A Path Planning Method Based on Improved Single Player-Monte Carlo Tree Search

YU-WEI XIA1, CHAO YANG1,2,3, AND BING-QIU CHEN1

1School of Computer Science and Information Engineering, Hubei University, Wuhan 430062, China
2Hubei Education Informatization Engineering Technology Research Center, Wuhan 430062, China
3Hubei Key Laboratory of Applied Mathematics, Faculty of Mathematics and Statistics, Hubei University, Wuhan 430062, China

Corresponding author: Chao Yang (stevenyc@hubu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61977021, in part by the Major Technological Innovation Special Project of Hubei Province under Grant 2019ACA144, in part by the Open Project of Hubei Key Laboratory of Applied Mathematics under Grant HBAM201902, and in part by the Intelligent Information Processing and Real Time Industrial System Hubei Provincial Key Laboratory Open Fund Project under Grant znxx2018MS05.

ABSTRACT With the thorough application and development of Intelligent Logistics System, the Path Planning Problem, which is one of the key problems of logistics scheduling, is accelerating to the direction of dynamics, diversification and complexity. In order to better adapt to the new changes, this paper proposes a path planning method based on improved SP-MCTS (Single-Player Monte Carlo Tree Search) algorithm. Unlike the previous heuristic algorithms, SP-MCTS does not completely depend on parameters, and can more effectively solve variable combinatorial optimization problems. Aiming at the problem of path planning, this paper enhances the simulation process of SP-MCTS by clonal selection algorithm, in order to improve the algorithm’s accuracy of locating high-potential branches, and further improve the accuracy of the decisions made by SP-MCTS. Experiments show that the method proposed in this paper is less affected by the parameters and can better solve the changing path planning problem. Compared with the existing path planning methods and original SP-MCTS algorithm, it can solve the changeable path planning problem more stably, faster and more accurately.

INDEX TERMS Single player-Monte Carlo tree search, clonal selection algorithm, path planning.

I. INTRODUCTION With the rapid popularization of 5G technology, the Industrial Internet is about to usher in the era of Enterprise Business Capacity (EBC). At this stage, Intelligent Logistics System will become an important part of the Industrial Internet, and logistics scheduling problems are becoming more and more complicated. Enterprise development is faced with the huge challenges of reducing logistics dispatching costs and improving logistics dispatching efficiency, and path planning is an indispensable part of logistics dispatching. Therefore, researching path planning problem and providing an efficient solution have important significance on the development of logistics enterprises.

In recent years, the methods for solving path planning problems are mostly heuristic intelligent algorithms [1]. This type of algorithm refers to solving path planning problems by simulating some natural phenomena or biological behavior processes, such as particle swarm optimization [2], ant colony algorithm [3], genetic algorithm [4], clone selection algorithm [5], etc. Guangsheng and Wusheng [6] proposed an improved particle swarm optimization algorithm to optimize the path planning of mobile robots. A new adaptive learning mechanism was introduced into the algorithm to adaptively select the most suitable search strategy at different stages of the optimization process. Literature [7] proposed an improved artificial bee colony algorithm for mobile robot path planning optimization, using Bezier curves to describe the path, and converting the path optimization problem into the location optimization problem of generating Bezier curve points. Literature [8] proposed a multi-population ant colony system algorithm, which is used for the path planning of food distribution services. Literature [9] proposed an improved invasive weed algorithm to optimize the path planning of multi-robots. The improved algorithm combines the invasive weed algorithm with reverse learning (OBL) to initialize the population, and uses breeding jumps to speed up the convergence speed of invasive weed algorithm. Literature [10] proposes an improved genetic algorithm with adaptive adjustment of crossover probability, which performs better in the
path planning of driverless vehicles. Literature [11] proposed a parallel multi-start and multi-target clonal selection algorithm for solving multi-objective Energy Reduction multi-depot vehicle routing problem. These heuristic algorithms have their own advantages and disadvantages, but they all depend on some parameters, the setting of related parameter values greatly affects the algorithm’s ability to solve problems, and can’t flexibly cope with the variable path planning problems.

The advantages and disadvantages of various metaheuristics in solving path planning are shown in Table 1.

| Algorithm                     | Algorithm introduction                                                                 | Algorithm advantages                                      | Algorithm disadvantages                                      |
|-------------------------------|----------------------------------------------------------------------------------------|------------------------------------------------------------|--------------------------------------------------------------|
| Particle swarm optimization   | An intelligent bionic optimization algorithm for simulating flight and foraging behavior of birds |
| Artificial bee colony algorithm | Starting from the behavior of individuals in the bee colony, imitating the bee colony foraging process in nature to find the optimal solution to the problem |
| Ant colony optimization       | Intelligent bionic algorithm based on ant foraging behavior                               |
| Invasive weed optimization    | A random search algorithm that simulates the process of weed colonization in nature |
| Genetic algorithms            | A highly parallel, random, self-organizing and adaptive bionic optimization algorithm based on "survival of the fittest" |
| Clonal selection algorithm    | Selection mechanism of antibody clones of human immune cells                              |

The above metaheuristic algorithms can also be generally divided into the following categories: inspired by the behavior of biological groups, representative algorithms include particle swarm optimization, artificial bee colony algorithm, ant colony optimization; inspired by biological evolution, representative algorithms include invasive weed optimization, genetic algorithms; inspired by the structure or organization of organisms, representative algorithms include clonal selection algorithms, etc.

As can be seen from the Table 1, algorithms inspired by the behavior of biological groups generally converge slowly. Although the algorithm inspired by biological evolution has good robustness, it is easy to fall into local optimum, and the quality of the solution obtained for the path planning problem is poor. Algorithms inspired by biological structure or organization have good adaptability, but may be premature. Therefore, it is meaningful to consider a new method that can solve different kinds of path planning problems.

SP-MCTS is a type of single-player strategy research algorithm. The study of SP-MCTS in literature [12] shows that it has the advantages of not completely relying on parameters and less dependence on artificial experience, and gradually serves as a heuristic algorithm to solve combinatorial optimization problems. The search process of SP-MCTS is mainly divided into four steps: selection, expansion, simulation and back-propagation. The existing SP-MCTS use greedy algorithm in simulation process [12]–[15]. However, the greedy algorithm is easy to fall into the local optimum, which means it has great limitations, cannot guarantee the accuracy of the simulation results, and affects the decision-making ability of the entire algorithm.

Therefore, we propose a novel improved SP-MCTS method for path planning problem. Aiming at the problem that the original method is easy to fall into local optimization in simulation stage, we use clonal selection algorithm to improve SP-MCTS, raising the search ability and decision accuracy of the SP-MCTS. The comparison experiment with existing methods proves that our method can better solve the variable path planning problem, with higher accuracy and less parameters influences, and has more advantages in stability and time consumption.
The rectangle in Figure 1 represents the specific environment or specific limitations of path planning problems. Special environments such as marine path planning environment, environment applied to GPS system road planning, etc., and special restrictions such as maximum load limitation in logistics transportation, maximum vehicle travel distance limitation, etc. The objective function refers to the target requirements of the sought path, such as the shortest path and the least time-consuming path. Then the path that optimizes the value of the objective function is obtained according to the algorithm.

Consequently, the path planning problem is to find a walking path that makes the objective function obtain the optimal value under the specific environment or restricted conditions.

B. RELATED WORK

The SP-MCTS algorithm is a decision-making algorithm based on simulation results. The algorithm is divided into four steps: selection, expansion, simulation, and back-propagation. Its principle [17] can be shown in Figure 2.

FIGURE 2. SP-MCTS algorithm principle: selection, expansion, simulation and back-propagation.

1) WHY WE CHOOSE SP-MCTS TO SOLVE ROUTE PLANNING PROBLEMS

The common heuristic algorithm depends on its related parameters, and the algorithm parameters need to be adjusted through manual experience. When the scale of the path planning problem is complex and variable, the algorithm results will be greatly affected by the parameters. Therefore, we propose a novel improved SP-MCTS method to solve path planning problems.

The SP-MCTS algorithm is derived from MCTS (Monte Carlo Tree Search) algorithm [18], which is mostly used in single-player game strategy research. It has good strategy selection capabilities, also has advantages in global search problems and solving high-dimensional problems. It is now gradually applied to solve combinatorial optimization problems.

Literature [12] applied SP-MCTS algorithm to solve actual reentrant scheduling problem, pointing out that SP-MCTS has the advantage of not completely relying on parameters, so the solution based on SP-MCTS is still effective when the target problem changes. Literature [13] applied SP-MCTS algorithm to solve robot scheduling problem in hybrid pipelines, and verified the advantages and effectiveness of the algorithm in the optimization problem. In Literature [14], the MCTS algorithm is used to solve geometric matching problem, which proves the advantages of MCTS algorithm in time efficiency and solution accuracy. It can be seen that SP-MCTS is not only suitable for the formulation of single-player game strategies, but also has advantages in solving the NP difficult problem of a tree structure in search space. Since the search space of dynamic and complex path planning problems is also a tree structure, the advantages of using SP-MCTS are particularly obvious.

Therefore, the SP-MCTS algorithm is suitable for solving path planning problems. In the following, we will describe the existing SP-MCTS algorithm mechanism, simulation process, etc. in detail, and propose a path optimization method based on the SP-MCTS algorithm improved by clone selection algorithm.

2) LIMITS OF EXISTING SP-MCTS ALGORITHM

The main idea of SP-MCTS algorithm is to use the results of simulation stage as reference value, evaluate the reference value in selection stage, and use the obtained evaluation value as the decision basis of SP-MCTS algorithm. It can be seen that the results of simulation stage play a crucial role in the final decision, which means improving the quality of the simulation results can improve the solving ability of SP-MCTS algorithm.

There are few existing literatures that apply SP-MCTS algorithm to solve combinatorial optimization problems, and SP-MCTS algorithm in these literatures adopts greedy algorithm in simulation stage. Although the greedy algorithm has advantage of fast solution speed, it only judges based on the current situation and can only make a local optimal choice [19]. Since path planning problems have various types such as local path planning and global path planning [16], the greedy algorithm cannot be suitably applied to all kinds of existing path planning problems. In addition, because the greedy algorithm is easy to fall into local optimization when solving path planning problems, the ability to accurately locate high-potential branches during simulation stage is weak, so it is impossible to provide reliable simulation results for SP-MCTS algorithm, which certainly affects the efficiency of the algorithm.

3) WHY WE CHOOSE CLONAL SELECTION ALGORITHM TO IMPROVE SP-MCTS

It can be seen from Section 2.2.2 that the simulation stage of the existing SP-MCTS algorithm has much room for improvement.

As a mature meta-heuristic algorithm, clonal selection algorithm achieves rapid survival of the fittest through high mutation process [20]. The algorithm first randomly selects N antibodies to form a candidate set, selects the best n antibodies from the candidate set for cloning and mutation, to maintain the diversity of antibodies. Sort the mutated antibodies and original antibodies according to their affinity and select the n best antibodies from them, the add d new antibodies that randomly generated to form a new round of candidate
set to prevent the algorithm from falling into local optimality. After multiple rounds of screening, the optimal candidate set is formed and output the optimal solution.

We analyzed the characteristics of the clone selection algorithm, and combined with the simulation stage of SP-MCTS algorithm to make the following summary.

- Clonal selection algorithm takes into account both global search and local search in implementation process [20], so it is suitable for many types of path planning problems.
- Clonal selection algorithm has better population diversity due to the function of cloning operator. Therefore, when using clonal selection algorithm in simulation stage of SP-MCTS algorithm, it can search as many feasible solutions as possible in a limited time to improve the effectiveness of the simulation results, improve the effectiveness of the simulation results.
- Clonal selection algorithm has low time complexity and fast convergence speed. Therefore, it can quickly provide simulation results in simulation stage, and improve the convergence speed of the entire algorithm.

It can be seen from the above analysis that clone selection algorithm is very suitable for simulation stage of SP-MCTS algorithm. Using the antibody diversity and rapid convergence of clonal selection algorithm can simulate to as many situations as possible for simulation process of SP-MCTS, and provide more valuable simulation results for decision-making of SP-MCTS algorithm, also avoid the problem that greedy algorithm falls into local optimum and results in poor simulation results.

III. SOLVING PATH OPTIMIZATION PROBLEM BASED ON IMPROVED SP-MCTS ALGORITHM

A. RELATED DEFINITIONS

In this paper, we use improved SP-MCTS (Clonal Single Player-Monte Carlo Tree Search, CSP-MCTS) to search for the optimal path. The search process consists of four steps: selection, expansion, simulation, and back-propagation. Think of the path planning problem as a single-player game, specifying the starting point \( s \) as the game start point and the path end point \( e \) as the game end point. Through continuous search to find the optimal next step until the end point, then get the optimal path.

**Definition 1:** Selection refers to determining the optimal next step, and the selected points are combined in order to form the optimal path.

**Definition 2:** Expansion means that after point \( C_i \) is selected, the points in the next set of walkable candidate points \( \{C_1, C_2, \cdots, C_j\} \) are added as child nodes of \( C_i \) in the search tree, and the priority is to add the points closer to \( C_i \).

**Definition 3:** Simulation refers to searching the optimal path from the newly added child node to the end point \( e \) through clonal selection algorithm, and assigning the reward value to the child node of \( C_i \) through the simulation result, so as to select the best next step of \( C_i \).

**Definition 4:** Back-Propagation refers to converting the simulation results into revenue values and updating the information of all selected nodes from the simulation node to the root node.

**Definition 5:** The objective function \( \text{fun}(a, b) \) refers to the goal requirements that the path planning problem needs to achieve, \( a \) is the starting point of the path, and \( b \) is the end point of the path. For example, the objective function of the shortest path problem is the path distance.

**Definition 6:** The simulation result value \( s_{\text{result}} \) refers to the optimal objective function value obtained by the simulation stage algorithm.

**Definition 7:** The reward value \( Q(C_i) \) is the reciprocal of \( s_{\text{result}} \), and its definition is shown in formula (1).

\[
Q(C_i) = \frac{1}{s_{\text{result}}(C_i)} \tag{1}
\]

The point \( C_{i-k} \), which passes before \( C_i \), will update its own reward value through the back-propagation process after the end of a round of simulation. The update method is shown in formula (2).

\[
Q(C_{i-k}) = \frac{1}{\text{fun}(C_{i-k}, C_i) + s_{\text{result}}(C_i)} \tag{2}
\]

B. SOLUTION PROCESS

The process of searching the optimal path by SP-MCTS is actually a tree building process, and the initial state of the search tree \( T \) is shown in Figure 3. The starting point \( C_i \) is the root node, there is only \( C_s \) on initial tree, and then the tree building process is gradually completed by expanding nodes.

![Search tree initialization state: there is only one root node.](image)

The points to be traversed in the path are regarded as nodes, and each node has two attributes: the reward value \( Q(C_i) \) and the number of visits \( t(C_i) \). The algorithm continuously selects the optimal next step through the process of repeated selection, expansion, simulation and back-propagation, and outputs the optimal path until the termination condition is met.

1) SELECTION

When point \( C_i \) is selected, we need to select the optimal next step of \( C_i \) from the next set of candidate \( \{C_1, C_2, \cdots, C_j\} \).

Unlike MCTS algorithm, SP-MCTS uses an improved UCT function [20] in the selection stage to calculate the UCT value of each node. The specific improvement is to add the third term in formula (3) to the traditional UCT function, the purpose is to more fully consider the magnitude of the
change in reward value and ensure the rationality of branch search [13].

\[
\text{UCT} (C_i) = \frac{Q(C_i)}{t(C_i)} \frac{\ln t(C_i)}{t(C_i)} + c \sum Q(C_i)^2 - t(C_i) \cdot \left( \frac{Q(C_i)}{t(C_i)} \right)^2 + D
\]  

(3)

In formula (3), both \(c\) and \(D\) are constants, \(c\) is the exploration constant to make sure that the weights of \(Q(C_i)\) and \(t(C_i)\) are fair, and \(D\) is to ensure that the potential of nodes with few visits is not underestimated. The selection process selects the next node according to the calculated UCT value of each node. Usually the node with the largest UCT value is the high-potential branch, that is, the optimal next step to be selected.

2) EXPANSION

After selecting the optimal next point \(C_j\) of \(C_i\), the algorithm continues to search for the optimal next step of \(C_j\). The point in the next candidate set of \(C_j\) will be added to the search tree \(T\). The process of adding nodes is the expansion of the search tree \(T\). As shown in figure 2, the yellow node in the second step is the newly expanded node.

3) SIMULATION

After expanding \(T\), in order to continue the optimal next step selection, it is necessary to simulate the newly added node to obtain the reward value of the new node. To solve path planning problem more effectively, we use clone selection algorithm to improve the simulation stage of the original SP-MCTS algorithm.

In simulation process, the newly expanded node the starting point \(S_i\) of simulated path, and the optimal path under specific objective function from \(S_i\) to the end point \(C_e\) searched by clone selection algorithm is simulation value. The specific implementation process is shown in Figure 4.

Input is a digital coding path with \(S_i\) as the starting point and \(C_e\) as the ending point. The initial antibody population is the initial random path sequence, and the affinity is the reciprocal of the objective function. The greater the affinity, the better the antibody. Through the cloning and mutation process of clonal selection algorithm, a large number of path combinations can be quickly simulated, and the more simulated the more accurate the simulation results. At the same time, through the continuous selection mechanism, the antibody with the best affinity is used as output to ensure the quality of the simulation result.

4) BACK-PROPAGATION

The main work of back-propagation is to update the attributes of each node after the simulation process. Updating the reward value \(Q(C_i)\) and access count \(t(C_i)\) of all selected nodes from the current node back to the root node, as shown in step 4 of Figure 2.

5) THE IMPROVED SP-MCTS

After back-propagation process ends, if the algorithm searches for the end point, it stops and outputs the optimal path. Otherwise, continue to repeat the above four stages. The pseudo code of the entire algorithm is shown below.

IV. EXPERIMENT AND ANALYSIS

In order to verify the effectiveness and advantages of the method proposed in this paper to solve the path planning problem, we take the common shortest path planning problem.
as an example and introduce the data set proposed by Gerhard et al as the test set [21]. Each of test set contains a set of distance matrices of city sequences. The distance matrix is the commonly used experimental data in path planning problem. The experiment is divided into three parts: Parameter selection experiment, Performance test experiment and Related algorithm comparison experiment.

A. PARAMETERS SELECTION EXPERIMENT

In the selection stage of the algorithm, the constant $c$ and $D$ in formula (3) needs to be determined to ensure the accuracy of the algorithm’s decision-making. Due to the small number of parameters, we conduct Comprehensive experiments to determine the values of the parameters. In this experiment, we use three test sets of path planning problem with different dimensions. The test set dimensions are 51, 101 and 280, and the purpose is to allow the experiment to be carried out under different dimensions of data to improve the accuracy, reliability and persuasiveness of our experiment, and make sure the values we choose can make the algorithm meet the needs of different dimension problems. According to literature [13]–[15], we take each parameter as 3 levels, as shown in Table 2.

\[
\text{TABLE 2. Levels of parameters.}
\]

| Parameter | Level 1 | Level 2 | Level 3 |
|-----------|---------|---------|---------|
| $c$       | 0.01    | 0.1     | 0.3     |
| $D$       | 0.01    | 0.1     | 1       |

According to comprehensive experiment design, there are 16 combinations of parameters, and the average value of three test data sets independently run 30 times under each parameter combination is used to participate in the evaluation. The evaluation standard $E$ calculation is shown in formula (4), the smaller the value of $E$, the better the experiment results.

\[
E = \frac{\sum_{i=1}^{\text{num}} \text{avgi_data}}{\text{num}}
\]  

(4)

where \text{avgi_data} represents the average value of the i-th test data set running 30 times independently, and \text{num} is the number of test data sets participating in the Parameter experiment. The relevant experimental data is shown in Table 3. According to the experimental data in Table 3, evaluate the influence of the parameters on the algorithm at each level, and the average value is the evaluation index of the degree of influence. The relevant data is shown in Table 4, and the influence trend on the algorithm optimization result is shown in Figure 5.

\[
\text{FIGURE 5. Influence of parameter values at different levels on the optimal solution of CSP-MCTS algorithm.}
\]

The ordinate in Figure 5 is the mean distance, which is the objective function of the algorithm. The shorter the mean distance, the higher the efficiency of the algorithm. We can intuitively see from Figure 5 that when $c$ takes level 2 and $D$ takes level 1, the mean distance is the shortest, which means the algorithm has the highest efficiency. Corresponding to the value of $c$ takes 0.1, the value of $D$ takes 0.01.

What’s more, it can be seen from the data in Table 3 and Table 4 that the difference of the algorithm optimization results under different parameters is very small, indicating that the parameters have little effect on the algorithm optimization efficiency. Furthermore, it illustrates the advantage that CSP-MCTS algorithm does not completely depend on parameters.

B. PERFORMANCE TEST EXPERIMENT

In order to verify the effect of the method proposed in this paper on path planning problem, we test the performance of the method. In this experiment, we select 9 sets of experimental data of different dimensions, compare the optimal solution with ideal solution obtained by running the CSP-MCTS algorithm independently for 50 times under each set of data, and calculate variance to verify the accuracy and stability of
TABLE 5. Performance experiment related data.

| Problem dimension | Ideal solution | CSP-MCTS error | mean | variance |
|-------------------|----------------|----------------|------|----------|
| 51                | 426            | 0              | 426  | 0        |
| 76                | 538            | 0              | 538  | 0        |
| 101               | 629            | 0              | 629  | 0        |
| 130               | 6110           | 0              | 6110 | 0        |
| 159               | 42080          | 0              | 42080| 0        |
| 200               | 29368          | 0              | 29368| 0        |
| 225               | 3916           | 3              | 3920 | 2        |
| 264               | 49135          | 0              | 49135| 0        |
| 280               | 2579           | 0              | 2581 | 2        |

CSP-MCTS algorithm. The experimental results are shown in Table 5.

“Error” in Table 5 represents the gap between the optimal solution and the ideal solution. From this column, it can be seen that CSP-MCTS algorithm can effectively solve the path planning problem in different dimensions. This is because the simulation results provided by clone selection algorithm make CSP-MCTS accurately locate the high-potential branch, and make the correct choice at every step. It can be seen from the column of “variance” that CSP-MCTS algorithm has high stability.

In addition, time complexity is one of the important performances of an algorithm. The runtime of SP-MCTS algorithm can be simply be computed as $O(mkI/C)$ [22], where $m$ is number of children of a node, $k$ is number of simulations of a child, $I$ is the number of iterations, and $C$ is the number of cores available.

To test the time performance of CSP-MCTS, we compare the time between CSP-MCTS algorithm and PSO + ACO + kOPT [23] algorithm with less time complexity. The relevant experiment results are shown in Figure 6, the ordinate represents the average running time of 50 times, the unit is second.

![Figure 6. Comparison of time consumption of PSO + ACO + kOPT algorithm and CSP-MCTS algorithm in finding optimal solutions on different dimensional problems.](image)

As can be seen from the Figure 6, in small problems, CSP-MCTS algorithm has no obvious advantage in time, but as the scale of the problem increases, CSP-MCTS algorithm has more advantages in terms of time performance, and can solve complex path planning problem quickly and stably.

C. COMPARATIVE EXPERIMENT

The comparison experiment is divided into two parts. The first part is to compare CSP-MCTS algorithm with ant colony algorithm (ACO), MSSICA algorithm [24], and the original SP-MCTS algorithm [12], [15], [16] to verify CSP-MCTS Solution effect. ACO is a classic algorithm for solving path planning problems. MSSICA is the shortest path planning problem solving algorithm that has been newly proposed in recent years and has a good effect. Therefore, the comparison algorithm is representative and the comparison results are more convincing.

The second part is to compare CSP-MCTS algorithm with ACO algorithm, and verify the advantages of CSP-MCTS algorithm without relying on parameters through different parameter combinations in the two algorithms.

1) ALGORITHM EFFECT COMPARISON EXPERIMENT

In this experiment, we choose 6 different dimension data sets. The parameter values of CSP-MCTS algorithm and SP-MCTS algorithm are set to the same, $c$ is 0.1 and $D$ is 0.01. ACO algorithm and MSSICA algorithm are intelligent bionic algorithms, their structure are different with SP-MCTS algorithm, so it’s not proper to set the relevant parameters of all comparison algorithms to the same value. In order to ensure the validity and fairness of the experiment, we obtained the related parameters and termination conditions that can achieve the best results of the other two comparison algorithms through references, then comparing with their best effects.

Each set was independently executed 30 times by each algorithm. The average solution error rate (MER) and best solution error rate (BER) of the algorithm are calculated according to the relevant experimental results, use these two indicators as reference to compare the solving ability of the algorithm. The calculation methods of these two indicators are shown in formulas (5) and (6). $\text{avg} \_ \text{result}$ is the average of the results of 30 experiments; $\text{best} \_ \text{result}$ is the optimal solution obtained from 30 experiments, and $\text{opt}$ is the ideal solution. Related data is shown in Table 6.

$$\text{MER} = \frac{\text{avg} \_ \text{result} - \text{opt}}{\text{opt}} \quad (5)$$

$$\text{BER} = \frac{\text{best} \_ \text{result} - \text{opt}}{\text{opt}} \quad (6)$$

It can be seen from the data in Table 6:

- Whether it is MER or BER value, CSP-MCTS algorithm is better than the comparison algorithms, which proves that CSP-MCTS algorithm is not only efficient but also stable.
- CSP-MCTS algorithm is more effective in solving path planning problem than the original SP-MCTS algorithm. It proves that after improving by clone...
Y.-W. Xia et al.: Path Planning Method Based on Improved SP-MCTS

| Problem dimension | Error rate(%) | MSSICA | ACO | SP-MCTS | CSP-MCTS |
|-------------------|--------------|--------|-----|---------|----------|
| 51                | MER          | 1.83   | 2.78| 0       | 0        |
|                   | BER          | 1.08   | 0   | 0       | 0        |
| 76                | MER          | 2.76   | 3.56| 0       | 0        |
|                   | BER          | 1.77   | 1.57| 0       | 0        |
| 100               | MER          | 0.15   | 5.88| 0.08    | 0        |
|                   | BER          | 0.02   | 2.11| 0       | 0        |
| 150               | MER          | 0.83   | 6.76| 0.17    | 0        |
|                   | BER          | 0.32   | 3.08| 0.02    | 0        |
| 200               | MER          | 0.80   | 8.01| 0.21    | 0.03     |
|                   | BER          | 0.51   | 4.42| 0.06    | 0        |
| 380               | MER          | 2.24   | 10.12| 1.91    | 0.88     |
|                   | BER          | 1.41   | 5.56| 1.04    | 0.68     |

2) PARAMETER-DEPENDENT EXPERIMENT

In this experiment, we compare CSP-MCTS algorithm with ACO algorithm. By changing the parameters $c$ and $D$ in CSP-MCTS algorithm and the maximum number of ant colony iterations maxIter in ACO algorithm, to observe the degree of influence of the parameters on the algorithm. Taking the data with the dimension of 226 as an example, 10 experiments are performed on the same data set under each parameter combination, and the average value is the result for comparative experiments. The values of related parameters are shown in Table 7, and the experimental results are shown in Figure 7a.

![Figure 7a](image)

**FIGURE 7**. The effect of parameters on algorithm optimal solutions in two different situations.

It can be seen from Figure 7a that under different parameter conditions, the optimization efficiency of ACO algorithm largely depends on the parameter maxIter, while the parameters in CSP-MCTS algorithm have little effect on the search result. It is proved that CSP-MCTS algorithm has the advantage of not completely relying on parameters compared to the heuristic algorithm like ACO.

Also it can be seen from Figure 7b that under the condition of the same parameter and different problem dimensions, the gap between the optimal solution and the ideal solution of ACO algorithm increases with the increase of the problem dimension, which means the algorithm cannot maintain the solution efficiency. But CSP-MCTS algorithm has always maintained a high efficiency optimization ability. It can be seen that CSP-MCTS algorithm can solve more variable path planning problems more effectively than the traditional heuristic algorithm.

D. ANALYSIS

SP-MCTS is a type of decision-making algorithm based on simulation, and the accuracy of decision results mostly depends on the simulation results. The core parameters $c$ and $D$ in SP-MCTS are only to control the weights of node’s simulation results and the number of visits, in to ensure that the branch potential with fewer visits will not be underestimated, so the efficiency of the algorithm is very little affected by the parameters, and we proved this view through parameters selection experiment and parameter-dependent experiment. In addition, we also proved that CSP-MCTS algorithm has better stability and better solution performance through performance experiments of the algorithm. This is because the algorithm makes decisions based on simulation results, which makes the decision results more reliable and effective. Then, we proved that CSP-MCTS algorithm is superior to the SP-MCTS algorithm, the newer heuristic algorithm, and the traditional heuristic algorithm in terms of time consumption and solution effect through comparative experiments, indicating that clone selection algorithm can quickly provide reliable simulation results, improve the accuracy of SP-MCTS algorithm to locate high-potential branches, and enable the algorithm to quickly and accurately locate the optimal branch based on the simulation results.

V. CONCLUSION

This paper proposes a solution to path planning problem based on the improved SP-MCTS algorithm. We apply...
SP-MCTS algorithm to path planning problem, and use clonal selection algorithm to improve simulation stage of SP-MCTS algorithm, aiming to improve the accuracy of the algorithm simulation and decision-making results. We have proved through experiments that the improved SP-MCTS algorithm has better solving ability on path planning problems and has the advantage of not completely relying on parameters. Compared with existing methods, it can solve the changing path planning problem more stably, faster, more accurately and more effectively.

Also, CSP-MCTS algorithm has certain limitations. The accuracy of the algorithm to find an optimal solution is in a positive relationship with time consumption. The more simulations, the more accurate decisions the algorithm will make, but the longer time the algorithm will consume. Therefore, it is very important to choose an appropriate algorithm in the simulation stage to ensure that as many results as possible can be simulated quickly in the shortest possible time. In the future, I will do further research on this limitation and apply CSP-MCTS algorithm to more complex combinatorial optimization problems.

REFERENCES

[1] P. Yan, L. Huali, X. Lining, and R. Teng, “Review of vehicle path optimization problems and solution methods,” Control Theory Appl., vol. 36, no. 10, pp. 1573–1584, Sep. 2019.

[2] Q. Lin, S. Liu, Q. Zhu, C. Tang, R. Song, J. Chen, C. A. C. Coello, K.-C. Wong, and J. Zhang, “Particle swarm optimization with a balanceable fitness estimation for many-objective optimization problems,” IEEE Trans. Evol. Comput., vol. 22, no. 1, pp. 32–46, Feb. 2018.

[3] M. Rojas Santiago, S. Muthuswamy, and M. Hulett, “An ACO algorithm for scheduling a flow shop with setup times,” Int. J. Ind. Syst. Eng., vol. 1, no. 1, p. 1, 2020.

[4] L. Jun and S. Zhebing, “Research on path planning based on improved D * lite genetic algorithm,” Mach. Tool Hydraul., vol. 47, no. 11, pp. 39–42, Nov. 2019.

[5] X. Yan, P. Li, K. Tang, L. Gao, and L. Wang, “Clonal selection based intelligent parameter inversion algorithm for prestack seismic data,” Inf. Sci., vol. 517, pp. 86–99, May 2020.

[6] G. Li and W. Chou, “Path planning for mobile robot using self-adaptive learning particle swarm optimization,” Sci. China Inf. Sci., vol. 61, no. 5, May 2018, Art. no. 052204.

[7] H. Q. Wang, Y. Y. Hu, W. D. Liao, T. B. Yan, and D. Y. Wang, “Path planning algorithm based on improved artificial bee colony algorithm,” Control Eng. China, vol. 23, no. 9, pp. 1407–1411, 2016.

[8] Y.-B. Cheng, T. Huang, H. T. Huang, Y.-J. Gong, and J. Zhang, “Multi-population ant colony system for multiple path planning of food delivery applications,” in Proc. IEEE Symp. Ser. Comput. Intell. (SSCI), Nov. 2018, pp. 68–73.

[9] M. R. Panda, P. K. Das, S. Dutta, and S. K. Pradhan, “Optimal path planning for mobile robots using oppositional invasive weed optimization,” Comput. Intell., vol. 34, no. 4, pp. 1072–1100, Nov. 2018.

[10] J. Xin, J. Zhong, J. Sheng, P. Li, and Y. Cui, “Improved genetic algorithms based on data-driven operators for path planning of unmanned surface vehicle,” Int. J. Robot. Autom., vol. 34, no. 6, 2019.

[11] E. Rapanaki, I.-D. Psychas, M. Marinaki, Y. Marinakis, and A. Migdalas, “A clonal selection algorithm for multiobjective energy reduction multi-depot vehicle routing problem,” in Proc. LOD, 2018, pp. 381–393.

[12] S. Matsumoto, N. Hirose, K. Itanaga, N. Ueno, and H. Ishii, “Monte-Carlo tree search for a reentrant scheduling problem,” in Proc. 40th Int. Conf. Comput. Indutrial Eng., Jul. 2010, pp. 1–6.

[13] G. Jian, Y.-Y. Shi, P. Z. Wei, J.-F. Zhang, “Based on SP-MCTS algorithm hybrid flow shop scheduling robot,” J. Comput. Integr. Manuf. Syst., vol. 25, no. 9, pp. 2208–2218, 2019.

[14] M. A. Pinheiro, J. Kybic, and P. Fua, “Geometric graph matching using Monte Carlo tree search,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 11, pp. 2171–2185, Nov. 2017.

[15] M. H. S. Segler, M. Preuss, and M. P. Waller, “Planning chemical syntheses with deep neural networks and symbolic AI,” Nature, vol. 555, no. 7698, pp. 604–610, Mar. 2018.

[16] Z. Guanglin, H. Xiaomei, C. Jianfei, Z. Lei, and Y. Tao, “Overview of path planning algorithms and their applications,” Modern Machinery, vol. 37, no. 5, pp. 85–90, 2011.

[17] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavenner, D. Perez, and S. Samothrakis, and S. Colton, “A survey of Monte Carlo tree search methods,” IEEE Trans. Comput. Intell. AI Games, vol. 4, no. 1, pp. 1–43, Mar. 2012.

[18] M. P. Schadd, M. H. Winands, H. J. Van Den Herik, G. M. B. Chaslot, and J. W. Uiterwijk, “Single-player Monte-Carlo tree search,” Proc. Comput. Games, vol. 34, no. 5, pp. 3–11, 2008.

[19] L. Dongdong, “Research on the optimization of picking path of dual-zone warehouse,” M.S. thesis, Dept. Syst. Theory. China, Qingdao Univ., Qingdao, China, 2015.

[20] L. N. de Castro and F. J. Von Zuben, “Learning and optimization using the clonal selection principle,” IEEE Trans. Evol. Comput., vol. 6, no. 3, pp. 239–251, Jun. 2002.

[21] (Jul. 2020). Gerhard Reineit Discrete and Combinatorial Optimization. [Online]. Available: http://comopt.ufi.unicarlstadt.de/software/TSPLIB95/top/

[22] Y. Jin and S. Benjamin. CME 323, Report Stanford, America. Accessed: Feb. 27, 2016. [Online]. Available: https://stanford.edu/~rezab/classes/cme323/s15/projects/montecarlo_search_tree_report.pdf

[23] I. Khan, M. K. Maiti, and M. Maiti, “Coordinating particle swarm optimization, ant colony optimization and K-opt algorithm for traveling salesman problem,” in Proc. Int. Conf. Math. Comput., Singapore: Springer, 2017, pp. 103–119.

[24] C. Menghui, L. Junlin, X. Jianfeng, and L. Xiangjun, “Diversified search empire competition algorithm for traveling salesman problem,” Comput. Appl., vol. 39, no. 10, pp. 2992–2996, 2019.

YU-WEI XIA is currently pursuing the M.S. degree with Hubei University. Her research interests include heuristic algorithm, software engineering, and artificial immunology.

CHAO YANG received the Ph.D. degree in information security from Wuhan University, Hubei, China, in 2012. He is currently an Associate Professor with the School of Computer Science and Information Engineering, Hubei University, Hubei, China. His research interests include information security and artificial intelligence.

BING-QIU CHEN received the B.S. degree in information and computing science from Huanggang Normal University, in 2017. He is currently pursuing the M.E. degree in computer technology with Hubei University, Hubei, China. His research interests include artificial immune systems and heuristic algorithm.

* * *