithm, and is 32 for the new genetic algorithm.

Fig 10 shows the distance comparison between the two algorithms using bar graphs.

From the simulation, we can get:

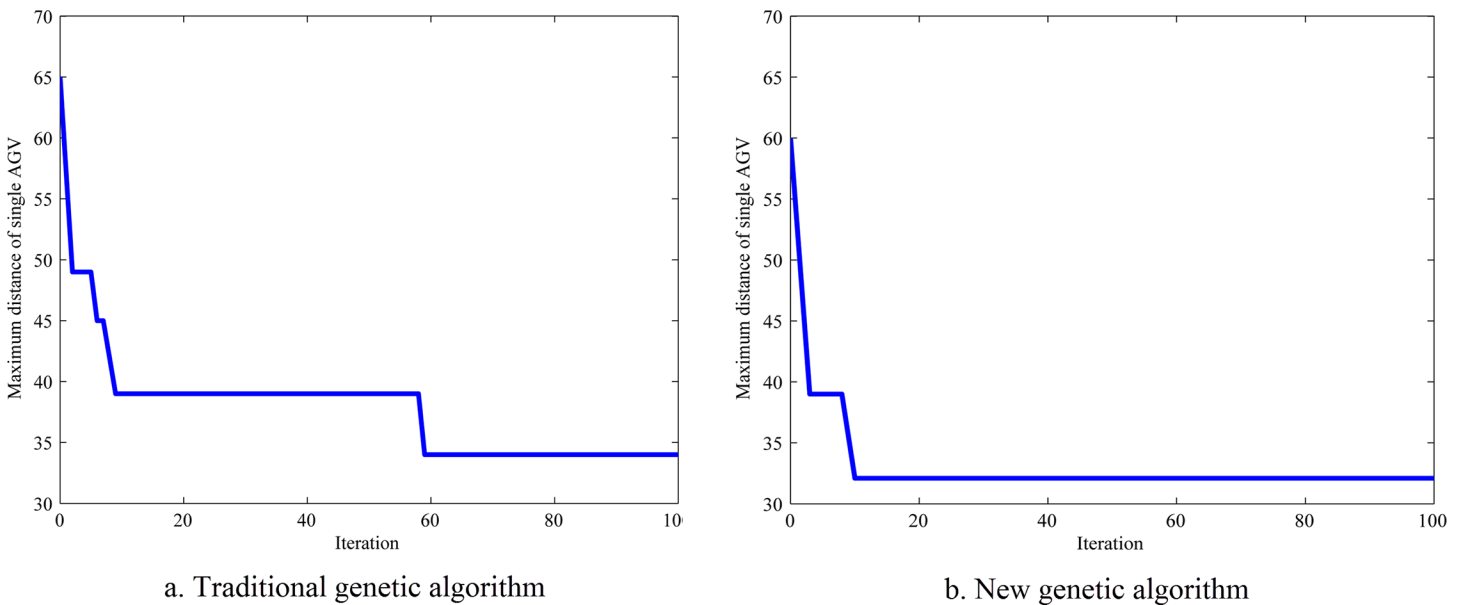
(1) The improved genetic algorithm have the shorter total path distance than that of the traditional genetic algorithm.

(2) The convergence speed of the improved genetic algorithm is faster than that of the traditional genetic algorithm.



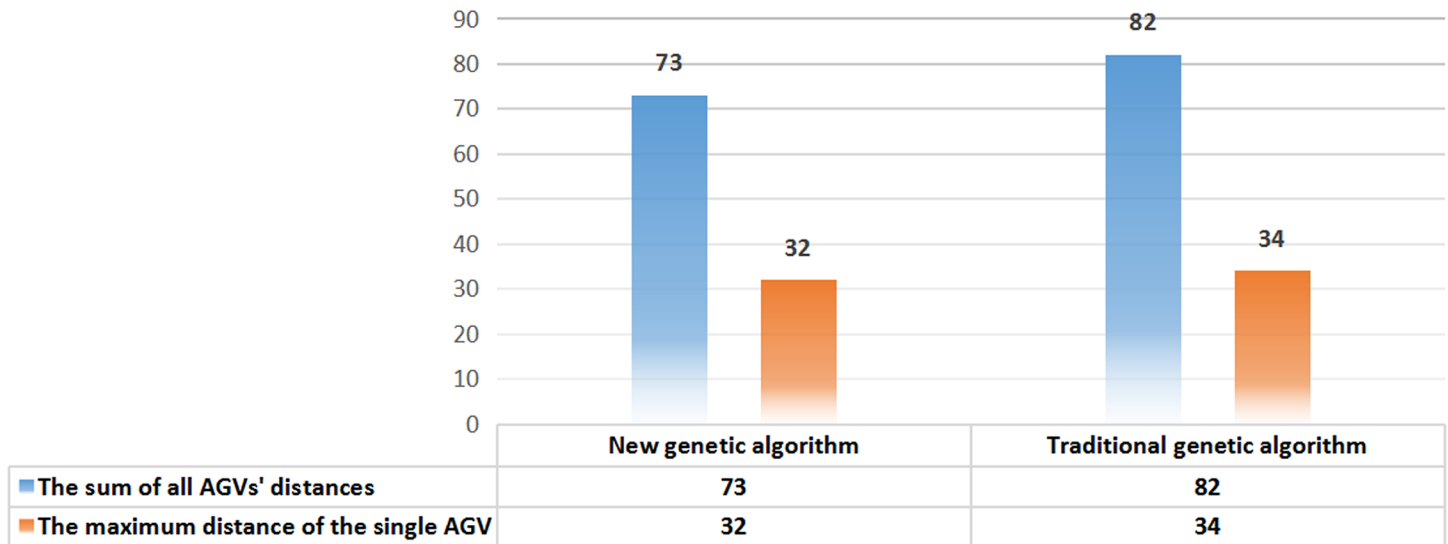
**Fig 8. AGV diagram and map of the traditional genetic algorithm.**

<https://doi.org/10.1371/journal.pone.0181747.g008>



**Fig 9. Maximum distance of single AGV.**

<https://doi.org/10.1371/journal.pone.0181747.g009>



**Fig 10. Distance comparison between two algorithms.**

<https://doi.org/10.1371/journal.pone.0181747.g010>

## Conclusions

We recast the multi-AGV path planning problem into the framework of an genetic algorithm to investigate the improved genetic algorithm on multi-AGV path optimization. In the improved genetic algorithm, by using three-exchange crossover heuristic operators with more information than that of the traditional two-exchange crossover heuristic operators, we get more optimal offsprings. By exerting double-path constraints of both minimizing the total path distance of all AGVs and minimizing each AGV path distance, we gain an optimal shortest total path distance in AGV delivery task. The simulation results show that all AGV path distance and the longest single AGV path distance are shortened by using the improved genetic algorithm.

## Supporting information

**S1 File. Program of AGV diagram and map.**

(ZIP)

**S2 File. Program of maximum distance of single AGV.**

(ZIP)

## Acknowledgments

This research was supported by the National Natural Science Foundation of China (No. 61573205) and the Shandong Provincial Natural Science Foundation of China (No. ZR2015FM017)

## Author Contributions

**Conceptualization:** Zengliang Han, Dongqing Wang.

**Data curation:** Zengliang Han, Zhiyong Zhao.

**Writing – original draft:** Zengliang Han, Dongqing Wang.

**Writing – review & editing:** Dongqing Wang, Feng Liu.

## References

1. Hu XS, Jiang JC, Egardt B, Cao DP. Advanced power-source integration in hybrid electric vehicles: multicriteria optimization approach. *IEEE Transactions on Industrial Electronics*. 2015; 62(12), 7847–7858. <https://doi.org/10.1109/TIE.2015.2463770>
2. Hu XS, Moura Scott J, Nilolce M, Egardt B, Cao DP. Integrated optimization of battery sizing, charging, and power management in plug-in hybrid electric vehicles. *IEEE Transactions on Control Systems Technology*. 2012; 24(3), 1036–1043. <https://doi.org/10.1109/TCST.2015.2476799>
3. Hu XS, Zou Y, Yang YL. Greener plug-in hybrid electric vehicles incorporating renewable energy and rapid system optimization. *Energy*. 2016; 111(20): 971–980. <https://doi.org/10.1016/j.energy.2016.06.037>
4. Hu XS, Clara MM, Yang YL. Charging, power management, and battery degradation mitigation in plug-in hybrid electric vehicles: a unified cost-optimal approach. *Mechanical Systems and Signal Processing*. 2017; 87(17), 4–16. <https://doi.org/10.1016/j.ymssp.2016.03.004>
5. Zhang Z, Zhao DB, Gao JW, Wang DQ, Dai YJ. FMRQ—a multiagent reinforcement learning algorithm for fully cooperative tasks. *IEEE Transactions on Cybernetics*. 2017; 47(6): 1367–1379. <https://doi.org/10.1109/TCYB.2016.2544866> PMID: 27101627
6. Anvari B, Angeloudis P, Ochieng WY. A multi-objective GA-based optimisation for holistic manufacturing, transportation and assembly of precast construction. *Automation in Construction*. 2016; 71(2): 226–241. <https://doi.org/10.1016/j.autcon.2016.08.007>
7. Hadas Y, Nahum OE. Urban bus network of priority lanes: A combined multi-objective, multi-criteria and group decision-making approach. *Transport Policy*. 2016; 52: 186–196. <https://doi.org/10.1016/j.tranpol.2016.08.006>
8. Sabar NR, Kieu LM, Chung E, Tsubota T, de Almeida PEM. A memetic algorithm for real world multi-intersection traffic signal optimisation problems. *Engineering Applications of Artificial Intelligence*. 2017; 63: 45–53. <https://doi.org/10.1016/j.engappai.2017.04.021>
9. Xu G, Liang X, Yao S, Chen D, Li Z. Multi-objective aerodynamic optimization of the streamlined shape of high-speed trains based on the Kriging model. *PLoS One*. 2017; 12(1): e0170803. <https://doi.org/10.1371/journal.pone.0170803> PMID: 28129365
10. Chodrow PS, al-Awwad Z, Jiang S, González MC. Demand and congestion in multiplex transportation networks. *PLoS One*. 2016; 11(9): e0161738. <https://doi.org/10.1371/journal.pone.0161738> PMID: 27657738
11. Ramezani M, Bashiri M, Tavakkoli-Moghaddam R. A new multi-objective stochastic model for a forward/reverse logistic network design with responsiveness and quality level. *Applied Mathematical Modelling*. 2013; 37(1-2): 328–344. <https://doi.org/10.1016/j.apm.2012.02.032>
12. Wang F, Lai XF, Shi N. A multi-objective optimization for green supply chain network design. *Decision Support Systems*. 2011; 51(2): 262–269. <https://doi.org/10.1016/j.dss.2010.11.020>
13. Liu Q, Zhang CY, Zhu K, Rao YQ. Novel multi-objective resource allocation and activity scheduling for fourth party logistics. *Computers & Operations Research*. 2014; 44: 42–51. <https://doi.org/10.1016/j.cor.2013.10.010>
14. Said H, El-Rayes K. Automated multi-objective construction logistics optimization system. *Automation in Construction*. 2014; 43: 110–122. <https://doi.org/10.1016/j.autcon.2014.03.017>
15. Qu S, Ji Y. The worst-case weighted multi-objective game with an application to supply chain competitions. *PLoS One*. 2016; 11(1): e0147341. <https://doi.org/10.1371/journal.pone.0147341> PMID: 26820512
16. Falsafi H, Zakariazadeh A, Jadid S. The role of demand response in single and multi-objective wind-thermal generation scheduling: a stochastic programming. *Energy*. 2014; 64: 853–867. <https://doi.org/10.1016/j.energy.2013.10.034>

17. Geng Z, Cui Y, Xia L, Zhu Q, Gu X. Compromising adjustment solution of primary reaction coefficients in ethylene cracking furnace modeling. *Chemical Engineering Science*. 2012; 80: 16–29. <https://doi.org/10.1016/j.ces.2012.05.039>
18. Ojalehto V, Miettinen K, Laukkanen T. Implementation aspects of interactive multiobjective optimization for modeling environments: the case of GAMS-NIMBUS. *Computational Optimization and Applications*. 2014; 58(3): 757–779. <https://doi.org/10.1007/s10589-014-9639-y>
19. Lu L, Cook ACM, Robinson TJ. A case study to demonstrate Pareto frontiers for selecting a best response surface design with simultaneously optimizing multiple criteria. *Applied Stochastic Models in Business & Industry* 2012; 3(28):85–96.
20. Qin AK, Huang VL, Suganthan PN. Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Transactions on Evolutionary Computation*. 2009; 13(2): 398–417. <https://doi.org/10.1109/TEVC.2008.927706>
21. Neri F, Carlos C. Memetic algorithms and memetic computing optimization: a literature review. *Swarm and Evolutionary Computation*. 2012; 2: 1–14. <https://doi.org/10.1016/j.swevo.2011.11.003>
22. Zhang L, Wang ZP, Hu XS, Dorrell David G. Experimental impedance investigation of an ultracapacitor at different conditions for electric vehicle applications. *Journal of Power Sources*. 2015; 287(15): 129–138. <https://doi.org/10.1016/j.jpowsour.2015.04.043>
23. Zhang L, Hu XS, Wang ZP, Sun FC, Dorrell David G. Fractional-order modeling and State-of-Charge estimation for ultracapacitors. *Journal of Power Sources*. 2016; 314(16): 28–34. <https://doi.org/10.1016/j.jpowsour.2016.01.066>
24. Zhang L, Dorrell David G. Genetic algorithm based optimal component sizing for an electric vehicle. The 39th Annual Conference of the IEEE Industrial Electronics Society (IECON 2013), Vienna, Austria, 2013.
25. Vis IFA, De Koster R, Roodbergen KJ, Peeters LWP. Determination of the number of automated guided vehicles required at a semi-automated container terminal. *Journal of the Operational Research Society*. 2001; 52(4): 409–417. <https://doi.org/10.1057/palgrave.jors.2601094>
26. Koo P, Jang J, Suh J. Estimation of part waiting time and fleet sizing in AGV systems. *International Journal of Flexible Manufacturing Systems*. 2005; 16(3): 211–228. <https://doi.org/10.1007/s10696-005-1008-9>
27. Vivaldini K, Rocha LF, Martarelli NJ, Becker M, Moreira AP. Integrated tasks assignment and routing for the estimation of the optimal number of AGVS. *The International Journal of Advanced Manufacturing Technology*. 2016; 82(1): 719–736. <https://doi.org/10.1007/s00170-015-7343-4>
28. Martínez-Barberá H, Herrero-Pérez D. Autonomous navigation of an automated guided vehicle in industrial environments. *Robotics and Computer-Integrated Manufacturing*. 2010; 26(4): 296–311. <https://doi.org/10.1016/j.rcim.2009.10.003>
29. Jaiganesh V, Kumar JD, Girijadevi J. Automated guided vehicle with robotic logistics system. *Procedia Engineering*. 2014; 97: 2011–2021. <https://doi.org/10.1016/j.proeng.2014.12.444>
30. Li B, Liu H, Xiao D, Yu GZ, Zhang YM. Centralized and optimal motion planning for large-scale AGV systems: A generic approach. *Advances in Engineering Software*. 2017; 106: 33–46. <https://doi.org/10.1016/j.advengsoft.2017.01.002>
31. Xin JB, Negenborn RR, Lodewijks G. Trajectory planning for AGVs in automated container terminals using avoidance constraints: a case study. *IFAC Proceedings*. 2014; 47(3): 9828–9833. <https://doi.org/10.3182/20140824-6-ZA-1003.01306>
32. Xidias EK, Azariadis PN. Mission design for a group of autonomous guided vehicles. *Robotics and Autonomous Systems*. 2011; 59(1): 34–43. <https://doi.org/10.1016/j.robot.2010.10.003>
33. Bocewicz G, Nielsen I, Banaszak Z. Automated guided vehicles fleet match-up scheduling with production flow constraints. *Engineering Applications of Artificial Intelligence*. 2014; 30: 49–62. <https://doi.org/10.1016/j.engappai.2014.02.003>
34. Ji M, Xia J. Analysis of vehicle requirements in a general automated guided vehicle system based transportation system. *Computers & Industrial Engineering*. 2010; 59(4): 544–551. <https://doi.org/10.1016/j.cie.2010.06.013>
35. Draganjac I, Miklič D, Kovačić Z, Vasiljević G, Bogdan S. Decentralized control of multi-AGV systems in autonomous warehousing applications. *IEEE Transactions on Automation Science and Engineering*. 2016; 13(4): 1433–1446. <https://doi.org/10.1109/TASE.2016.2603781>
36. Smolic-Rocak N, Bogdan S, Kovacic Z, Petrovic T. Time windows based dynamic routing in Multi-AGV systems. *IEEE Transactions on Automation Science and Engineering*. 2010; 7(1): 151–155. <https://doi.org/10.1109/TASE.2009.2016350>

37. Kok KY, Rajendran P. Differential-evolution control parameter optimization for unmanned aerial vehicle path planning. *PLoS One*. 2016; 11(3): e0150558. <https://doi.org/10.1371/journal.pone.0150558> PMID: 26943630
38. Wang JQ, Ersoy OK, He MY, Wang F. Multi-offspring genetic algorithm and its application to the traveling salesman problem. *Applied Soft Computing*. 2016; 43: 415–423. <https://doi.org/10.1016/j.asoc.2016.02.021>
39. Wang YZ, Chen Y, Lin Y. Memetic algorithm based on sequential variable neighborhood descent for the minmax multiple traveling salesman problem. *Computers & Industrial Engineering*. 2017; 106: 105–122. <https://doi.org/10.1016/j.cie.2016.12.017>
40. Jiang ZB, Yang Q. A discrete fruit fly optimization algorithm for the traveling salesman problem. *PLoS One*. 2016; 11(11): e0165804. <https://doi.org/10.1371/journal.pone.0165804> PMID: 27812175
41. Hsueh CF. A simulation study of a bi-directional load-exchangeable automated guided vehicle system. *Computers and Industrial Engineering*. 2010; 58(4): 594–601. <https://doi.org/10.1016/j.cie.2009.12.008>
42. Xin JB, Negenborn RR, odewijks G. Trajectory planning for AGVs in automated container terminals using avoidance constraints: a case study. *IFAC Proceedings Volumes*. 2014; 47(3): 9828–9833. <https://doi.org/10.3182/20140824-6-ZA-1003.01306>
43. Miyamoto T, Inoue K. Local and random searches for dispatch and conflict-free routing problem of capacitated AGV systems. *Computers & Industrial Engineering*. 2016; 91: 1–9. <https://doi.org/10.1016/j.cie.2015.10.017>
44. Alcaidea D, Chub C, Katsc V, Levnerd E, Sierksmae G. Cyclic multiple-robot scheduling with time-window constraints using a critical path approach. *European Journal of Operational Research*. 2007; 177(1): 147–162. <https://doi.org/10.1016/j.ejor.2005.11.019>
45. Bocewicz G, Banaszak Z, Nielsen I. Multimodal processes prototyping subject to fuzzy operation time constraints. *IFAC-Papers OnLine*. 2015; 48(3): 2103–2108. <https://doi.org/10.1016/j.ifacol.2015.06.399>
46. Contreras-Bolton C, Parada V. Automatic combination of operators in a genetic algorithm to solve the traveling salesman problem. *PLoS One*. 2015; 10(9): e0137724. <https://doi.org/10.1371/journal.pone.0137724> PMID: 26367182
47. Long Q, Wu CZ, Huang TW, Wang XY. A genetic algorithm for unconstrained multi-objective optimization. *Swarm and Evolutionary Computation*. 2015; 22: 1–14. <https://doi.org/10.1016/j.swevo.2015.01.002>
48. Prakash A, Chan FTS, Deshmukh SG. FMS scheduling with knowledge based genetic algorithm approach. *Expert Systems with Applications*. 2011; 38(4): 3161–3171. <https://doi.org/10.1016/j.eswa.2010.09.002>
49. Endo S, Konishi M, Yoshida M. Motion planning method for mobile robots in large scale transportation systems by genetic algorithms. *Trans Inst Syst Control Inform Eng*. 2000; 45(13): 115–23.
50. Ingber L. Very fast simulate annealing, *Math Comput Modeling*. 1989; 32(12): 967–973. [https://doi.org/10.1016/0895-7177\(89\)90202-1](https://doi.org/10.1016/0895-7177(89)90202-1)
51. Yang SQ, Yang JM, Sun C. Improved exponentiation scale transformation in application of genetic algorithm. *Computer Engineering and Applications*. 2014; 169(17): 8–11.
52. Egbelu PJ. A framework for the selection of idle vehicle home locations in an automated guided vehicle system, *International Journal of Production Research*. 2009; 44(3): 543–562.
53. Xia GM, Zeng JC. A stochastic particle swarm optimization algorithm based on the genetic algorithm of roulette wheel selection, *Computer Engineering and Science*. 2007; 29(6): 6–11.
54. Van den Bergh F, Engelbrecht A. Using Neighborhood with the Guaranteed Convergence PSO. *Proc of 2003 IEEE Swarm Intelligence Symp*. 2003; 62(16): 235–342.
55. Bai SF. Research on multiple AGV system scheduling based on mixed regional control model. *Nanjing University of Aeronautics and Astronautics*. 2012; 31(3): 10–22.
56. Jin F, Fan K, Wang JL. Research on AGVs scheduling based on queuing theory. *Chinese Journal of Scientific Instrument*. 2004; S(1): 844–846.