

Research Article

Intelligent Critical Path Computation Algorithm Utilising Ant Colony Optimisation for Complex Project Scheduling

Xiaokang Han ^{1,2}, Wenzhou Yan,¹ and Mei Lu¹

¹School of Management, Xi'an University of Architecture and Technology, Xi'an 710055, China

²Project Control Department, Hualu Engineering & Technology Co., Ltd., Xi'an 710065, China

Correspondence should be addressed to Xiaokang Han; xk65@xauat.edu.cn

Received 24 March 2021; Revised 6 June 2021; Accepted 23 June 2021; Published 2 July 2021

Academic Editor: Zhile Yang

Copyright © 2021 Xiaokang Han et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In large and complex project schedule networks, existing algorithms to determine the critical path are considerably slow. Therefore, an algorithm with a faster convergence is needed to improve the efficiency of the critical path computation. The ant colony algorithm was first applied to the travelling salesman problem to determine the shortest path. However, many problems require the longest path in practice; the critical path in the scheduling problem is the longest path in the scheduling network. In this study, an improved ant colony algorithm to determine the critical path by setting the path distance and time as negative, while the transition probability remains unchanged, is proposed. The case of a coal power plant engineering, procurement, and construction (EPC) project was considered. The results show that a peak number of optimal solutions appeared at approximately the 9th iteration; however, instabilities and continued fluctuations were observed even afterward, indicating that the algorithm has a certain randomness. Convergence is apparent at the 29th iteration; after the 34th iteration, a singular optimal solution, the longest or critical path, is obtained, indicating that the convergence rate can be controlled and that the critical path can be obtained by setting appropriate parameters in the solution method. This has been found to improve the efficiency of calculating the critical path. Case validation and algorithm performance testing confirmed that the improved ant colony algorithm can determine the critical path problem and make it computationally intelligent.

1. Introduction

With the growing number of large, complex, and systematic projects, there has been an increase in the number of units, work packages, and activities in project scheduling. When over a thousand activities are involved, the computational speed of existing algorithms can no longer meet the demand of scheduling management; therefore, an intelligent algorithm should be used to improve computing efficiency. The traditional Dijkstra method is a network optimisation analysis of operational research that can determine the longest and shortest paths in a network, and the Dijkstra's algorithm searches to find the globally shortest path between two nodes of a graph [1]. It is recognised as the best method for finding the shortest path of a network [2, 3], but it has too many iterations and thus performs at a low computational speed for large and

complex project networks. Primavera 6, project, and other scheduling software offer methods of critical path computation centred on the analysis of conventional operation research optimisation methods.

The project scheduling problem was first proposed in the 1960s, and the research of Conway et al. [4] on the scheduling problem is usually regarded as the formal beginning of scheduling theory research. Subsequently, Pritsker et al. [5] proposed a resource-constrained project scheduling problem in 1969.

The critical path methodology (CPM) is a tool for planning, scheduling, and coordinating complex engineering projects [6]. It is used to determine the shortest construction period of the project by estimating the floating time of each logical path in its scheduling network. The critical path determines the sequence of activities in the network; its floating time is zero. The activities on the critical

path are the most important ones in the project; this path is the longest of the entire project scheduling network.

Determining the critical path involves finding the longest path in the entire project scheduling network. It can be estimated with the program evaluation and review technique (PERT), the double-code network diagram, or the single-code network diagram. However, in a large and complex project scheduling network, conventional computation methods converge considerably slowly. Therefore, a method of faster convergence is needed to improve the computational efficiency of the critical path.

Xu et al. [7] proposed an algorithm to generate all critical paths in the order of increasing path length, which simplifies its data structure. Wang [8] proposed a new algorithm for solving critical paths without topological sorting, which led to graph storage and the algorithm-solving process occupying less space and reduced its time complexity. Li et al. [9] proposed the free time-difference theorem and k -order key route algorithm, which realise global optimisation through local optimisation and simplify the calculation workload. Wang et al. [10] proposed a subcritical route algorithm on the CPM network based on the characteristics of node time difference; a layer-upon-layer simplification of the CPM network was performed, after which the subcritical route was rapidly solved. Zhang et al. [11] applied object-oriented technology to realise active network diagrams and critical path algorithms, which reduced the complexity of storage and computation. Han et al. [12] proposed a critical path algorithm involving the optimisation of matrix multiplication to improve its real-time performance. Han et al. [13] proposed a reasoning model for emergency measures that can be applied in the scheduling control of industrial projects, which is an excellent way to provide effective case support and decision data for the improvement of early warning and feedback tracking theory in project scheduling control. Han [14] proposed a WBS-free scheduling method based on the database relational model, which solves the problem of diversity in the scheduling form and implements the innovation of the scheduling method. Guo et al. [15] proposed a coastal ship path planning model based on the optimized deep Q network (DQN) algorithm in order to better realize the ship path planning in the process of navigation. Deng et al. [16] proposed an improved differential evolution (HMCQDE) combining the quantum computing characteristics of the quantum evolutionary algorithm (QEA) and the divide-and-conquer idea of the cooperative coevolution evolutionary algorithm (CCEA) in order to overcome the low solution efficiency, insufficient diversity in the later search stage, slow convergence speed, and a high search stagnation possibility of the differential evolution (DE) algorithm. Deng et al. [17] proposed an enhanced MSIQDE (improved QDE with multistrategies) algorithm based on mixing multiple strategies. Deng et al. [18] designed an improved quantum evolutionary algorithm (QEA) based on the niche coevolution strategy and enhanced particle swarm optimization (PSO) and proposed an IPOQEA-based gate allocation method to allocate the flights to suitable gates within different periods to solve this bottleneck of gate resource.

Since the beginning of the century, most explorations of critical path algorithms have been carried out by optimisation methods such as sorting, time difference, and image; however, they do not consider the computational speed of scheduling networks with thousands of activities. This study was the first study to use the ant colony algorithm for critical path determination.

The ant colony algorithm is a swarm intelligence algorithm, which can make the computation of the critical path converge faster and obtain the critical path intelligently. In this study, the ant colony algorithm was utilised to develop a novel, efficient, and convenient method to compute the critical path and to optimise project scheduling. The study focuses on the intelligent critical path computation algorithm of the project scheduling network.

2. Project Scheduling Network and Critical Path

The project scheduling network is a directed acyclic graph with start and endpoints, in which the critical path is simply the longest one. There are two representations of activities in the network diagram: activity on node (AoN), which is a single-code network diagram, where nodes represent the activities as shown in Figure 1 and activity on arc (AoA), which is a double-code network diagram, where arcs (arrow lines) represent the activities as shown in Figure 2.

The digraph is marked as $D = (V, A)$, where D denotes the project scheduling network as a function of V and A , while V and A represent the node (i.e., the set of activities in the network) and arc (i.e., the logical relations connecting the activities) of D , respectively. Additionally, $(i, j) \in A$ denotes all activities i and j related, such that activity i precedes activity j .

3. Ant Colony Algorithm

The ant colony algorithm simulates and uses the characteristics of the foraging behaviour of ants to determine the shortest path. For example, ants can detect pheromones in their neighbourhood and can accordingly release them in response; however, the pheromone volatilises over time. In the double-bridge experiment, artificial ants were designed to simulate movement in a double-bridge system and thus determine the shortest path. Artificial ants have effective mechanisms for the renewal and volatilisation of pheromones. Moreover, the path-finding ability of ants can be used to avoid repeated searches and premature convergence. Artificial ants, similar to real ones, can communicate through pheromones and cooperate with each other to complete tasks through local behaviours based on probability decision-making, and all of them are self-organised [19].

The ant colony algorithm had its earliest application in the travelling salesman problem (TSP). A TSP is an optimisation problem involving n cities and aims to determine the shortest path through all of them while passing through each of them only once. The parameters of the algorithm are as follows: Q is the number of ants in the colony, $B_i(T)$ is the number of ants in city i at time T , D_{ij} is the distance between

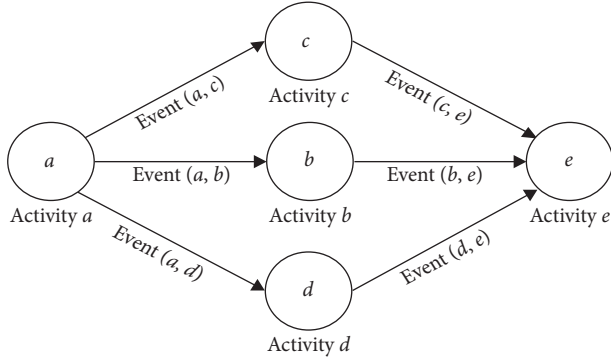


FIGURE 1: Single-code network diagram representation of project planning network activities (AoN).

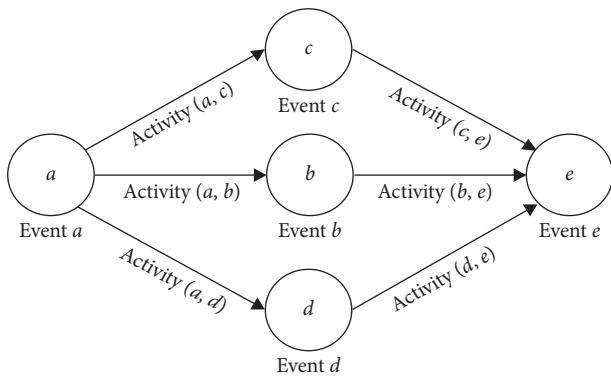


FIGURE 2: Double-code network diagram representation of project planning network activities (AoA).

cities i and j , η_{ij} is the visibility of edge (i, j) and remains unchanged, τ_{ij} is the pheromone track intensity on edge (i, j) , $\Delta\tau_{ij}$ is the amount of pheromone per unit length left by ants on edge (i, j) , $P_{ij}^s(t)$ is the transfer probability of ants s from cities i to j , i is the current city of the ants, and j is the city that is yet to be visited. Therefore, at time t , the transfer probability of ants in cities i and j can be calculated as follows:

$$P_{ij}^s(t) = \begin{cases} \frac{\tau_{ij}^a(t)\eta_{ij}^b(t)}{\sum_{e \in \text{allowed}_s} \tau_{ie}^a(t)\eta_{ie}^b(t)}, & j \in \text{allowed}_s, \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

where $\text{allowed}_s = \{0, 1, 2, \dots, n-1\}$ indicates that the cities allowed to be selected by the ants in the next step will change with the process of ants s , while a and b reflect the relative importance of the accumulated and heuristic information in the process of ant path selection, respectively.

The advantages of this algorithm are as follows:

- (i) As a population-based evolutionary algorithm, it is easier to realise parallel computing
- (ii) The algorithm has high reliability and stability, is not easily disturbed, and has a wide range of applications

- (iii) The algorithm has strong adaptability and can be combined with other heuristic algorithms to further improve its performance

In other words, the ability of the ant colony algorithm to obtain the best solution is satisfactory and suitable for solving longest path problems, such as determining the critical path.

4. Intelligent Critical Path Algorithm Based on the Improved Ant Colony Algorithm

An improved ant colony algorithm is proposed to solve the longest path problem by setting the path distance and time to be negative and keeping the transition probability unchanged.

4.1. Estimating the Critical Path. The critical path problem is a typical combination optimisation problem. Using the dual-code network chart to show the project network, we define the collection of n as a set of events; this makes the critical path problem equivalent to determining the longest path to the event.

Let event 1 and event n , both of which are unique, be the initial and end nodes of the project scheduling network. Let D_{ij} be the distance between events i and j , and the critical path problem of the network can be represented by the digraph $D = (V, A)$, where V is the set of events and A is the set of distances between events.

To determine the longest path, the distance between events i and j is defined as $-d_{ij}$. If each ant has the same speed, the ant will first arrive at the j event where $-d_{ij}$ is smaller, which means that the j event at d_{ij} is larger. Thus, the problem of finding the longest path is effectively solved.

$b_i(t)$ ($i = 1, 2, 3, \dots, n$) is the number of ants in event i at time t , and $m = \sum_{i=1}^n b_i(t)$ is the total number of ants in all n events.

Each ant has the following characteristics:

- (i) Each ant moves between events at the same speed, denoted by v
- (ii) The time taken by each ant to move from event i to j is negative, expressed as $(-d_{ij}/v)$
- (iii) The probability of an ant moving from event i to j is determined by, and is directly related to, the pheromone concentration between the two events
- (iv) Each ant starts from event i (start node), makes its way to event n (end node), and then returns the same way to event i ; the smaller the time t (negative value) of the return to event i , the earlier does the ant arrive.
- (v) Each ant moving from event i to j leaves an equivalent amount of pheromone in the unit length of the activity (i, j)

Let $\tau_{ij}(t)$ be the pheromone intensity of activity (i, j) at time t . The ant selects the next event j from event i at time t and arrives at event j at time $t + (-d_{ij}/v)$. Therefore, if m ants transfer m times in $(t, t + (-d_{ij}/v))$ as an iteration of the ant

colony algorithm, then each iteration of the algorithm takes $2n$ movements for each ant to complete a round trip.

$$\tau_{ij} \left(t + \frac{-d_{ij}}{v} \right) = \rho \tau_{ij}(t) + \Delta \tau_{ij}. \quad (2)$$

Here, ρ , where $0 < \rho < 1$, is the degree of pheromone volatilisation. The longer the time, the greater is the degree of pheromone volatilisation.

$$\Delta \tau_{ij} = \sum_{k=1}^m \Delta \tau_{ij}^k. \quad (3)$$

Here, $\Delta \tau_{ij}^k$ is the pheromone quantity per unit length of the ant on activity (i, j) between time t and time $t + (-d_{ij}/v)$. It can be calculated as

$$\Delta \tau_{ij}^k = \begin{cases} \frac{Q}{L_k}, & \text{If the } k^{\text{th}} \text{ ant goes through activity } (i, j), \\ 0, & \text{If the } k^{\text{th}} \text{ ant doesn't go through activity } (i, j), \end{cases} \quad (4)$$

where Q is a constant, and L_k is the length of the path taken by the k^{th} ant. Let $\tau_{ij}(0) = c$. This denotes that the pheromone intensity of activity (i, j) at the beginning of the experiment is some constant c .

To satisfy the constraint that each ant passes through all n events, each ant is associated with data from a tabu table. The tabu table stores data on all the events that have been passed by the ants till time t and prevents them from passing any of those same events again until after $2n$ iterations are complete. This can be used to calculate the ant's current solution after completing a round trip. $\text{Tabu}_k(s)$ is defined as a dynamic growth vector including the tabu table of the k^{th} ant; tabu_k is a set of elements from this vector, and $\text{tabu}_k(s)$ denotes the s^{th} element in the table (i.e., the current event s occupied by the k^{th} ant).

The visibility size and pheromone concentration around the ant determine its next event. Therefore, the transfer probability of ant k from event i to j at time t is calculated as follows:

$$p_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha(t) \eta_{ij}^\beta(t)}{\sum_{k \in \text{allowed}_k} \tau_{ik}^\alpha(t) \beta_{ik}^\beta(t)}, & j \in \text{allowed}_k, \\ 0, & \text{otherwise,} \end{cases} \quad (5)$$

where $\text{allowed}_k = \{0, 1, 2, \dots, n\}$ indicates that the event is amongst those that are next selected by ant k ; α and η reflect the relative importance of the accumulated and heuristic information, respectively, in the process of the ant's path selection; and η_{ij} denotes the visibility. Furthermore,

$$\eta_{ij} = \frac{1}{d_{ij}}. \quad (6)$$

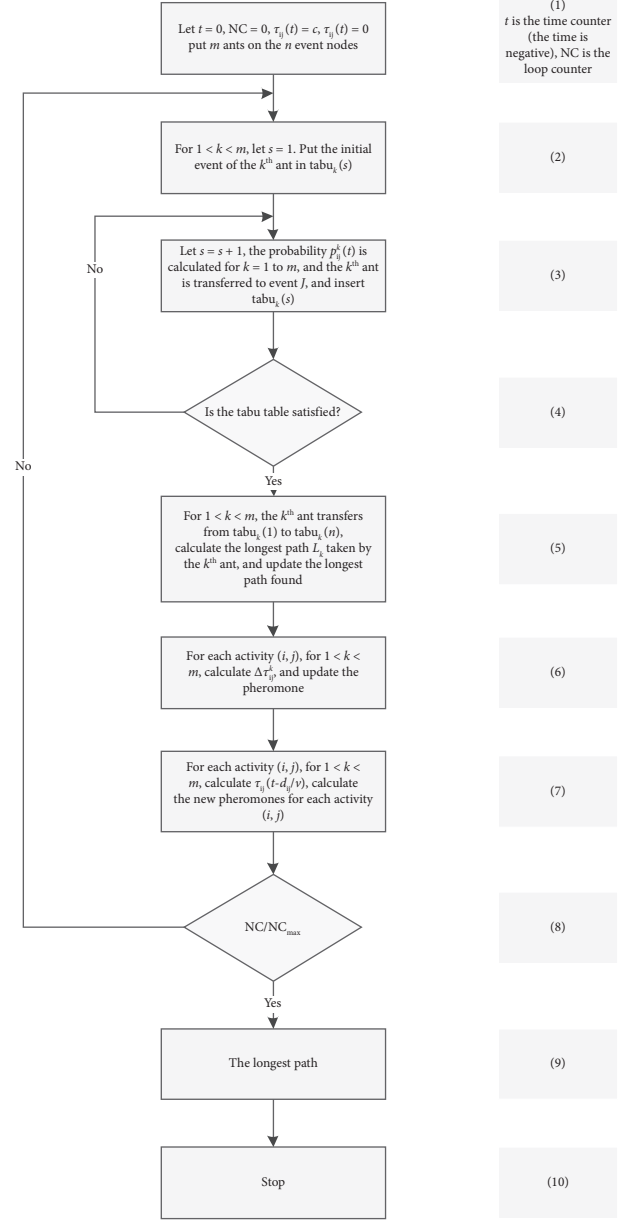


FIGURE 3: Improved ant colony algorithm flow chart.

4.2. *Determining the Critical Path Based on the Improved Ant Colony Algorithm.* As shown in Figure 3, the steps to solve the critical path problem using the improved ant colony algorithm 1 are as follows:

5. Results

The case of a coal power plant engineering, procurement, and construction (EPC) project was considered. Our proposed algorithm was applied to this case, and its performance was evaluated and discussed. The scheduling network for this project includes engineering, procurement, and

- (1) Initialization process: let $t := 0$; /* t is the time counter */
 NC: = 0; /* NC is the loop counter */
 $\tau_{ij}(0) := C$; /* let the initial value of pheromone intensity for each path (i, j) be C */
 $\Delta\tau_{ij} = 0$; /* let the initial value of incremental pheromone intensity be 0 */
 η_{ij} ; /* $\eta_{ij} = 1/d_{ij}$ in critical path*/
 $\text{tabu}_k = \emptyset$; /* the tabu_k table is empty in the initial phase */
 The m ants were randomly placed on n events
- (2) Let $s := 1$ /* s is the tabu_k table index, placing the initial event of each ant in the current tabu_k table */
 For $i := 1$ to n do;
 For $k := 1$ to m do;
 $\text{tabu}_k(s) = i$; /* placing the initial event of the k^{th} ant in the current tabu_k table */
- (3) Repeating the following steps until the tabu_k table is full. /* This step is going to be repeated $(n - 1)$ times */
 Let $s := s + 1$;
 For $i := 1$ to n do;
 For $k := 1$ to m do;
 The next event is selected with probability $p_{ij}^k(t)$, whose probability is specifically given by equation (5);
 At time t , the k^{th} ant transfer to event j at event $i = \text{tabu}_k(s - 1)$;
 Add the event j to $\text{tabu}_k(s)$.
- (4) Whether the tabu_k table is full?
- (5) For $k := 1$ to m do
 The k^{th} ant transfer from $\text{tabu}_k(n)$ to $\text{tabu}_k(1)$, and the ant returns to the starting event after one cycle;
 Calculate the distance travelled by the k^{th} ant;
 Update the longest travel path found
- (6) For each path (i, j) ;
 For $k := 1$ to m do;
 $\Delta\tau_{ij}^k$ is calculated according to equation (3).
- (7) For each path (i, j)
 The pheromone intensity $\tau_{ij}(t + (-d_{ij}/v))$ is calculated according to equation (2);
 Let $t := t + (-d_{ij}/v)$;
 NC: = NC + 1;
 Let $\Delta\tau_{ij} = 0$ for each path (i, j) .
- (8) If $\text{NC} < \text{NC}_{\max}$
 Clear all tabu_k table;
 Return to step (2);
- (9) Else
 Output the longest path, the critical path.

ALGORITHM 1: The intelligent critical path computation algorithm steps utilising ant colony optimisation.

TABLE 1: Activities involved in a project scheduling network for a coal plant.

Activity	Activity name	Duration (months)
(a, b)	Piling plan and detail drawing	1
(b, c)	Piling construction (including pile testing)	1
(b, e)	Foundation plan (and beam) layout drawing	1
(c, d)	Construction of reinforced concrete foundation	1
(e, f)	Steel structure detail drawing	2
(f, g)	Procurement of steel structure	2
(g, j)	Erection of steel structure	1
(a, h)	Procuring specifications of the static equipment	1
(h, i)	Procurement of static equipment	5
(j, k)	Installation of static equipment	2

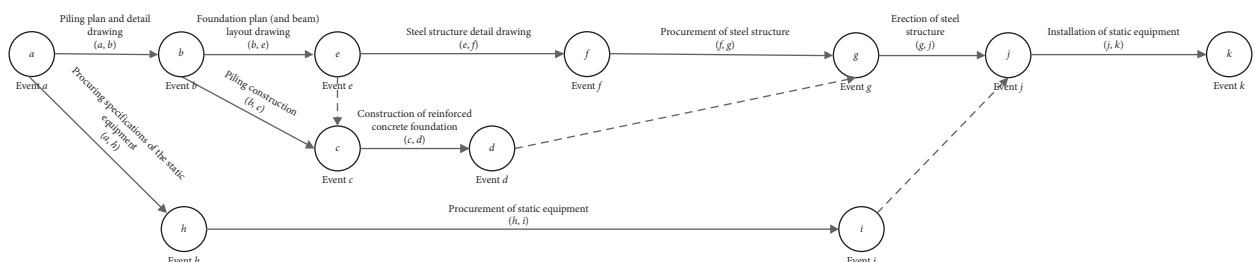


FIGURE 4: Double-code network diagram of the scheduling network for a coal plant.

TABLE 2: The coordinates of events involved in a project scheduling network for a coal plant.

Events	Coordinates
<i>a</i>	(0, -1)
<i>b</i>	(-1, -1)
<i>c</i>	(-1.7071, -1.7071)
<i>d</i>	(-2.7071, -1.7071)
<i>h</i>	(-0, -2)
<i>i</i>	(-5, -2)
<i>e</i>	(-2, -1)
<i>f</i>	(-4, -1)
<i>g</i>	(-6, -1)
<i>j</i>	(-7, -1)
<i>k</i>	(-9, -1)

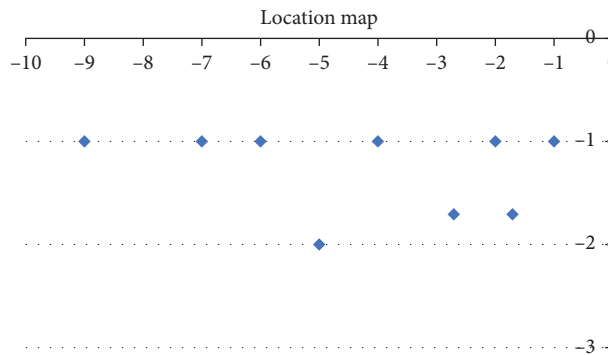


FIGURE 5: The location map of events involved in a project scheduling network for a coal plant.

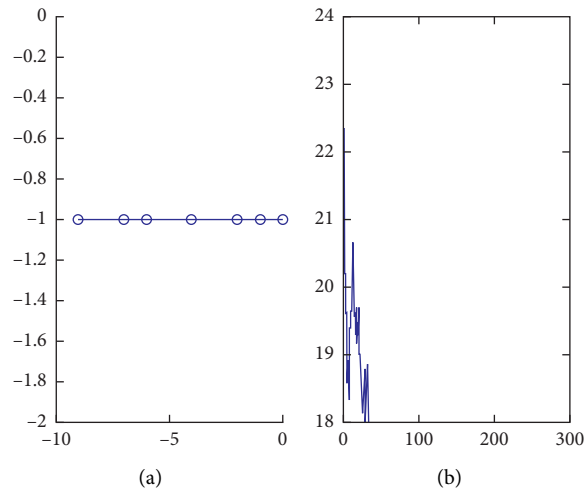


FIGURE 6: Iteration times of the optimal solution (critical path).

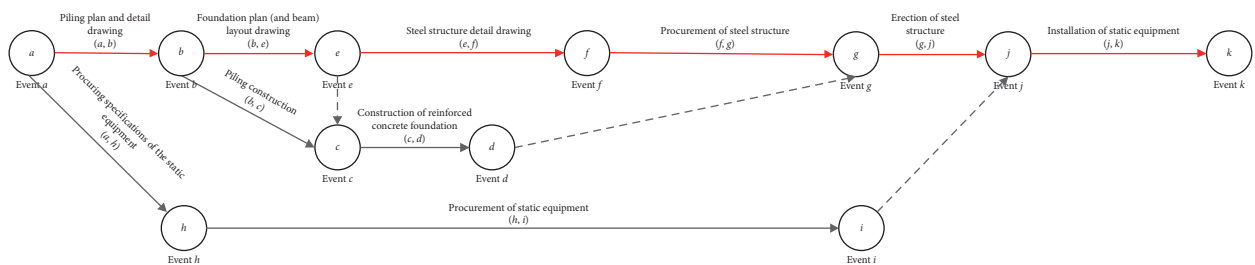


FIGURE 7: Critical path diagram of a coal plant scheduling network.

TABLE 3: Activities involved in the critical path of a coal plant scheduling network.

Activity	Activity name	Critical path (month)
(a, b)	Piling plan and detail drawing	1
(b, c)	Piling construction (including pile testing)	
(c, e)	Foundation plan (and beam) layout drawing	1
(c, d)	Construction of reinforced concrete foundation	
(e, f)	Steel structure detail drawing	2
(f, g)	Procurement of steel structure	2
(g, j)	Erection of steel structure	1
(a, h)	Procuring specifications of the static equipment	
(h, i)	Procurement of static equipment	
(j, k)	Installation of static equipment	2
Total		9

construction activities with a contract period of nine months. As given in Table 1 and Figure 4, a double-code network diagram was used to describe the scheduling network.

The following will use MATLAB to program the performance test of the case and algorithm:

The coordinates of events $a, b, c, d, h, i, e, f, g, j,$ and k are given in Table 2:

The location map on the plane Cartesian coordinate system drawn according to the 11 event coordinates in Table 2 is shown in Figure 5.

The improved ant colony algorithm was used to count all solutions and calculate the number of optimal solutions. The consequent relationship obtained between the number of iterations and optimal solutions is shown in Figure 5.

As shown in Figure 6, a peak number of optimal solutions appeared at approximately the 9th iteration; however, instabilities and continued fluctuations were observed even afterward, indicating that the algorithm has a certain randomness. Convergence is apparent at the 29th iteration; after the 34th iteration, a singular optimal solution, the longest or critical path, is obtained, indicating that the convergence rate can be controlled and that the critical path can be obtained by setting appropriate parameters in the solution method.

As shown in Figure 7 and Table 3, the critical path of the optimal solution, denoted by a red line in Figure 6, is as follows: (a, b) piling plan and detail drawing \rightarrow (c, e) foundation plan (and beam) layout drawing \rightarrow (e, f) steel structure detail drawing \rightarrow (f, g) procurement of steel structure \rightarrow (g, j) erection of steel structure \rightarrow (j, k) installation of static equipment.

The improved ant colony algorithm was used to calculate the critical path of this case, thus demonstrating its feasibility for critical path calculation in other cases as well. After testing its performance, it is concluded that the algorithm can be improved by setting appropriate parameters.

6. Discussion

This study contributes to the literature by exploratively examining the intelligent critical path computation algorithm of the project scheduling network. P6 (Oracle Primavera) and MS Project software can apply this method for scheduling calculation for complex project scheduling.

There has been limited research into the comparison of the effectiveness of other intelligent algorithms and improved ant colony algorithm in computing critical path. There has been limited research into the visibility and pheromone settings which can accelerate algorithm convergence. How to set visibility and pheromone well is a research direction of the ant colony algorithm for the project scheduling network.

7. Conclusions

In this article, an intelligent critical path computation algorithm was proposed to improve search accuracy, accelerate convergence, and achieve the optimal solution. A coal power plant engineering, procurement, and construction (EPC) project was selected to test the improved ant colony algorithm. The obtained results showed that the intelligent critical path computation algorithm is significantly better than the traditional algorithms.

The critical path method is a theoretical method used to optimise project scheduling management and scheduling preparation. A convenient and effective computation of the critical path is thus conducive to better project management; therefore, innovation in this field is necessary. This study used a double-code network diagram and the ant colony algorithm to estimate the shortest possible distance for the TSP. The algorithm was improved to determine the critical path by setting path distance and time as negative values and by keeping the transfer probability unchanged; the intelligent critical path algorithm based on the improved ant colony algorithm was thus established. Through case verification and algorithm performance tests, it was proven that the improved ant colony algorithm can solve the critical path problem.

The main findings can be summed up as follows:

- (i) A mathematical model of the critical path computation method based on the improved ant colony algorithm was established
- (ii) The intelligent computation algorithm was successfully applied to the estimation of the critical path, which makes the critical path computation intelligent
- (iii) In a project case, the improved ant colony algorithm was successfully applied to determine its critical path and its length

Data Availability

The data used to support the findings of this study are included within this article and available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (51808424 and 51478384) and the Industrial Building Environment and Energy Conservation Innovation Team of China (2017KCT-14).

References

- [1] E. W. Dijkstra, "A note on two problems in connection with graphs," *Numerische Mathematics*, vol. 1, no. 1, pp. 269–271, 1959.
- [2] F. Mirahadi and B. Y. McCabe, "EvacuSafe: a real-time model for building evacuation based on dijkstra's algorithm," *Journal of Building Engineering*, vol. 34, Article ID 101687, 2020.
- [3] D. D. Zhu and J. Q. Sun, "A new algorithm based on dijkstra for vehicle path planning considering intersection attribute," *IEEE Access*, vol. 9, pp. 19761–19775, 2021.
- [4] R. W. Conway, W. L. Maxwell, and L. W. Miller, *Theory of Scheduling*, Addison-Wesley, Boston, MA, USA, 1967.
- [5] A. A. B. Pritsker, L. J. Waiters, and P. M. Wolfe, "Multiproject scheduling with limited resources: a zero-one programming approach," *Management Science*, vol. 16, no. 1, pp. 93–108, 1969.
- [6] J. E. Kelley, "Critical-path planning and scheduling: mathematical basis," *Operations Research*, vol. 9, no. 3, pp. 296–320, 1961.
- [7] F. S. Xu and Q. Huang, "A new algorithm for finding critical path," *Chinese Computer Engineering and Applications*, vol. 24, no. 12, pp. 82–84, 2005.
- [8] M. F. Wang, "New algorithm for finding critical paths," *Chinese Computer Engineering*, vol. 34, no. 9, pp. 106–108, 2008.
- [9] X. M. Li, J. X. Qi, and Z. X. Su, "Free float theorem and algorithm of seeking the k -th order critical path," *Chinese Journal of Management Science*, vol. 12, no. 2, pp. 98–104, 2009.
- [10] J. M. Wang and W. W. Zhou, "Simple algorithm to solve the secondary critical path of CPM network based on node time difference," *Chinese Operation and Management*, vol. 2, no. 5, pp. 137–140, 2016.
- [11] J. W. Zhang and S. Zhou, "Realizations of activity network graphs and critical path algorithm based on object oriented technique," *Chinese Computer Engineering and Applications*, vol. 25, no. 4, pp. 234–237, 2016.
- [12] X. D. Han, F. Gao, and L. W. Zhang, "Novel real-time algorithm for critical path of linear network coding," *Chinese Computer Science*, vol. 47, no. 9, pp. 232–237, 2020.
- [13] X. K. Han, W. Z. Yan, and M. Lu, "Research on reasoning concerning emergency measures for industrial project scheduling control," *Advances in Civil Engineering*, vol. 2021, Article ID 5595354, 13 pages, 2021.
- [14] X. K. Han, "WBS-free scheduling method based on database relational model," *International Journal of System Assurance Engineering Management*, vol. 12, no. 3, Article ID 01106-x, 2021.
- [15] S. Y. Guo, X. G. Zhang, Y. Q. Du et al., "Path planning of coastal ships based on optimized DQN reward function," *Journal of Marine Science and Engineering*, vol. 9, no. 2, p. 210, Article ID 9020210, 2021.
- [16] W. Deng, S. F. Shang, X. Cai et al., "Quantum differential evolution with cooperative coevolution framework and hybrid mutation strategy for large scale optimization," *Knowledge-Based Systems*, vol. 224, Article ID 107080, 2021.
- [17] W. Deng, J. J. Xu, X. Z. Gao, and H. M. Zhao, "An enhanced MSIQDE algorithm with novel multiple strategies for global optimization problems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–10, Article ID 3030792, 2020.
- [18] W. Deng, J. J. Xu, H. M. Zhao et al., "A novel gate resource allocation method using improved PSO-based QEA," *IEEE Transactions on Intelligent Transportation Systems*, Article ID 3025796, pp. 1–9, 2020.
- [19] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, 1997.