

# Optimization of Path Planning for Construction Robots Based on Multiple Advanced Algorithms

Kang Tan

Department of Civil Engineering, Dalian University of Technology, Dalian, China

Email: [philosophyedu66@yahoo.com](mailto:philosophyedu66@yahoo.com)

**How to cite this paper:** Tan, K. (2018) Optimization of Path Planning for Construction Robots Based on Multiple Advanced Algorithms. *Journal of Computer and Communications*, 6, 1-13.  
<https://doi.org/10.4236/jcc.2018.67001>

**Received:** May 26, 2018

**Accepted:** July 14, 2018

**Published:** July 17, 2018

Copyright © 2018 by author and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

---

## Abstract

There are many processes involved in construction, it is necessary to optimize the path planning of construction robots. Most researches focused more on optimization algorithms, but less on comparative analysis based on the advantages and shortcomings of these algorithms. Therefore, the innovation of this paper is to analyze three advanced optimization algorithms (genetic algorithm, hybrid particle swarm algorithm and ant colony algorithm) and discuss how these algorithms can improve the optimization performance by adjusting parameters. Finally, the three algorithms are compared and analyzed to find an optimization algorithm that is suitable for path planning optimization of construction robots. The purpose of the optimization is to obtain the maximum benefit with the least cost and complete project in an efficient and economical way.

## Keywords

Path Planning, Construction Robots, Optimization Algorithm

---

## 1. Introduction

To develop intelligent construction robots, the navigation system that can provide efficient path planning algorithm is necessary. The purpose of path planning method for construction robot is to find the shortest and collision-free path from initial position to target position. Koo presented an improved bug-based algorithm which can produce an effective path in an unknown environment with both stationary and movable obstacles. The contributions, which make it possible to generate an effective and short path, are an improved method to select local directions, a reverse mode, and a simple leaving condition [1]. Soltani presented the application of path planning in construction sites according to multiple objectives. It quantitatively evaluates the performance of three optimization

algorithms (Dijkstra, A\*, and Genetic algorithms) that are used to find multi-criteria paths in construction sites based on transportation and safety-related cost. The accuracy of the path solution and the time complexity of the optimization algorithm are compared and analyzed. According to the criteria, planners will see the shortest and safest route with low risk and high reliability [2].

Soltani also presented a framework based on transportation costs, safety and reliability for construction path planning. He studied a fuzzy-based multi-objective optimization method to make optimal strategic decisions on the movement path of construction site, and make detailed decisions on the distance of the workplace [3]. Sivakumar investigated the potential of applying genetic algorithms to automate the path planning of cooperative construction manipulators. The basic premise of this work is that automating the different planning steps will contribute to more reliable plans and thus promote the usage of cooperative manipulator. Two methodologies have been proposed using the concept of Configuration Space (C-Space) technique in conjunction with the genetic search [4].

Nieuwenhuisen described a new robotic method that can calculate a smooth, collision-free, and high-quality path map. This roadmap can be used to get optimal paths for robots. He also described the application of this technique for planning the movement of entity groups and creating smooth camera movements through the environment [5]. Lee presented a new sensor-based path planning algorithm, which is designed for an automated construction equipment (ACE). It is based on the practical assumptions that the robot can measure instantaneous velocity of obstacles in a range of vision and has memory of generated path from tracking point. The ACE algorithm guarantees reachability in an unknown environment with multiple moving obstacles and composite obstacles [6].

Lu has developed a practical method to solve the basic problems and limitations of existing resource scheduling methods by using the critical path method (CPM). The proposed method is referred to as the resource activity critical path method (RACPM), where resource dimensions other than activity and time are highlighted in project scheduling. By running RACPM under different condition, we can study the impact of various resources on project, which leads to a comprehensive schedule that provides a timetable for establishing, estimating, and controlling budgets [7]. Stouffs developed rules-based simulation programs for building construction. According to the specified plan, the program generates and simulates the motion path of each robot, which avoids obstacles and combines interactions, safety and other considerations [8].

Chang developed an automatically and efficiently plan with steps. The first step is to convert the crane installation site into configuration space, including the crane's load capacity and obstacles. The second step is to find an availability path in configuration space by using the probabilistic roadmap method (PRM). Three tests were conducted in this study to verify the behavior of the proposed

method. The results show that the proposed method can produce an effective installation path to operate and is suitable for crane installations, helping engineers to verify planning decisions [9]. Lei proposed a method based on robot motion planning to solve crane path planning problems. The proposed method performs a two-dimensional path planning of the crane and considers the rotation of the lifting object during its movement. It has been implemented in computer to provide a user-friendly interface to help practitioners perform collision-free path planning and check the feasibility of different stages of the project path [10].

From above review, we can conclude that most researches focused more on optimization algorithms, but less on comparative analysis based on the advantages and disadvantages of these algorithms. Therefore, the innovation of this paper is to analyze three optimization algorithms and discuss how these algorithms can improve the optimization performance by adjusting parameters. In this paper, a specific case has been analyzed, we need to choose a shortest path that goes through all the points. Finally, the three algorithms are compared and analyzed to find an optimization algorithm that is suitable for path planning optimization of construction robots.

## 2. Genetic Algorithm

### 2.1. The Principle of Genetic Algorithm

In computer science and operations research, genetic algorithm (GA) is a method inspired by the natural selection process and belongs to a larger class of evolutionary algorithms (EA). Genetic algorithms are often used to generate high-quality solutions for optimization and search problems by relying on bio-inspired operators such as mutation, crossover, and selection [11]. In genetic algorithms, a set of candidate solutions to optimization problems (called individual, creatures or phenotypic) evolve into a better solution. Each candidate solution has a set of attributes that can be mutated and changed (its chromosome or genotype). Traditionally, solutions are represented as strings in binary form, but other encodings are also possible. The implementation of genetic algorithms begins with a set of typical random chromosomes. These structures are then evaluated and assigned reproductive opportunities, providing better chromosomes with more reproductive opportunities than the chromosomes of poor solutions [12].

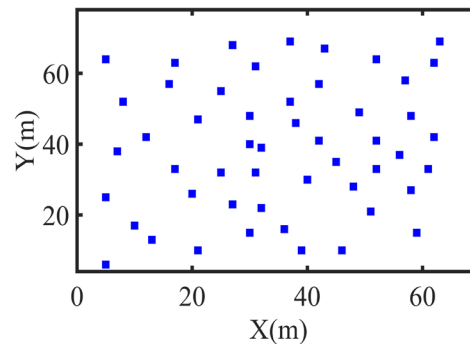
Evolution usually begins with a randomly generated set of individuals and is an iterative process called generations. In each generation, assess the fitness of every individual in the population. Adaptability is usually the value of the objective function that solves the optimization problem. More suitable individuals are randomly selected from the current population, and each individual's genome is modified (recombinant and possibly randomly mutated) to form a new generation. Then use the next generation of candidate solutions in the next iteration of the algorithm. Typically, the algorithm terminates when it produces the maxi-

imum generations or meets a satisfactory level of fitness.

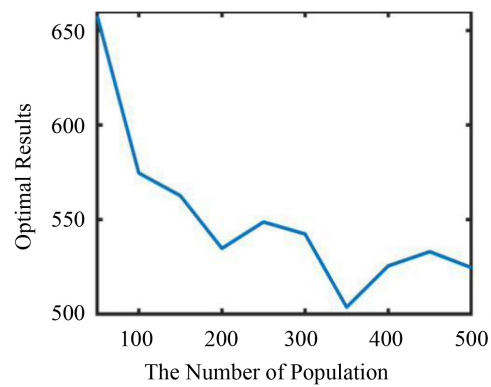
The first step of genetic algorithm is population initialization. Since the genetic algorithm cannot directly deal with the parameters of problem, the feasible solution to the problem to be solved must be represented as a chromosome or an individual in the genetic space through coding. Common coding methods are grey coding, real coding, and structural coding. The real number coding does not have to be converted numerically, and the genetic algorithm operation can be performed directly on the expression of the solution. This article uses real coding to define each chromosome as real variable. Secondly, the fitness function is criterion to distinguish individual good from bad in a group, and is the only basis for natural selection. Generally, it is obtained by transforming an objective function. This article is to find the maximum value of the function as the individual fitness value. The larger the value of individual function is, the better the fitness is. Thirdly, the selection operation is to select a good individual from the old group with a certain probability to form a new group to multiply next generation of individuals. The probability that the individual is selected is related to the fitness value. The higher the individual fitness is, the greater the probability of being selected is. There are various methods for selecting the genetic algorithm, such as roulette method and competition game method. In this paper, the roulette method is adopted. Fourthly, cross operation refers to randomly selecting two individuals from population and transferring excellent genetic characteristics to substrings through the exchange combination of two chromosomes to generate new excellent individuals. Since individuals use real numbers, the crossover method uses real number method. Finally, the last step of genetic algorithm is mutation operation. The purpose of the mutation operation is to maintain the diversity of the population. The mutation operation selects one individual from the population and mutates to produce a better individual.

## 2.2. Optimal Results of Genetic Algorithm

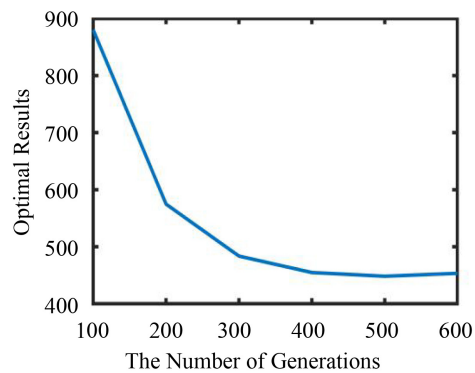
In this paper, a specific case is analyzed, we need to choose a shortest path that goes through all the points which are showed in **Figure 1**. There are five factors (the number of population, the number of generations, cross possibility, mutation possibility and generation gap) affecting the optimization performance of genetic algorithm. From **Figure 2**, we can see that as the value of population increases, the optimal results decrease and there will be some fluctuations in this process, indicating that the results are closer to optimal solution, so we can increase the value of population to improve the optimized performance. From **Figure 3**, we can see that as the value of generations increases, the optimal results decrease, indicating that the results are closer to optimal solution. Therefore, the performance of optimization can be improved by increasing the value of generations. From **Figure 4**, we can see that when the cross possibility is 0.5, the algorithm works best, and other values will reduce the optimization performance. From **Figure 5**, we can see that when the mutation possibility is 0.05, the



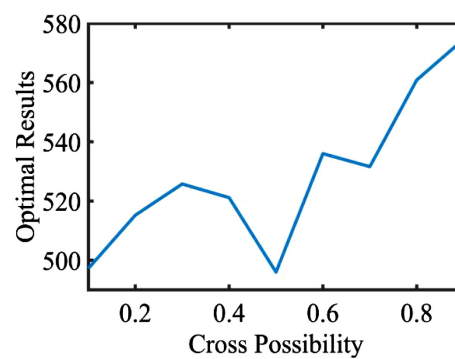
**Figure 1.** Points distribution.



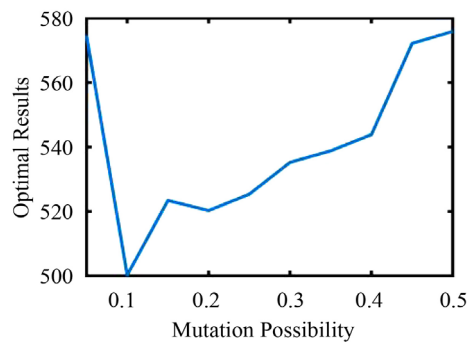
**Figure 2.** Population's effects on results.



**Figure 3.** Generations' effects on results.



**Figure 4.** Cross possibility's effects on results.



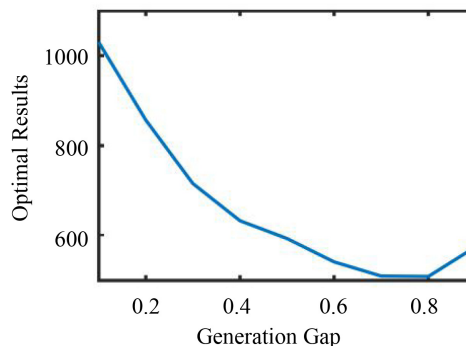
**Figure 5.** Mutation possibility's effects on results.

algorithm works best, and other values will reduce the optimized performance. From **Figure 6**, it can be seen that as the generation gap increases, the results continuously decrease, indicating that the results at this time are closer to the optimal solution. Therefore, the performance of optimization can be improved by increasing the value of generation gap.

### 3. Hybrid Particle Swarm Algorithm

#### 3.1. The Principle of Hybrid Particle Swarm Algorithm

Particle swarm optimization (PSO) originated from simulating the social behavior of birds, which was developed by Kennedy and Eberhart. In particle swarm optimization algorithm, each particle flies in the search space, and its speed is adjusted by its own flight memory and companion's flight experience. All particles have fitness value determined by fitness function [13]. The main idea of the hybrid particle swarm algorithm is to integrate the genetic algorithm (GA) with the particle swarm algorithm (PSO). It consists of three major operators: enhancement, crossover and mutation. The first operator is enhancement. The enhancement operation attempts to simulate the maturation in the natural world, and the individuals become more suitable for the environment after changing. In addition, by using these enhanced elites as parents, the offspring will develop better than the original elite. In the PSO, the same generation promotes themselves based on their own private perceptions and social interactions. The second operator is crossover. In order to cultivate outstanding individuals, parents are selected from among the elites in the crossover operation. The adoption of crossover can be seen as an elite crossover to improve search capabilities. In hybrid particle swarm algorithm, in the case of the same generation of work, the crossover operation introduces the concept of evolutionary evolution and the survival of the social adaptation test. The third operator is mutation. In hybrid particle swarm optimization, mutations occur in crossover operations. Thus, mutation is operator in which genes are randomly altered so that new genes can be introduced into the population. However, we should use mutations with caution because it is random search operator. Uniform mutations are used here, which is randomly selected mutated genes from the corresponding search interval [14].



**Figure 6.** Generation gap's effects on results.

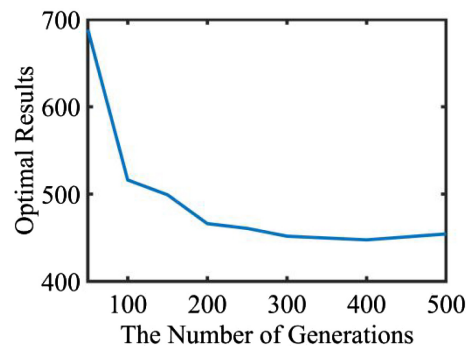
### 3.2. Optimal Results of Hybrid Particle Swarm Algorithm

In this paper, a specific case is analyzed, we need to choose a shortest path that goes through all the points which are showed in **Figure 1**. There are two factors (the number of generations and the number of individuals) affecting the optimization performance of hybrid particle swarm algorithm. From **Figure 7**, we can see that as the value of generations increases, the optimal results decrease rapidly at the initial stage and slowly at the latter stage. This indicates that the results are closer to the optimal solution. Therefore, with the value of generations increases, the optimized performance improves. From **Figure 8**, we can see that as the value of individuals increases, the optimized value decreases continuously, and it declines quickly in the early stage and slowly in the later stage. This shows that the results are closer to the optimal solution, so we can increase the value of individuals to improve the optimized performance.

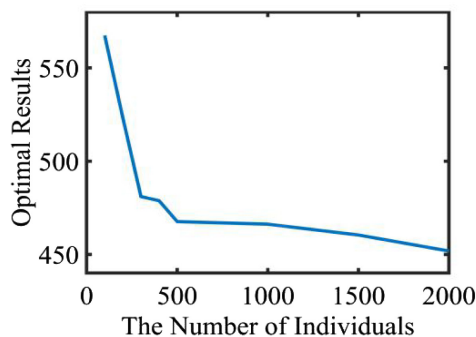
## 4. Ant Colony Algorithm

### 4.1. The Principle of Ant Colony Algorithm

The ant colony algorithm (ACO) is a method to simulate the behavior of ant foraging. It solves traveling salesman and quadratic distribution problems. Ants are social insects that behave more like group than individual [15]. The ant colony algorithm uses simple agent called ant to iteratively construct a solution to the optimization problem. Ant's solution is guided by artificial pheromone trajectories and problem-specific heuristics. Basically, for optimization problem, the pheromone indicates the strength of the ant of the solution component, and the strength is determined based on the contribution of each solution component in the objective function. A single ant builds a complete solution by starting from an empty solution and adding components iteratively until a complete solution is built. After establishing a complete solution, each ant provides feedback on the solution by placing pheromones on each solution component. Typically, solution components are used as part of better solutions and ants are used in many iterations to get more pheromone, and they are more likely to be used in future iterations [16]. The ant colony algorithm mainly consists of the following four steps. The first step is initialization. Set the initial population of colonies



**Figure 7.** Generations' effects on results.



**Figure 8.** Individuals' effects on results.

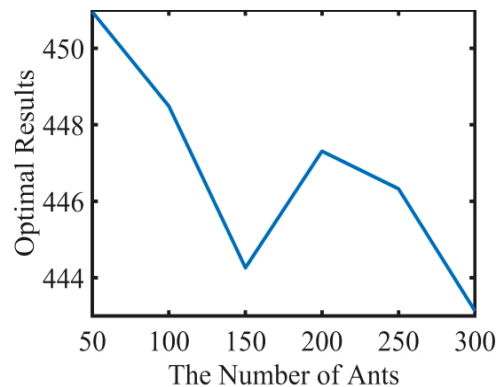
and pheromone trails. Randomly place the starting nodes of all ants. The second step is solution construction. Taking into account heuristic information dependence and problem path-tracking advantages, each ant chooses the next node that will not move with probability. Repeat this step until the solution building is complete. The third step is to update the path. The solution is evaluated according to the quality of the solution and stores the pheromone in the solution path. The better the solution is, the greater the amount of pheromone deposition is. The fourth step is the evaporation of pheromones. At the end of the iteration, in order to build a complete solution, the pheromone path is reduced by a constant factor [17].

#### 4.2. Optimal Results of Ant Colony Algorithm

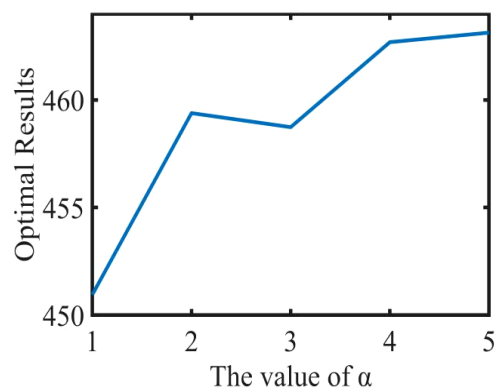
In this paper, a specific case is analyzed, we need to choose a shortest path that goes through all the points which are showed in **Figure 1**. There are six factors (the number of ants, the value of  $\alpha$ , the value of  $\beta$ , the value of  $\rho$ , the value of  $Q$ , max iteration) affecting the optimization performance of ant colony algorithm. The value of  $\alpha$  represents the importance factor of pheromone, the value of  $\beta$  represents the importance factor of heuristic function, the value of  $\rho$  is the pheromone evaporation coefficient, the value of  $Q$  is the factor of total pheromone release. From **Figure 9**, we can see that as the number of ants increases, the optimal results decrease, but there will be some fluctuations in the process, indicating that the results at this time are closer to the optimal solution. Therefore,



increasing the number of ants can be used to improve optimized performance. From **Figure 10**, we can see that as the value of  $\alpha$  increases, the optimal results keep rising and there will be some fluctuations in this process, which means that the results at this time are getting far away from the optimal solution. Therefore, the optimized performance can be improved by reducing the value of  $\alpha$ . From **Figure 11**, we can see that with the increase of  $\beta$ , the optimization result decreases rapidly at the initial stage, and at this time, the optimal solution is continuously approached. When the value of  $\beta$  is close to 4, the optimal solution is obtained. As the value of  $\beta$  continues to increase, the optimal results will continue to increase, gradually deviating from the optimal solution. From **Figure 12**, we can see that as the value of  $\rho$  increases, the optimal results decrease rapidly at the initial stage, and at this point, the optimal solution is gradually approached. When the value of  $\rho$  is 0.3, the optimal solution is obtained. With the value of  $\rho$  continuing to increase, the optimal results will continue to increase, gradually deviating from the optimal solution. From **Figure 13**, we can see that with the increase of  $Q$ , the optimal results decline quickly at the initial stage, which are close to the optimal solution at this time. When the value of  $Q$  is 3, the optimal solution is obtained. As the value of  $Q$  increases, the optimal results will increase, gradually deviating from the optimal solution. From **Figure 14**, we can see that as the max iteration increases, the optimization results decrease rapidly



**Figure 9.** Ants number's effects on results.



**Figure 10.**  $\alpha$ 's effects on results.

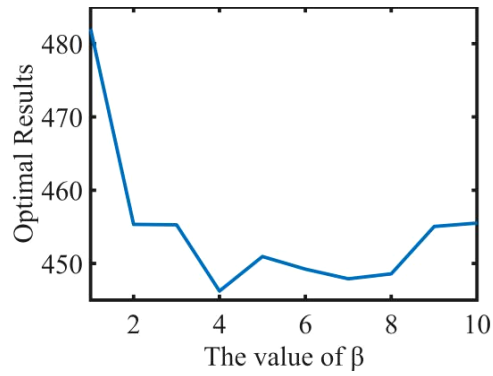


Figure 11.  $\beta$ 's effects on results.

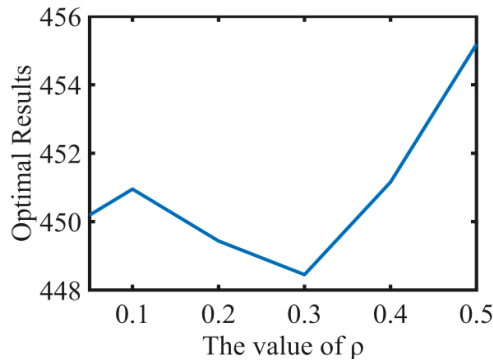


Figure 12.  $\rho$ 's effects on results.

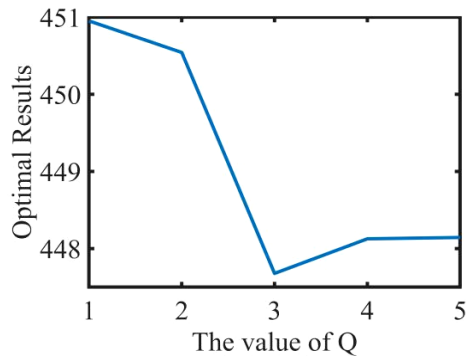
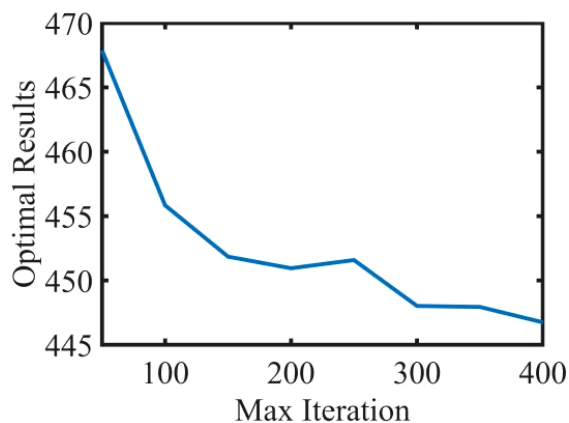


Figure 13.  $Q$ 's effects on results.

at an early stage, and then gradually tend to be flat. At this time, the optimal results are close to the optimal solution. Therefore, in order to obtain better optimal results, the max iteration should be increased.

### 5. Comparison of Different Optimization Algorithms

In order to choose suitable algorithm, we need to compare the optimization performance of the three algorithms based on their optimal results and convergence speed. Since the goal of this paper is to select the shortest path for robot construction, the smaller the optimal result is, the better the optimization performance is. Firstly, comparing the optimal results of the three algorithms. The

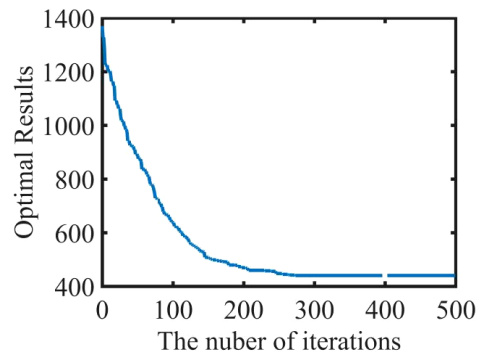


**Figure 14.** Max iteration's effects on results.

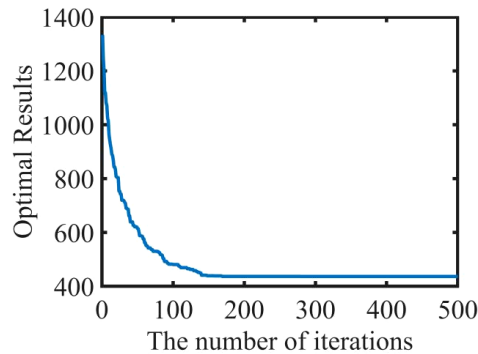
optimal result of genetic algorithm is 440.818, the optimal result of hybrid particle algorithm is 436.482, and the optimal result of ant colony algorithm is 441.253. It can be concluded that the optimal result of hybrid particle algorithm is the smallest and its optimization performance is the best. Secondly, compare the convergence speed of the three algorithms. From **Figure 15**, we can see that for genetic algorithm, as the number of iterations increases, the optimal results decrease rapidly in [0, 100], decrease smoothly in [100, 300], and remain unchanged in [300, 500]. From **Figure 16**, we can see that for hybrid particle algorithm, the optimal results decrease quickly in [0, 50], as the number of iterations increase, decrease more gradually in [50, 150], and remain unchanged in [150, 500]. From **Figure 17**, we can see that for ant colony algorithm, as the number of iterations increases, the optimal results decrease rapidly in [0, 30], decline slowly in [30, 300] with sudden changes in some places, and remain unchanged in [300, 500]. After comparing convergence speed of three algorithms, we can see that ant colony algorithm converges fastest, followed by hybrid algorithm, and genetic algorithm converges slowest. It can be seen from the figure that the convergence speed of ant colony algorithm and the hybrid algorithm are relatively close and are better than genetic algorithm. In summary, after considering the two key factors, this paper recommends to use hybrid particle algorithm, which can achieve better optimization performance.

## 6. Conclusion

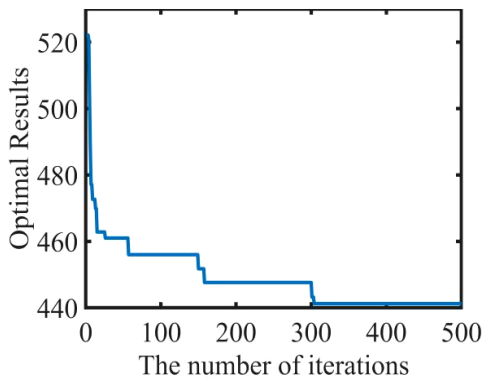
The optimization of path planning for construction robots will improve the efficiency of construction process and save costs. Many researchers have proposed optimization methods to solve this problem, but there is less research on comparative analysis based on the advantages and shortcomings of these algorithms. Therefore, this paper analyzes the three optimization algorithms and discusses how these algorithms can improve the optimization performance by adjusting parameters. Finally, the three algorithms are compared and analyzed to find an optimization algorithm that is suitable for solving this problem. Based on above



**Figure 15.** Genetic algorithm.



**Figure 16.** Hybrid particle algorithm.



**Figure 17.** Ant colony algorithm.

analysis, this paper recommends to use hybrid particle algorithm to solve the problem of path optimization for construction robots.

## References

- [1] Koo, K.J., Russell, J.S. and Kim, S.K. (2003) Construction Robot Path-Planning for Earthwork Operations. *Journal of Computing in Civil Engineering*, **17**, 97-104. [https://doi.org/10.1061/\(ASCE\)0887-3801\(2003\)17:2\(97\)](https://doi.org/10.1061/(ASCE)0887-3801(2003)17:2(97))
- [2] Soltani, A.R., Tawfik, H., Goulermas, J.Y. and Fernando, T. (2002) Path Planning in Construction Sites: Performance Evaluation of the Dijkstra A\* and GA Search Algorithms. *Advanced Engineering Informatics*, **16**, 291-303. [https://doi.org/10.1016/S1474-0346\(03\)00018-1](https://doi.org/10.1016/S1474-0346(03)00018-1)

- [3] Soltani, A.R. and Fernando, T. (2004) A Fuzzy Based Multi-Objective Path Planning of Construction Sites. *Automation in Construction*, **13**, 717-734. <https://doi.org/10.1016/j.autcon.2004.04.012>
- [4] Sivakumar, P., Varghese, K. and Babu, N.R. (2000) Path Planning of Cooperative Construction Manipulators Using Genetic Algorithms. *Isarc Proceedings*, Taipei, 18-20 September 2000, 1-6. <https://doi.org/10.22260/ISARC2000/0070>
- [5] Nieuwenhuisen, D., Kamphuis, A., Mooijekind, M. and Overmars, M.H. (2004) Automatic Construction of High Quality Roadmaps for Path Planning. UU-CS, Utrecht.
- [6] Lee, S. and Adamsb, T.M. (1999) A Path Planning Algorithm for Automated Construction Equipment. *Proceedings of the 16th ISARC*, Madrid, 22-24 September 1999, 543-548.
- [7] Lu, M. and Li, H. (2003) Resource-Activity Critical-Path Method for Construction Planning. *Journal of Construction Engineering & Management*, **129**, 412-420. [https://doi.org/10.1061/\(ASCE\)0733-9364\(2003\)129:4\(412\)](https://doi.org/10.1061/(ASCE)0733-9364(2003)129:4(412))
- [8] Stouffs, R., Krishnamurti, R., Lee, S. and Oppenheim, I.J. (1994) Construction Process Simulation with Rule-Based Robot Path Planning. *Automation in Construction*, **3**, 79-86. [https://doi.org/10.1016/0926-5805\(94\)90035-3](https://doi.org/10.1016/0926-5805(94)90035-3)
- [9] Chang, Y.C., Hung, W.H. and Kang, S.C. (2012) A Fast Path Planning Method for Single and Dual Crane Erections. *Automation in Construction*, **22**, 468-480. <https://doi.org/10.1016/j.autcon.2011.11.006>
- [10] Lei, Z., Behzadipour, S., Al-Hussein, M. and Hermann, R. (2011) Application of Robotic Obstacle Avoidance in Crane Lift Path Planning. *Proceedings of the 28th ISARC*, Seoul, Korea, 29 June-2 July 2011, 292-297. <https://doi.org/10.22260/ISARC2011/0052>
- [11] Mitchell, M. (1998) An Introduction to Genetic Algorithms. MIT Press, Cambridge.
- [12] Whitley, D. (1994) A Genetic Algorithm Tutorial. *Statistics and Computing*, **4**, 65-85. <https://doi.org/10.1007/BF00175354>
- [13] Li, S., Wu, X. and Tan, M. (2008) Gene Selection Using Hybrid Particle Swarm Optimization and Genetic Algorithm. *Soft Computing*, **12**, 1039-1048. <https://doi.org/10.1007/s00500-007-0272-x>
- [14] Juang, C.F. (2004) A Hybrid of Genetic Algorithm and Particle Swarm Optimization for Recurrent Network Design. *IEEE Transactions on Systems Man & Cybernetics Part B Cybernetics A Publication of the IEEE Systems Man & Cybernetics Society*, **34**, 997-1006. <https://doi.org/10.1109/TSMCB.2003.818557>
- [15] Shelokar, P.S., Siarry, P., Jayaraman, V.K. and Kulkarni, B.D. (2007) Particle Swarm and Ant Colony Algorithms Hybridized for Improved Continuous Optimization. *Applied Mathematics & Computation*, **188**, 129-142. <https://doi.org/10.1016/j.amc.2006.09.098>
- [16] Rajendran, C. and Ziegler, H. (2004) Ant-Colony Algorithms for Permutation Flowshop Scheduling to Minimize Makespan/Total Flowtime of Jobs. *European Journal of Operational Research*, **155**, 426-438. [https://doi.org/10.1016/S0377-2217\(02\)00908-6](https://doi.org/10.1016/S0377-2217(02)00908-6)
- [17] Wang, J.F., Liu, J.H. and Zhong, Y.F. (2005) A Novel Ant Colony Algorithm for Assembly Sequence Planning. *International Journal of Advanced Manufacturing Technology*, **25**, 1137-1143. <https://doi.org/10.1007/s00170-003-1952-z>