Measurement and control of system resilience recovery by path planning based on improved genetic algorithm

YuMei Wu\textsuperscript{1,2}, Zhen Li\textsuperscript{2,3}, Chenxu Sun\textsuperscript{2,3}, ZhaoBin Wang\textsuperscript{2,3}, DongSheng Wang\textsuperscript{2,4} and Zhengwei Yu\textsuperscript{1,2}

Abstract
Aiming at the problems of basic genetic algorithm in the field of path planning to system resilience recovery such as excessive randomness of initial population, slow convergence, low efficiency of evolution operator, and poor population diversity, this paper uses quotient model to measure resilience, uses overall task importance to measure system performance, and proposes an improved genetic algorithm on initial population and evolutionary operation. Improved genetic algorithm (IHGA) proposes a new greedy model that considers system node tasks importance, travel time, and maintenance time, which uses greedy ideas to generate partial high-quality initial population. And a new operator is also designed as intra-group head-to-head mutation operator (IHMO) to control the evolution to be more determinate and less ineffectively random. The simulation results in three cases show that the IHGA overcomes the defects and can better effectively recover system resilience with comparison to basic genetic algorithm (BGA) and multi-chromosome genetic algorithm (MCGA). Specially, it has obviously better optimal solution, convergence, and stability, especially in the harsh conditions as shorter repair time, more and unbalanced demands for spare parts, which shows the IHGA has great value to deal with measurement and control of system resilience recovery in practice.

Keywords
Measurement, resilience recovery, path planning, genetic algorithm

Date received: 11 March 2021; accepted: 15 April 2021

Introduction
The term of resilience is originated from early social ecology,\textsuperscript{1-3} which is defined as the ability of an entity to return normal state when being subjected to an event that has changed its state in general semantics.\textsuperscript{4} Similar to the concepts of reliability, survivability, and tolerance, resilience is described as a performance indicator of the ability of a system to respond to changes and reduce risks.

In recent years, with the increase of natural disasters and terrorist attacks, the concept of resilience has become an important term in many fields, such as infrastructure system,\textsuperscript{5-7} engineering field,\textsuperscript{4} and system safety.\textsuperscript{3,9} In engineering systems, resilience mainly involves three elements: the ability to predict disturbances or failures, the ability to maintain performance when disturbances or failures occur, and the ability to recover performance after disturbances or failures. Among them, the ability of disturbances or failures prediction and performance retention should be considered carefully in the design of resilient system, and the ability of system to recover from disturbances or failures is the very focus of disaster reconstruction.

System resilience is an extension of traditional system reliability and risk concepts, and has become one of the characteristics that require to be considered in the field of system evaluation. Improving system resilience could help reduce the risk of system damaged in an uncertain environment, making the system more

1School of Reliability and Systems Engineering, Beihang University, Beijing, China
2Reliability and Systems Engineering Open Group, Jiangsu University of Science and Technology, Zhenjiang, China
3School of Electronics and Information, Jiangsu University of Science and Technology, Zhenjiang, China
4School of Computer Science and Application, Jiangsu University of Science and Technology, Zhenjiang, China

Corresponding author:
Zhen Li, School of Electronics and Information, Jiangsu University of Science and Technology, No.666 Changhui Road, Dantu District, Zhenjiang City, Jiangsu Province, China. 212100.
Email: justlz@just.edu.cn

Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (https://creativecommons.org/licenses/by/4.0/) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (https://us.sagepub.com/en-us/nam/open-access-at-sage).
reliable, safe, and stable. Therefore, it is of great practical significance to study the resilience recovery of system under attack.

From the literature, we can see that there are three recovery mechanisms when system nodes are failed after being attacked: node backup strategy, node replacement strategy and node maintenance strategy. This paper makes research on the node maintenance strategy to recover system resilience. In general, not all failed nodes can be repaired at the same time, so the failed nodes should be restored under a certain maintenance sequence in a limited time and cost.

In this paper, the maintenance strategy to recover system resilience is abstracted into a path planning. According to the different characteristics of the algorithm, the path planning algorithm can be divided into two categories: precise algorithm and heuristic algorithm. The precise algorithms mainly include branch decomposition algorithm, LaGrange relaxation, branch and bound method. Heuristic algorithms are represented by simulated annealing, genetic algorithm, tabu search algorithm, ant colony algorithm, and particle swarm algorithm. In recent years, more and more scholars have chosen heuristic algorithm to solve the path planning problem of recovery. Jing et al. proposed an efficient and low-cost NUWS resilience recovery approach. That is, an improved genetic algorithm was proposed to determine and optimize the position of the relay weapon units. In which, the authors combined elite individual retention strategy and roulette selection strategy, and proposed adaptive crossover and mutation operators to prevent premature convergence. However, the individual convergence rate of this method in solving the multi-chromosome coding problem is slow, and the quality of the solution is not good enough. Li et al. proposed a two-layer programming model to solve the optimization problem of traffic network recovery strategy in ERP, considering the maximum of system resilience and the user path selection behavior after disaster. Also, the paper proposed a novel algorithm to optimize path planning by integrating Frank-Wolfe algorithm into genetic algorithm. However, the basic genetic algorithm (BGA) has shortcomings of redundant search space, low efficiency of evolution operator, and poor population diversity. The genetic algorithm proposed by Zhang et al. also has the same problems as BGA in the optimization of moving path of mobile water sink. Zhang and Wei designed a parallel coding method for bridge inspection and restoration after disaster, and developed a hybrid genetic algorithm that combines traditional genetic algorithms and specially designed heuristic algorithms, improving program’s computational efficiency. The paper provided a new idea for the priority decision-making issue of emergency inspection and recovery after disaster, but the hybrid genetic algorithm may not be able to obtain the global optimal solution. Wang et al. used MVODM strategy to generate the initial solution of genetic algorithm, and introduced two-way/three-crossing greedy operator to improve the quality of the solution. However, it has poor population diversity in the middle and later period. Ye et al. proposed a multi-chromosome genetic algorithm with complex mutation operator and multi-chromosome coding method (MCGA), which avoided search space redundancy and speed up the algorithm convergence, but maybe not able to get a better feasible solution.

Through the analysis above, this paper adopts genetic algorithm to study path planning of system resilience recovery. The path planning of resilience recovery is a discrete combinatorial optimization problem. The characteristics of the solution of this problem are just in line with the essence and characteristics of genetic algorithm designing. Another, genetic algorithm has strong versatility and good parallelism, so many literatures have used the algorithm to solve similar problems. In addition, some scholars also use particle swarm algorithm or wolf swarm algorithm in the solution of similar problem. As particle swarm algorithm cannot effectively solve the discrete and combinatorial optimization problems, and the wolf pack algorithm is too scattered with low globality, difficult to find the optimal solution. Therefore, it is not as good as the genetic algorithm in terms of the degree of incorporation of the algorithm principle.

Genetic algorithm is a random search algorithm that simulates natural selection theory and biological evolution mechanism. With good global search capabilities and parallelism, it is widely used in path planning, but it also has some shortcomings such as excessive randomness of the initial population, slow convergence, low quality of convergent individuals, low efficiency of genetic evolution operators, poor population diversity, and no control to improve the certainty of evolution direction. At the same time, many literatures look at all nodes as having the same importance which is not accord to the situation in practice. Actually we need to repair nodes are more important to recover system resilience in prior.

In order to deal with the problems above, this paper proposes an improved genetic algorithm based on improved initial population and intra-group head-to-head mutation operator (IHGA) to cope with the path planning of system to recovery resilience. This paper makes major innovations and contributions as follows: (1) A planning method is designed to search for optimal path that maximizes system resilience under time, spare parts constraint, and multi-personnel maintenance. (2) This paper proposes a method to improve the quality of initial population, that is, uses a new greedy model and random method together to generate initial population. In it, the greedy model that considers node task importance, travel time, and maintenance time to generate part of high-quality initial population. The mixed initial population including greedy part and random part improves the initial solution that will cause a better solution and faster convergence. (3) IHGA proposes intra-group head-to-head mutation operator (IHMO), which comprehensively considers
Recovery(t) is the system performance recovered at time t, and Loss(t_0) is the loss value of system performance.

**Optimization model**

In this section, resilience recovery model is established for three cases. Case 1 refers to the situation that requires to maximize the recovery of system resilience within a limited time after damaged. Case 2 is more severe than Case 1, it has more constraints, such as shorter repairing time and adding demands of different types of spare parts. In Case 3, it adds more and unbalanced demand for spare parts of each damaged node. As shown in formula 1 above, system resilience is related to system performance. This paper uses overall task importance of system as a measurement of system performance, the mathematical model of recovery path planning to system resilience is as follows.

**Case 1.** The mathematical model in Case 1 can be described as: Given that the system is damaged, there are N nodes damaged in need of repair are in uneven distribution, each node with different task importance. Set S maintenance personnel start from different node, and repair damaged nodes respectively in a limited time (t_l). Each personnel must recover at least x nodes and the nodes passed through are not repeated. The goal is to achieve the highest overall task importance of system for maintenance within a limited time, as shown in equations (2) to (4).

\[
\max f = \sum_{k=1}^{s} \sum_{i=m}^{r} z_{ik} \tag{2}
\]

\[
T_k = \sum_{i=m}^{r} h_{ik} \leq t_l, k = 1, ..., S \tag{3}
\]

\[
r - m \geq x \tag{4}
\]

Where \(z_i\) represents the task importance of node \(i\), \(k\) represents the index of the maintenance personnel, \(i\) represents the index of the node, \(m\) represents the start node of the maintenance personnel \(k\), and \(r\) represents the end node of the maintenance personnel \(k\). \(h_{ik}\) represents the time required for the \(k\)-th maintenance personnel to repair the \(i\)-th node. It can be seen from equation (2) that the solution to system resilience recovery is the maintenance path which makes \(f\) maximum.

**Case 2.** Case 2 simulates the system suffered attack, there are \(N\) damaged nodes for repair are in uneven distribution. It is known that the maintenance of each node equipment requires several \(A\), \(B\), and \(C\) spare parts. Let maintenance personnel \(S\) carry limited spare parts \(G1, G2, G3\) and start from different locations to repair within a limited time \(t_l\). In case 2, the demand of node spare parts is in the range of spare parts \(A\), \(B\), and \(C\) to \((1, 3)\).
Similarly, each repairer must recover at least \( x \) nodes, and the nodes passed during repair process cannot be repeated. Assume that task importance matrix of damaged nodes is \( Z = (z_{ij})_{m \times 1} \), where \( z_i \) represents task importance of node \( i \), the matrix of spare parts required for maintenance is \( C = (c_{ij})_{m \times 3} \), and the matrix of time required for maintenance is \( H = (h_{ij})_{m \times 1} \). The system task importance after maintenance is shown in equation (2) above. The mathematical model in Case 2 can be described as formula (5) to (9).

\[
\begin{align*}
T_k &= \sum_{i=1}^{r} h_{ik} \leq t_i, k = 1, \ldots, S \quad (5) \\
A &= \sum_{i=1}^{r} c_{ik} \leq G_1, k = 1, \ldots, S \quad (6) \\
B &= \sum_{i=1}^{r} c_{2k} \leq G_2, k = 1, \ldots, S \quad (7) \\
C &= \sum_{i=1}^{r} c_{3k} \leq G_3, k = 1, \ldots, S \quad (8) \\
r - m &\geq x \quad (9)
\end{align*}
\]

Case 3. From the relationship between the demand and supply of spare parts for node maintenance, it is significantly required to discuss the issue of system resilience recovery in the case of balanced and unbalanced demand of node spare parts. That is the difference in Case 2 and Case 3. In Case 3, the demand for node spare parts is more than other two cases and unbalanced. The demand for \( A \), \( B \), and \( C \) has increased to the demand for spare parts \( A \) and \( B \) are both \( (1, 5) \), and that for spare parts \( C \) is \( (3, 7) \). Case 3 simulates the system is more damaged and in an unbalanced need of more spare parts.

Introduction of MCGA and design of improved genetic algorithm

Introduction of MCGA

The difference between MCGA and BGA is reflected in the evolution operator. The algorithm proposes a complex mutation operator for the multi-chromosome encoding method, namely the complex mutation operator tree. The operator is generated by a combination of path mutation operator, span mutation operator and sliding mutation operator. Among them, the path mutation operator only performs mutation operations within a single chromosome, that is, exchanges any two gene fragments within any single chromosome. Spanning mutation operator is an operation of mutation between different chromosomes, that is, two gene sequences randomly selected in any two chromosomes are exchanged. The sliding mutation operator is a kind of adjacent mutation in which all chromosomes participate. Specifically, the last gene of each chromosome is moved to the first gene of the previous chromosome to form a sliding mutation operator. In algorithm principle, MCGA adds a variety of random mutations, but it does not control the direction of the mutation and consider the optimization for initial population. Therefore, this article makes further improvements to the genetic algorithm.

The design of IHGA

The recovery path planning is to solve a specific problem involving multiple maintenance personnel simultaneously repairing damaged nodes. In Case 1, we only take time constraint into account. While in Case 2 and Case 3, we should consider both time constraint and spare parts constraint with balanced and unbalanced demand for spare parts. In view of the shortcomings of BGA, such as slow convergence, premature and poor solution, this paper proposes IHGA to improve the path planning of system resilience recovery.

IHGA is shown in Figure 2. Firstly, this paper adopts coding method of multi-chromosomes integer coding. Individual composed of multi-chromosomes with different length represents a solution of path planning. Secondly, the initial population are generated by the combination of random method and greedy algorithm. The details of initial population generation through greedy algorithm are shown in Section 3.2. Thirdly, calculate fitness value of each individual according to fitness function, and record optimal fitness value and path solution. In the process of population iteration, continuously update historical optimal fitness value, which ensures that population always evolves in a better direction.

In order to increase population diversity and search space of solution, this paper adopts an elite selection strategy of paralleled multi-population. Elite individuals are selected from initial population for crossover and mutation. And this paper takes into account the characteristics of multi-chromosome encoding method and initial population generation, based on the ideal of greedy, and proposes intra-group head-to-head mutation operator–IHMO, which eliminates inferior individuals, speeds up algorithm convergence and improves the solution. Details of evolutionary operator are given in section 3.3. After evolution operators are executed, the offspring population is finally generated. Finally, when algorithm stopping criterion (the maximum number of iterations) is met, IHGA gives optimal fitness value and corresponding path solution.

Initial population

In this paper, the initial population is optimized by the idea of greedy combined with randomness. As the initial population size in IHGA is represented by \( M \), the first \( M/2 \) individuals are randomly generated, and the rest individuals are selectively generated by greedy
The algorithm flow chart is shown in Figure 3, and steps are as follows:

1. Among the $N$ damaged nodes, the first node of each maintenance team is randomly generated.
2. Set the following greedy degree model:

$$F(X, Y) = \frac{\text{degree}(X)}{d(X, Y)/v + h(X)} * A$$  \hspace{1cm} (10)$$

Where $F(X, Y)$ represents the greedy degree of node $X$ to node $Y$, $\text{degree}(X)$ represents the task importance of node $X$, $d(X, Y)$ represents the distance between node $X$ and node $Y$, $v$ represents the walking speed of maintenance personnel, $h(X)$ represents the time required for node $X$ to recover, and $A$ is the adjustment coefficient used to generate reasonable greedy degree.

1. Select the next node for repair in each group according to greedy algorithm, and the next node is the node with the highest greedy degree in current. By analogy, the path of each group for maintenance can be obtained, and then the path of every individual can be obtained. Circulate $M/2$ times and finally $M/2$ individuals are generated and added to initial population. Figure 4 shows how this path generated based on greedy model.

### Intra-group head-to-head mutation operator

The head node of each-group path for repair in parent elite individuals are matched with the corresponding location node in offspring individuals, which generated by evolutionary operator proposed in Ref.\textsuperscript{16}. If matched, the location of next node for repair after head node for each-group path is seen as mutation point, and exchange nodes in the corresponding location of parent individual and offspring individual. At last, the
operator modifies duplicate data after exchange and generate new offspring population. The above constitutes intra-group head-to-head mutation operator. The schematic diagram of the operator is shown in Figure 5.

This paper considers the positive impact of solution obtained by greedy algorithm to individual, so the proposed IHMO retains partial optimal characteristics of improved initial population. When initial population has the same head node with evolved population, the second node brings a fairly positive effects on improvement of solution. So the exchange of the second node and keep other nodes unchanged will evenly improve the population and keep the randomness as well.

This paper proposes a combination of IHMO and the evolutionary operator proposed in [16] to improve solution quality. IHMO will eliminate some inferior individuals. When used together with improved initial population by greedy model, it can speed up convergence, control, and enhance certainty of population evolution, and weaken invalid randomness. As a result, the population evolves in a better direction.

Simulation and analysis

After system is damaged, the node location information, node task importance, node repair time, and number of spare parts required for node repair are shown in Appendix B. The performance of BGA, MCGA and IHGA proposed in this paper are compared and analyzed in three cases, including the number of node recovery, fitness value, system resilience, convergence, and stability. The parameters of three algorithms are shown in Table 1. The mutation probability of 0.38 is an empirical value used by the algorithm in order to increase the diversity of the offspring population. The Case 1 is to simulate a case that require recover system resilience as much as possible within a limited time after damaged, which is mainly used to verify feasibility and effectiveness of IHGA. The Case 2 and Case 3 are used to simulate resilience recovery of two situation. The one is with uneven distribution of damaged nodes after the disaster and balanced demand for node spare parts, the other is with unbalanced demand for node spare parts. They are mainly used to verify adaptability of the algorithm in more real and harsh conditions.

Case 1

The information of damaged nodes in Case 1 is shown in Table B1 in Appendix B. In the experiment, the number of maintenance personnel is set to 5, and the algorithm sets random path groups, and each group contains at least three nodes. The maintenance personnel repair in parallel and the time constraint of each group is $T_k < 2000$ s. Figure 6 shows location of the damaged nodes. The simulation results of three algorithms are shown in Figure 7. They are respectively recovery paths of BGA, recovery paths of MCGA and recovery paths of IHGA.

The data analysis is shown in Table 2. It is not difficult to find that IHGA is better than BGA and MCGA in algebras to reach optimal solution and the quality of optimal solution. In Table 2, BGA recovers 34 nodes for the simulation of system recovery path in Case 1, MCGA recovered 36 nodes, and IHGA recovered the largest number of nodes, reaching 37 nodes. Similarly, IHGA has the highest system task importance with the smallest algebra to reach optimal solution. This indicates that the IHGA proposed in this paper has made further improvements than BGA and MCGA.

Comparing the evolution process of three algorithms with optimal task importance, as shown in Figure 8.
The initial optimal task importance of IHGA is higher than that of BGA and MCGA, indicating that the method of improving initial population proposed can effectively improve initial solution of algorithm. At the same time, we find that in the process of population iteration, IHGA has the fastest convergence and takes much priority over BGA and MCGA to achieve optimal task importance.

The equation (1) is used to measure system resilience, in which system performance is measured by the importance of system task. The system resilience before and after recovery of three algorithms is shown in Table 3. In it, resilience improvement 1 is the improvement degree of IHGA and MCGA on the basis of BGA, also resilience improvement 2 is the degree of resilience improvement of IHGA on the basis of MCGA.

By comparison, it can be found that the system resilience using IHGA is the highest in Case 1. Compared with BGA, the resilience improvement of MCGA has increased 13.3%, and that of IHGA has increased 16%. And Compared to MCGA, IHGA makes 2.3% improvement on system resilience in case 1.

**Case 2**

In Case 2, the location of damaged nodes is uneven, as shown in Figure 6. The number of maintenance personnel is set to 5, and the algorithm sets random path grouping, but each group repairs three nodes at least. Besides considering constraints of time and spare parts, this paper discusses two cases of balanced and unbalanced node spare parts demand respectively. The data of spare parts in two cases are shown in Tables B1 and B2 in Appendix B. Maintenance personnel repair in parallel and time constraint of each group is reduced to $T_k < 1800\, \text{s}$, and spare parts constraints all meet $A < 14$, $B < 14$, $C < 14$. The demand of node spare parts is balanced, as the range of spare parts $A$, $B$, and $C$ is $(1, 3)$.

The simulation results of three algorithms are shown in Figure 9, which are recovery paths of BGA, recovery

---

**Table 1. Algorithm characteristics and parameters.**

| Algorithm | Initial population size | Initial population generation | Number of iterations | Coding method | Mutation probability | Evolutionary operator | Size of each small population |
|-----------|-------------------------|-------------------------------|---------------------|---------------|---------------------|---------------------------|-------------------------------|
| BGA       | 80                      | Random                        | 500                 | Single chromosome coding | 0.38                | General mutation operator | 8                             |
| MCGA      | 80                      | Random                        | 500                 | Multi-chromosome coding | 0.38                | Complex mutation operator | 8                             |
| IHGA      | 80                      | Greedy model generation and random | 500               | Multi-chromosome coding | 0.38                | Complex mutation operator and IHMO operator in group | 8                             |

---

**Figure 5.** Intra-group head-and-head mutation operator.

**Figure 6.** Node location.
Figure 7. Comparison of recovery paths in Case 1: (a) recovery paths of BGA, (b) recovery paths of MCGA, and (c) recovery paths of IHGA.

Table 2. Comparison of node data recovered by three algorithms.

| Algorithm | Total nodes | Number of recovered nodes | Total task importance | Task importance of recovery | Convergence iteration |
|-----------|-------------|----------------------------|-----------------------|-----------------------------|----------------------|
| BGA       | 50          | 34                         | 170                   | 128                         | 243                  |
| MCGA      | 50          | 36                         | 170                   | 144                         | 291                  |
| IHGA      | 50          | 37                         | 170                   | 148                         | 58                   |

Table 3. System resilience comparison of the three algorithms.

| Recovery algorithm | Performance of system recovery | Performance loss of system | System resilience | Resilience improvement 1 | Resilience improvement 2 |
|--------------------|--------------------------------|-----------------------------|-------------------|--------------------------|--------------------------|
| BGA                | 128                            | 170                         | 0.75              | -                        | -                        |
| MCGA               | 144                            | 170                         | 0.85              | 13.3%                    | -                        |
| IHGA               | 148                            | 170                         | 0.87              | 16%                      | 2.3%                     |
The data analysis is shown in Table 4. In it, BGA recovered 28 nodes for the simulation of system recovery path in Case 2, MCGA recovered 31 nodes, and IHGA recovered the largest number of nodes, reaching 33. Similarly, IHGA has the highest system task importance.

Comparing the evolution process of three algorithms with optimal task importance, as shown in Figure 10. The initial optimal task importance of IHGA is higher than that of BGA and MCGA, indicating that the method of improving initial population proposed in this paper can effectively improve initial solution. Similarly, we find that IHGA takes priority over BGA and MCGA to achieve optimal task importance in the process of population iteration. That shows the IHMO proposed in this paper further improves the optimal solution.

The equation (1) is used to measure system resilience, in which system performance indicator is measured by the importance of system task. The system

![Figure 8](image1.png)

**Figure 8.** The evolution curve of three algorithms in Case 1.

---

![Figure 9](image2.png)

**Figure 9.** Comparison of recovery paths in Case 2: (a) recovery paths of BGA, (b) recovery paths of MCGA, and (c) recovery paths of IHGA.
resilience before and after recovery using three algorithms is shown in Table 5.

By comparison, it can be found that the system resilience using IHGA is the highest in Case 2. Compared with BGA, the resilience improvement of MCGA has increased 20.3%, and that of IHGA has increased 23.4%. And Compared to MCGA, IHGA makes 2.6% improvement on system resilience in case 2.

Case 3

In Case 3, the node location information in case 3 is the same as cases 1 and 2, but the demand for node spare parts is unbalanced. The demand for spare parts A and B are both (1, 5), and that for spare parts C is (3, 7). The simulation results of the three algorithms are shown in Figure 11, which are the recovery paths of BGA, the recovery paths of MCGA, and the recovery paths of IHGA, respectively.

The data analysis is shown in Table 6. In it, BGA recovered 13 nodes for the simulation of system recovery path in Case 3, MCGA recovered 16 nodes, and IHGA recovered the largest number of nodes, reaching 17. Similarly, IHGA has the highest system task importance. Its convergence algebra is better than MCGA which means the IHGA has a good convergence and has the ability to avoid premature as BGA.

Comparing the evolution process of three algorithms with optimal task importance, as shown in Figure 12. The initial optimal task importance of IHGA is higher than that of BGA and MCGA in case 3, indicating that the method of improving initial population proposed in this paper can effectively improve initial solution of algorithm. Similarly, we find that IHGA takes complete priority over BGA and MCGA to achieve optimal task importance in the process of population iteration. The above demonstrates that the IHMO proposed in this paper further improves the optimal solution.

| Recovery algorithm | Total nodes | Number of recovered nodes | Total task importance | Task importance of recovery | Convergence algebra |
|--------------------|-------------|--------------------------|-----------------------|---------------------------|-------------------|
| BGA                | 50          | 28                       | 170                   | 109                       | 142               |
| MCGA               | 50          | 31                       | 170                   | 131                       | 57                |
| IHGA               | 50          | 33                       | 170                   | 135                       | 90                |

| Recovery algorithm | Performance value of system recovery | Performance value of system loss | System resilience | Resilience improvement | Resilience improvement |
|--------------------|--------------------------------------|----------------------------------|-------------------|------------------------|------------------------|
| BGA                | 109                                  | 170                              | 0.64              | –                      | –                      |
| MCGA               | 131                                  | 170                              | 0.77              | 20.3%                  | –                      |
| IHGA               | 135                                  | 170                              | 0.79              | 23.4%                  | 2.6%                  |

Table 4. Comparison of node data recovered by three algorithms.

Table 5. System resilience comparison of the three algorithms.

Table 6. Comparison of node data recovered by three algorithms.

Figure 10. The evolution curve of three algorithms in Case 2.
The resilience measurement method of equation (1) is used to calculate system resilience, in which system performance indicator is measured by the importance of system task. The system resilience data before and after recovery using three algorithms is shown in Table 7.

**Figure 11.** Comparison of recovery paths in Case 3: (a) recovery paths of BGA, (b) recovery paths of MCGA, and (c) recovery paths of IHGA.

**Figure 12.** The evolution curve of three algorithms in Case 3.

**Figure 13.** The improvement of three algorithm in three cases.

The resilience measurement method of equation (1) is used to calculate system resilience, in which system performance indicator is measured by the importance of system task. The system resilience data before and after recovery using three algorithms is shown in Table 7.
By comparison, it can be found that the system resilience after IHGA is the highest in wartime (Case 3). Compared with BGA, the resilience improvement of MCGA has increased 17.1%, and that of IHGA has increased 28.6%. And compared to MCGA, IHGA makes 9.8% improvement on system resilience in case 2. This shows that IHGA has better solution than MCGA, and the system resilience has been significantly improved in the condition of actual wartime, such as the reduction of given time, uneven distribution of damaged nodes, and the increasing and unbalanced demand for spare parts. The significant differences in case 3 reflect the obvious superiority of IHGA under harsh conditions.

Table 7. System resilience comparison of the three algorithms.

| Recovery algorithm | Performance value of system recovery | Performance value of system loss | System resilience | Resilience improvement | Resilience improvement |
|--------------------|-------------------------------------|----------------------------------|-------------------|------------------------|------------------------|
| BGA                | 59                                  | 170                              | 0.35              | –                      | –                      |
| MCGA               | 69                                  | 170                              | 0.41              | 17.1%                  | –                      |
| IHGA               | 76                                  | 170                              | 0.45              | 28.6%                  | 9.8%                   |

Table 8. Comparison of improvement of the three algorithms.

| Recovery algorithm | Case 1 (%) | Case 2 (%) | Case 3 (%) |
|--------------------|------------|------------|------------|
| BGA                | –          | –          | –          |
| MCGA               | 13.3       | 20.3       | 17.1       |
| IHGA               | 16         | 23.4       | 28.6       |
| IHGA-MCGA          | 2.3        | 2.6        | 9.8        |

By comparison, it can be found that the system resilience after IHGA is the highest in wartime (Case 3).

Figure 14. Fitness curve for 15 runs in three cases: (a) in case 1, (b) in case 2, and (c) in case 3.
Analysis of improvement

This paper lists the improvement on system resilience of three algorithm in three cases as Table 8 and Figure 13.

It can been seen from Table 8, in the three cases, the MCGA and IHGA are both better than BGA, and IHGA is better than MCGA and BGA. At the same time, the improvement increases with the complexity of case as in Figure 13.

In Figure 13, IHGA makes obvious improvement than BGA and MCGA and the improvement increases with more complexity and constrains of system. It shows the IHGA has the better performance in system resilience recovery and can be better applied in practice.

Analysis of stability

In three cases above, this paper respectively counts 15 running results of three algorithms. The detailed data is shown in Appendix A. Fitness curve after 15 runs in each case is shown in Figure 14, and the convergence iteration is in Figure 15.

The detailed data of Figures 14 and 15 is in shown in Table 9.

Table 9. Comparison table of algorithm stability in three cases.

| Test | Average of fitness | Variance of fitness | Average of convergence iterations |
|------|--------------------|---------------------|----------------------------------|
|      | BGA | MCGA | IHGA | BGA | MCGA | IHGA | BGA | MCGA | IHGA |
| Case 1 | 131.53 | 144.4 | 145.67 | 39.18 | 2.91 | 3.69 | 172.93 | 172.13 | 139.06 |
| Case 2 | 107 | 132.13 | 133.13 | 5.87 | 2.25 | 2.91 | 185.8 | 174.66 | 151.33 |
| Case 3 | 58.07 | 66.9 | 74.9 | 2.99 | 3.79 | 1.53 | 196.27 | 228.73 | 139.67 |
Through horizontal observation, it can be found in the table that IHGA is better than BGA and MCGA in average fitness and average number of convergence iterations. The variance of IHGA’s fitness value is smaller than BGA and close to MCGA. This illustrates that IHGA has better effects with optimal solution and good stability.

Through vertical observation, it can be found that the average fitness and the average of convergence iterations of IHGA are significantly improved compared with MCGA in case 3. Simultaneously, the variance of fitness is better than that of BGA and MCGA. This indicates that the stability, convergence and optimal solution of IHGA proposed are better than BGA and MCGA, especially in the harsh conditions of shorter repair time, more and unbalanced demands for spare parts.

**Conclusion**

This paper proposes an improved genetic algorithm (IHGA) to solve the problem in recovery path planning of system resilience, which optimizes initial population on greedy idea and designs intra-group head-to-head mutation operator (IHMO). Three cases demonstrate that the IHGA can obtain optimal recovery path under less iterations when solving recovery path planning of system resilience. Ultimately, it can effectively improve system resilience. By making the comparison with BGA and MCGA, it is found that the improved genetic algorithm-IHGA is superior to the other two algorithms in convergence, optimal solution, solution improvement, and algorithm stability. In addition, IHGA has more significant advantages especially in the harsh conditions of shorter repair time, more and unbalanced demands for spare parts, which can meet more stringent actual requirements, and has better practical effects in application.

**Author contributions**

Conceptualization, YuMei Wu and Zhen Li; methodology, YuMei Wu, Zhen Li, and Chen Xu Sun; software, Chen Xu Sun; validation, Chen Xu Sun and Zhen Li; formal analysis, YuMei Wu, Zhen Li, and Chen Xu Sun; investigation, ZhaoBin Wang and DongSheng Wang; resources, Zhengwei Yu; data curation, Zhen Li and Chen Xu Sun; writing—original draft preparation, Zhen Li and Chen Xu Sun; writing—review and editing, YuMei Wu and Zhen Li; supervision, YuMei Wu and Zhen Li; project administration, Zhen Li; funding acquisition, YuMei Wu and Zhen Li. All authors have read and agreed to the published version of the manuscript.

**Declaration of conflicting interests**

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

**Funding**

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research was funded by National Natural Science Foundation of China, grant number “61973282,” and by Jiangsu University of Science and Technology, Reliability and System Engineering Open Group (JRSOG) Open Fund, grant number 2020002.

**ORCID iD**

Zhen Li https://orcid.org/0000-0002-6743-4829

**References**

1. Allenby B and Fink JH. Social and ecological resilience: Toward inherently secure and resilient societies. *Science* 2000; 24(1): 347–364.
2. Holling CS. Resilience and stability of ecological systems. *Ann Rev Ecol Syst* 1973; 4: 1–23.
3. Rose A and Liao SY. Modeling regional economic resilience to disasters: A computable general equilibrium analysis of water service disruptions. *J Reg Sci* 2005; 45(1): 75–112.
4. Hosseini S, Barker K and Ramirez-Marquez JE. A review of definitions and measures of system resilience. *Reliab Eng Syst Saf* 2016; 145(1): 47–61.
5. Hosseini S and Barker K. Modeling infrastructure resilience using Bayesian networks: a case study of inland waterway ports. *Comput Ind Eng* 2016; 93(3): 252–266.
6. Pant R, Barker K and Zobel CW. Static and dynamic metrics of economic resilience for interdependent infrastructure and industry sectors. *Reliab Eng Syst Saf* 2014; 125(5): 92–102.
7. Shafieezadeh A and Burden LI. Scenario-based resilience assessment framework for critical infrastructure systems: case study for seismic resilience of seaports. *Reliab Eng Syst Saf* 2014; 132: 207–219.
8. Costella MF, Saurin TA and de Macedo Guimares LB. A method for assessing health and safety management systems from the resilience engineering perspective. *Saf Sci* 2009; 47(8): 1056–1067.
9. Haimes YY, Crowther K and Horowitz BM. Homeland security preparedness: Balancing protection with resilience in emergent systems. *Syst Eng* 2008; 11(4): 287–308.
10. Cui Q, Li J, Ran H, et al. Resilience measurement of command information system super-network based on mission capability. *J Command Control* 2017; 3(2): 137–143.
11. Jing Y, Wu Y and Fan N. Resilience approach for heterogeneous distributed networked unmanned weapon systems. *J Beijing Inst Technol* 2015; 24(2): 180–187.
12. Li Z, Jin C, Hu P, et al. Resilience-based transportation network recovery strategy during emergency recovery phase under uncertainty. *Reliab Eng Syst Saf* 2019; 188(8): 503–514.
13. Zhang Y, Liang J, Zheng B, et al. Key management scheme based on route planning of mobile sink in wireless sensor networks. *Sensors* 2016; 16(2): 170–187.
14. Zhang Z and Wei HH. Modeling interaction of emergency inspection routing and restoration scheduling for postdisaster resilience of highway–bridge networks. *J Infrastruct Syst* 2021; 27(1): 1–17.
15. Wang Z, Liu R, Zhu Y, et al. Improved genetic algorithm for solving TSP problem. Electron Meas Technol 2019; 42(23): 91–96.

16. Ye D, Liu G and He B. Multi-chromosome genetic algorithm for multiple traveling salesman problem. J Syst Simul 2019; 31(1): 36–42.

17. Song T, Huo X and Wu X. A two-stage method for target searching in the path planning for mobile robots. Sensors 2020; 20(23): 6919.1–6919.18.

18. Xin JF, Zhong JB, Yang FR, et al. An improved genetic algorithm for path-planning of unmanned surface vehicle. Sensors 2019; 19(11): 2640.1–2640.23.

19. Barker K, Ramirez-Marquez JE and Rocco CM. Resilience-based network component importance measure. Reliab Eng Syst Saf 2013; 117: 89–97.

20. Singh A and Baghel AS. A new grouping genetic algorithm approach to the multiple traveling salesperson problem. Software Comput 2009; 13(1): 95–101.

21. Wang YZ, Chen Y and Yu YY. Improved grouping genetic algorithm for solving multiple traveling salesman problem. J Electron Inf Technol 2017; 39(1): 198–205.

22. Hao K, Zhao J, Yu K, et al. Path planning of mobile robots based on a multi-population migration genetic algorithm. Sensors 2020; 20: 5873.1–5873.23.

**Appendix A**

**Table A1.** The data of 15 runs in case 1.

| Number of running | Fitness | Convergence iteration |
|-------------------|---------|-----------------------|
|                   | BGA     | MCGA | IHGA | BGA | MCGA | IHGA |
| 1                 | 129     | 140  | 142  | 196 | 127  | 244  |
| 2                 | 130     | 146  | 145  | 71  | 104  | 426  |
| 3                 | 129     | 144  | 146  | 37  | 43   | 44   |
| 4                 | 128     | 144  | 146  | 268 | 176  | 170  |
| 5                 | 126     | 142  | 145  | 257 | 102  | 54   |
| 6                 | 146     | 146  | 149  | 373 | 194  | 58   |
| 7                 | 127     | 146  | 143  | 64  | 456  | 40   |
| 8                 | 129     | 145  | 145  | 124 | 71   | 51   |
| 9                 | 130     | 143  | 145  | 65  | 261  | 71   |
| 10                | 129     | 143  | 144  | 408 | 51   | 86   |
| 11                | 132     | 145  | 147  | 437 | 48   | 193  |
| 12                | 148     | 146  | 148  | 161 | 120  | 151  |
| 13                | 129     | 145  | 146  | 16  | 473  | 109  |
| 14                | 129     | 145  | 149  | 48  | 266  | 19   |
| 15                | 132     | 146  | 145  | 69  | 90   | 370  |

**Table A2.** The data of 15 runs in case 2.

| Number of running | Fitness | Convergence iteration |
|-------------------|---------|-----------------------|
|                   | BGA     | MCGA | IHGA | BGA | MCGA | IHGA |
| 1                 | 108     | 132  | 134  | 295 | 241  | 203  |
| 2                 | 109     | 129  | 133  | 315 | 243  | 121  |
| 3                 | 111     | 133  | 133  | 366 | 296  | 178  |
| 4                 | 111     | 132  | 135  | 469 | 94   | 95   |
| 5                 | 104     | 130  | 132  | 149 | 101  | 367  |
| 6                 | 107     | 135  | 132  | 202 | 117  | 55   |
| 7                 | 108     | 130  | 134  | 43  | 75   | 174  |
| 8                 | 108     | 133  | 128  | 203 | 174  | 5    |
| 9                 | 104     | 132  | 133  | 208 | 118  | 53   |
| 10                | 104     | 134  | 133  | 76  | 98   | 94   |
| 11                | 106     | 132  | 134  | 248 | 158  | 70   |
| 12                | 106     | 133  | 132  | 38  | 446  | 191  |
| 13                | 103     | 132  | 135  | 70  | 170  | 216  |
| 14                | 109     | 132  | 135  | 47  | 177  | 229  |
| 15                | 107     | 133  | 134  | 58  | 112  | 219  |

**Table A3.** The data of 15 runs in case 3.

| Number of running | Fitness | Convergence iteration |
|-------------------|---------|-----------------------|
|                   | BGA     | MCGA | IHGA | BGA | MCGA | IHGA |
| 1                 | 57      | 67   | 76   | 296 | 455  | 207  |
| 2                 | 59      | 69   | 75   | 10  | 66   | 43   |
| 3                 | 62      | 66   | 76   | 326 | 102  | 30   |
| 4                 | 64      | 64   | 74   | 181 | 201  | 67   |
| 5                 | 58      | 69   | 75   | 360 | 134  | 41   |
| 6                 | 57      | 69   | 75   | 127 | 459  | 65   |
| 7                 | 58      | 66   | 71   | 47  | 82   | 22   |
| 8                 | 57      | 65   | 76   | 492 | 315  | 87   |
| 9                 | 56      | 69   | 75   | 10  | 302  | 189  |
| 10                | 60      | 69   | 75   | 277 | 371  | 175  |
| 11                | 57      | 69   | 75   | 17  | 158  | 285  |
| 12                | 59      | 64   | 74   | 204 | 128  | 149  |
| 13                | 56      | 67   | 76   | 220 | 132  | 151  |
| 14                | 57      | 67   | 76   | 188 | 269  | 145  |
| 15                | 57      | 64   | 75   | 189 | 257  | 439  |
### Appendix B

**Table B1.** Information about damaged nodes in case 1.

| Node number | Abscissa (m) | Ordinate (m) | Task importance | Repair time (s) |
|-------------|--------------|--------------|-----------------|-----------------|
| 1           | 30.4         | 46.9         | 2               | 200             |
| 2           | 46.8         | 46.9         | 2               | 310             |
| 3           | 80.6         | 46.3         | 3               | 430             |
| 4           | 106.4        | 28.1         | 3               | 220             |
| 5           | 106.4        | 61.9         | 3               | 240             |
| 6           | 115.6        | 61.9         | 3               | 270             |
| 7           | 149.0        | 30.0         | 4               | 310             |
| 8           | 151.0        | 69.4         | 2               | 300             |
| 9           | 182.4        | 75.0         | 3               | 190             |
| 10          | 182.4        | 61.9         | 5               | 190             |
| 11          | 182.4        | 5.6          | 3               | 380             |
| 12          | 185.4        | 66.6         | 5               | 330             |
| 13          | 187.0        | 71.3         | 5               | 230             |
| 14          | 194.6        | 66.6         | 5               | 280             |
| 15          | 205.2        | 62.9         | 5               | 230             |
| 16          | 205.2        | 66.6         | 5               | 240             |
| 17          | 205.2        | 59.1         | 5               | 380             |
| 18          | 205.0        | 59.0         | 6               | 270             |
| 19          | 220.4        | 5.6          | 1               | 270             |
| 20          | 240.2        | 69.4         | 2               | 320             |
| 21          | 253.3        | 26.3         | 1               | 270             |
| 22          | 273.6        | 61.9         | 3               | 270             |
| 23          | 276.6        | 61.9         | 3               | 300             |
| 24          | 281.2        | 61.9         | 3               | 270             |
| 25          | 285.8        | 61.9         | 3               | 220             |
| 26          | 290.4        | 15.0         | 3               | 270             |
| 27          | 295.0        | 15.0         | 3               | 350             |
| 28          | 149.0        | 46.9         | 4               | 250             |
| 29          | 148.0        | 38.0         | 6               | 360             |
| 30          | 133.8        | 13.1         | 1               | 330             |
| 31          | 155.1        | 13.1         | 2               | 360             |
| 32          | 171.8        | 13.1         | 2               | 430             |
| 33          | 199.1        | 5.6          | 1               | 380             |
| 34          | 214.4        | 13.1         | 2               | 450             |
| 35          | 222.0        | 5.6          | 1               | 270             |
| 36          | 226.6        | 13.1         | 2               | 200             |
| 37          | 252.4        | 5.6          | 1               | 250             |
| 38          | 223.4        | 66.6         | 5               | 280             |
| 39          | 223.4        | 63.1         | 5               | 240             |
| 40          | 229.5        | 66.6         | 5               | 190             |
| 41          | 235.6        | 70.1         | 5               | 300             |
| 42          | 235.6        | 63.1         | 5               | 190             |
| 43          | 152.0        | 69.4         | 4               | 250             |
| 44          | 174.8        | 60.0         | 4               | 270             |
| 45          | 187.0        | 60.0         | 4               | 310             |
| 46          | 199.1        | 60.0         | 4               | 430             |
| 47          | 211.3        | 60.0         | 4               | 450             |
| 48          | 223.4        | 60.0         | 4               | 200             |
| 49          | 255.4        | 67.5         | 4               | 150             |
| 50          | 264.5        | 18.8         | 4               | 350             |

**Table B2.** Parts and time required for repair in case 2.

| Node number | Part A | Part B | Part C | Repair time (s) |
|-------------|--------|--------|--------|-----------------|
| 1           | 1      | 2      | 1      | 200             |
| 2           | 1      | 3      | 1      | 310             |
| 3           | 2      | 3      | 3      | 430             |
| 4           | 2      | 2      | 1      | 220             |
| 5           | 3      | 2      | 1      | 240             |
| 6           | 2      | 3      | 1      | 270             |
| 7           | 1      | 3      | 1      | 310             |
| 8           | 2      | 2      | 2      | 300             |
| 9           | 3      | 1      | 1      | 190             |
| 10          | 3      | 1      | 1      | 190             |
| 11          | 2      | 2      | 3      | 380             |
| 12          | 2      | 1      | 3      | 330             |
| 13          | 1      | 1      | 2      | 230             |
| 14          | 1      | 2      | 2      | 280             |
| 15          | 1      | 1      | 2      | 230             |
| 16          | 3      | 2      | 1      | 240             |
| 17          | 2      | 3      | 2      | 320             |
| 18          | 3      | 2      | 1      | 270             |
| 19          | 3      | 1      | 2      | 270             |
| 20          | 2      | 3      | 1      | 270             |
| 21          | 2      | 3      | 1      | 320             |
| 22          | 2      | 3      | 1      | 270             |
| 23          | 2      | 2      | 2      | 300             |
| 24          | 2      | 3      | 1      | 270             |
| 25          | 2      | 3      | 1      | 270             |
| 26          | 2      | 3      | 1      | 270             |
| 27          | 2      | 3      | 1      | 270             |
| 28          | 2      | 3      | 1      | 270             |
| 29          | 2      | 3      | 1      | 270             |
| 30          | 2      | 3      | 1      | 270             |
| 31          | 2      | 3      | 1      | 270             |
| 32          | 2      | 3      | 1      | 270             |
| 33          | 2      | 3      | 1      | 270             |
| 34          | 2      | 3      | 1      | 270             |
| 35          | 2      | 3      | 1      | 270             |
| 36          | 2      | 3      | 1      | 270             |
| 37          | 2      | 3      | 1      | 270             |
| 38          | 2      | 3      | 1      | 270             |
| 39          | 2      | 3      | 1      | 270             |
| 40          | 2      | 3      | 1      | 270             |
| 41          | 2      | 3      | 1      | 270             |
| 42          | 2      | 3      | 1      | 270             |
| 43          | 2      | 3      | 1      | 270             |
| 44          | 2      | 3      | 1      | 270             |
| 45          | 2      | 3      | 1      | 270             |
| 46          | 2      | 3      | 1      | 270             |
| 47          | 2      | 3      | 1      | 270             |
| 48          | 2      | 3      | 1      | 270             |
| 49          | 2      | 3      | 1      | 270             |
| 50          | 2      | 3      | 1      | 270             |
Table B3. Parts and time required for repair in case 3.

| Node number | Part A | Part B | Part C | Repair time (s) |
|-------------|-------|-------|-------|-----------------|
| 1           | 1     | 3     | 5     | 570             |
| 2           | 5     | 3     | 6     | 730             |
| 3           | 4     | 3     | 6     | 710             |
| 4           | 3     | 2     | 6     | 640             |
| 5           | 3     | 5     | 4     | 630             |
| 6           | 3     | 3     | 7     | 770             |
| 7           | 3     | 3     | 6     | 690             |
| 8           | 2     | 5     | 4     | 610             |
| 9           | 5     | 4     | 3     | 540             |
| 10          | 1     | 4     | 7     | 780             |
| 11          | 5     | 4     | 7     | 860             |
| 12          | 1     | 1     | 3     | 310             |
| 13          | 4     | 2     | 4     | 500             |
| 14          | 2     | 4     | 5     | 640             |
| 15          | 2     | 4     | 4     | 560             |
| 16          | 1     | 5     | 4     | 590             |
| 17          | 1     | 1     | 5     | 470             |
| 18          | 1     | 3     | 7     | 730             |
| 19          | 2     | 5     | 7     | 850             |
| 20          | 1     | 1     | 3     | 310             |
| 21          | 1     | 4     | 5     | 620             |
| 22          | 3     | 5     | 7     | 870             |
| 23          | 4     | 1     | 3     | 370             |
| 24          | 1     | 5     | 3     | 510             |
| 25          | 4     | 1     | 3     | 370             |
| 26          | 1     | 2     | 5     | 520             |
| 27          | 5     | 4     | 4     | 620             |
| 28          | 1     | 2     | 6     | 600             |
| 29          | 5     | 5     | 5     | 750             |
| 30          | 4     | 3     | 3     | 470             |
| 31          | 2     | 1     | 7     | 650             |
| 32          | 4     | 2     | 3     | 420             |
| 33          | 5     | 1     | 6     | 630             |
| 34          | 4     | 3     | 5     | 630             |
| 35          | 3     | 4     | 7     | 820             |
| 36          | 2     | 2     | 4     | 460             |
| 37          | 4     | 5     | 4     | 650             |
| 38          | 2     | 1     | 4     | 410             |
| 39          | 5     | 1     | 5     | 550             |
| 40          | 1     | 2     | 3     | 360             |
| 41          | 4     | 3     | 3     | 470             |
| 42          | 3     | 1     | 5     | 510             |
| 43          | 2     | 3     | 6     | 670             |
| 44          | 4     | 5     | 7     | 890             |
| 45          | 2     | 2     | 7     | 700             |
| 46          | 5     | 2     | 5     | 600             |
| 47          | 4     | 5     | 3     | 570             |
| 48          | 1     | 5     | 3     | 510             |
| 49          | 1     | 4     | 7     | 780             |
| 50          | 1     | 5     | 4     | 590             |