

Path Optimization Algorithms Based on Graph Theory

Biyuan Yao¹, Jianhua Yin^{1,*}, Hui Zhou¹ and Wei Wu^{1,2}

¹ College of Information Science and Technology,
Hainan University, Haikou 570228, China

² Institute of Deep-sea Science and Engineering,
Chinese Academy of Sciences, Sanya 572000, China

yaobiyuanyy@163.com, yinjh@hainu.edu.cn, zhouhui@hainu.edu.cn, wuwei@idsse.ac.cn

Abstract

Transport with minimum time cost and distance remains to be an important research area in intelligent transport systems. Shortest path algorithms are primary methods to address simplified problems, which could not be well applied in high-dimensional real situations. We realized the minimum cost and maximum flow result via classical iterative algorithm based on graph theory, adjacency matrix is well applied to express the relationship between transport nodes, a topological sorting transport map is adopted to verify these approaches.

Keywords: graph theory, improved Dijkstra algorithm, Ford-Fulkerson, topological sorting, path optimization

1. Introduction

Path planning technology is a hot point in many areas with widely real applications and theoretical research, in which the path optimization is one of the core contents. It strives to find a path from the initial state to the target state collision-free path in obstacle circumstance according to certain evaluation criteria. Many practical questions and solving ways are proposed in recent years^[1]. Single-source shortest path problem of non-negative edge weights has been conducted in-depth case studies by Dijkstra algorithm, Bellman-Ford algorithm^[2-3]. Meanwhile, in regarding to network flow problem derives from transport, many useful approaches are presented to address this problem from various aspects, including general augmenting path of Ford-Fulkerson algorithm, successive shortest augmenting path of Dinic algorithm^[4-5]. In addition, stochastic flow network computing is a classical problem from optimization area, it is another influential uncertainty NP-hard problem^[6-7].

Topological sorting transport map could be utilized efficiently to obtain selected paths. Study focused on vehicle path planning equals to address the problem of the shortest transport, *i.e.*, to seek the shortest path in given two vertices from weighted graph $G=(V,E,W)$, it is a hot issue in the field of research in computer science, operations research, geographic information systems and traffic navigation systems^[8]. This paper designed systemic scheme by adjacency matrix to meet a variety of different real networks application requirements.

Researches focused on deadlock testing, avoidance and release varies from different aspects did a good help for current scholars, while deadlock avoidance from graph theory is not common. Graph theory is well applied to establish a hierarchical model database, network structure model and the relational structure established by relational theory^[8]. Therefore, this paper proposed feasible approach to deadlock avoidance based on graph theory.¹

Jianhua Yin is the corresponding author.

To design an optimized model adapt to real requirements, this article studies the problem organized as follows: section 2 analyzed the characteristics of classical optimization methods and theories related to this paper; section 3 improved graph theory method with matrix theory to establish optimal control model and illustrated the algorithms with an example; section 4 simulated and tested the model in the previous section; section 5 is the conclusion; section 6 introduces our future works.

2. Preliminary Results

2.1. Rudimental Principles Related to Graph Theory

Graph is a special data structure consists of node sets and node relationships (usually between two nodes), where nodes sets and vertex sets are expressed as $V(G)$, $E(G)$, the number of nodes n is defined as the order of graph, the number of edges of the graph is usually represented by m .

Definition 1 Let graph $G=(V, E)$, vertex set is $V=\{v_1, v_2, \dots, v_n\}$ and edge set $E=\{e_1, e_2, \dots, e_n\}$, a_{ij} is the number of edges between vertex v_i and v_j , $a_{ij}=0,1,2$, then matrix $A=A(G)=(a_{ij})_{n \times n}$ is adjacent matrix of graph G .

Definition 2 Let graph $G=(V, E)$, vertex set is $V=\{v_1, v_2, \dots, v_n\}$ and edge set $E=\{e_1, e_2, \dots, e_n\}$, m_{ij} is the adjacent count number between v_i and e_j , $m_{ij}=0,1,2$, then matrix $M=M(G)=(m_{ij})_{n \times n}$ is incidence matrix of graph G .

Definition 3 In the capacity network $G=(V, E)$, let it meets the arc flow constraints and equilibrium conditions, and has a maximum traffic, this possible flow is defined as maximum flow of network.

Definition 4 Let the capacity network $G=(V, E)$ is a directed one, for each arc $\langle u, v \rangle \in E$, the related weight value $c(u, v)$ is the capacity of arc, and $c(u, v) > 0$. Meanwhile, the minimum cut-set is the smallest cut of capacity.

2.2. Problem Description

Transport schedule between two different cities with complex road systems is a real challenge, especially in Hainan Island, because of the national highways, provincial highways and rail networks various at different levels. Vector map expression and network topology construction are rudimental tools to achieve the shortest path analysis system, efficient implementation of the shortest path search algorithm is the core foundation related to this system.

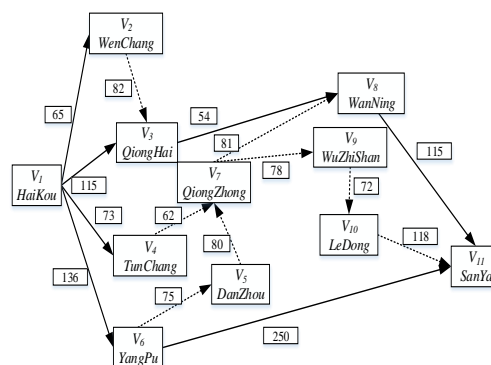


Figure 1. Hainan High-Speed Rails and Highways in Graph Description

As is indicated in Figure 1, high-speed rails and highways are shown with distances, thus formed a directed graph with weights in order to study the shortest path problem from graph theory aspect. Two vertexes and their edges could be defined clearly.

3. Optimization Modeling

In this section, we improved the Dijkstra algorithm and Warshall-Floyd algorithm by graph theory, to be specific, with adjacency matrix and incidence matrix based on graph storage theory.

3.1. Dijkstra and Warshall-Floyd Algorithm

Dijkstra algorithm is well recognized as an efficient technique to address shortest path problem. It could be adopted to compute the distance among one node to the rest, thus enable to calculate the optimized results, which is a greedy algorithm to single-source shortest path problem. Normal processing steps is shown in Figure 2, the details is described as follows ^[9,10]:

Step 1: Label the initial node V_s as $(0, S)$, initial all the rest nodes temporary mark $(T, 0)$.

Step 2: Identify all nodes adjacent to the initial node, label each node as (L, I) , where L is the distance between the two nodes, mark * represents the value is fixed and permanent.

Step 3: Based on the fixed node in the above step, find all adjacent nodes related to it (other permanent nodes excluded), recalculate their distances thus to renew results. Repeat the process until all nodes have been fixed.

Step 4: According to the permanent labels and values, search back forwards to achieve solutions.

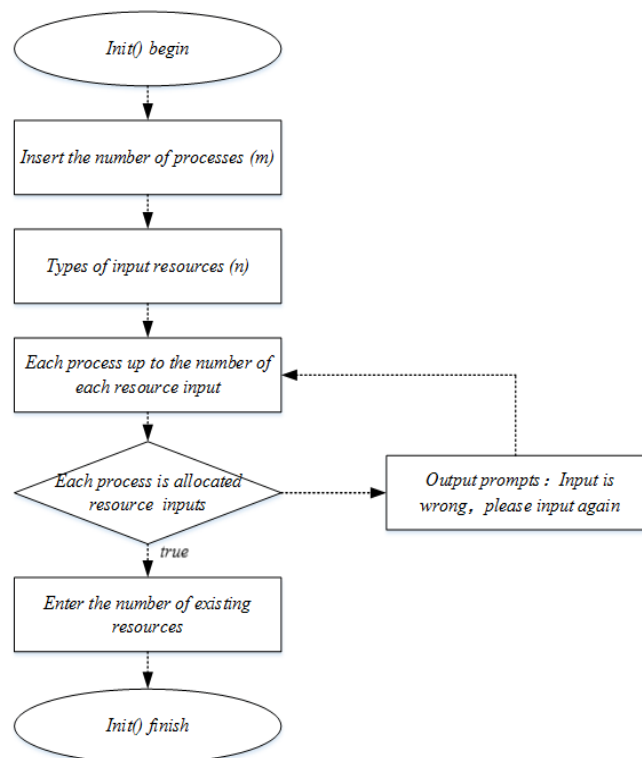


Figure 2. Dijkstra Algorithm Flowchart

3.2. Graph Expression with Adjacency Matrix and Incidence Matrix

Graph is an abstract tool to extract networks, it stores information by adjacency matrix and incidence matrix. This storage method is an influential factor to convergence speed and time complexity. The transport system aforementioned could be expressed by these matrixes as follows:

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$AI = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & -1 & 0 \end{bmatrix} \quad (1)$$

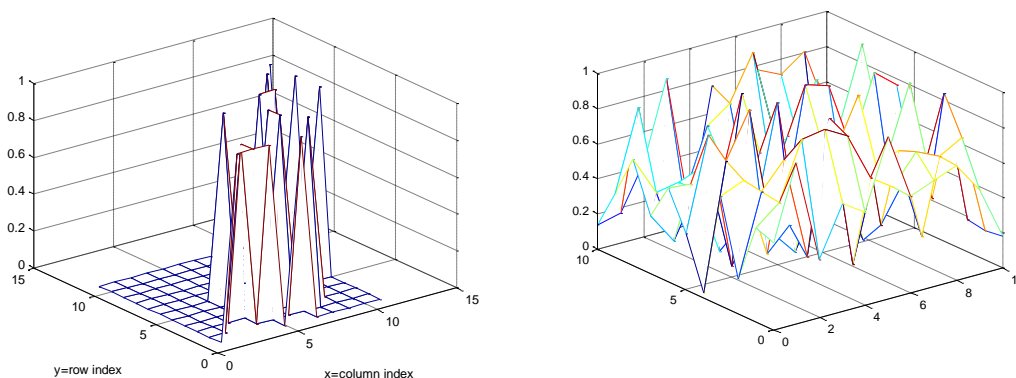
It is not necessarily symmetrical along the main diagonal in adjacency matrix, the count number of all the elements (with value 1) in the i^{th} row is the number stands for the output degree of vertex i , the count number of all the elements (with value 1) in the i^{th} column is the number stands for the input degree of vertex i .

Table 1. The Results Calculated using Dijkstra's Algorithm

To	From V_1							
V_2	65 (V_1, V_2)							
V_3	115 (V_1, V_3)	147 (V_1, V_2, V_3)						
V_4	73 (V_1, V_4)							
V_5	∞	211 (V_1, V_6, V_5)						
V_6	136 (V_1, V_6)							
V_7	∞	135 (V_1, V_4, V_7)	291 (V_1, V_6, V_5, V_7)					
V_8	∞	201 (V_1, V_2, V_3, V_8)	169 (V_1, V_3, V_8)	216 (V_1, V_4, V_7, V_8)	372 (V_1, V_6, V_5, V_7, V_8)			

V_9	∞	213 (V_1, V_4, V_7, V_9)	369 (V_1, V_6, V_5, V_7, V_9)		V_8			
V_{10}	∞	285 ($V_1, V_4, V_7, V_9, V_{10}$)	441 ($V_1, V_6, V_5, V_7, V_9, V_{10}$)					
V_{11}	∞	316 ($V_1, V_2, V_3, V_8, V_{11}$)	284 (V_1, V_3, V_8, V_{11})	331 ($V_1, V_4, V_7, V_8, V_{11}$)	487 ($V_1, V_6, V_5, V_7, V_8, V_{11}$)	403 ($V_1, V_4, V_7, V_9, V_{10}, V_{11}$)	559 ($V_1, V_6, V_5, V_7, V_9, V_{10}, V_{11}$)	386 (V_1, V_6, V_{11})
V_i	V_2	V_4, V_7	V_3, V_8	V_5, V_6		V_{11}	V_9, V_{10}	

It is not necessarily symmetrical along the main diagonal in adjacency matrix, the count number of all the elements (with value 1) in the i^{th} row is the number stands for the output degree of vertex i , the count number of all the elements (with value 1) in the i^{th} column is the number stands for the input degree of vertex i .



(a) Renderings of the Adjacency Matrix

(b) Relation Diagram between Matrixes

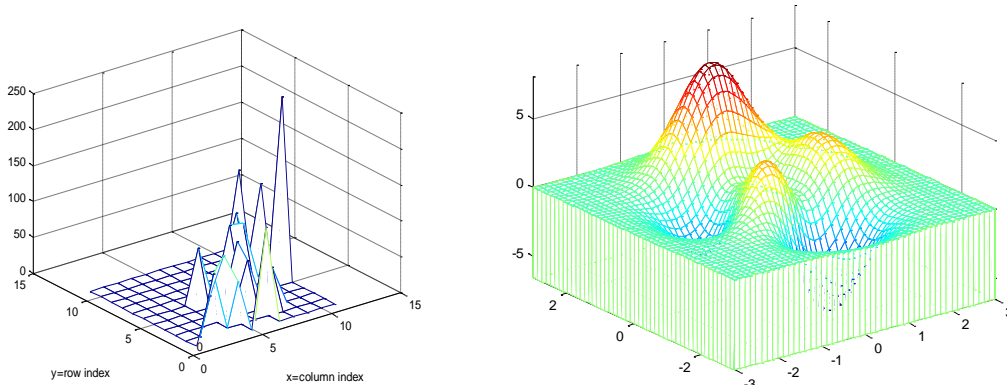
Figure 3. Relation Diagram of the Adjacency Matrix and Incidence Matrix

Mutual position could be read from adjacency matrix, Mutual connection could be read from incident matrix, therefore, three dimensional diagram is a good section to show this relationships^[11,12]. The three-dimensional renderings of the adjacency matrix is shown in Figure 3, it shows that the mutual position and straightforward connection between the network nodes.

According to the established data presenting model in Figure 2 and Figure 3, adjacency matrix topology data matrix was achieved:

$$A2 = \begin{bmatrix} 0 & 65 & 115 & 73 & \infty & 136 & \infty & \infty & \infty & \infty & \infty \\ \infty & 0 & 82 & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 0 & \infty & \infty & \infty & \infty & 54 & \infty & \infty & \infty \\ \infty & \infty & \infty & 0 & \infty & \infty & 62 & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & 0 & \infty & 80 & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & 75 & 0 & \infty & \infty & \infty & \infty & 250 \\ \infty & \infty & \infty & \infty & \infty & \infty & 0 & 81 & 78 & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & 0 & \infty & \infty & 115 \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & 0 & 72 & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & 0 & 118 \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & 0 \end{bmatrix} \quad (2)$$

Each element in the matrix represents the path weights between every two nodes, ∞ represents two nodes are not connected, the topology of the nodes could be reflected in Figure 4, in (a) describes the distances between two locations, and the global circumference is indicated in (b).



(a) Distances among Locations (b) Transport Global Smooth Characters

Figure 4. Path Weights among Nodes

3.3 Graph Algorithm with Minimum Cost and Maximum Flow

In real circumstances, different road sections involves in different charges. In this section, minimum cost and maximum flow are parallel targets in optimization, the cargo amount and paths are within consideration [14].

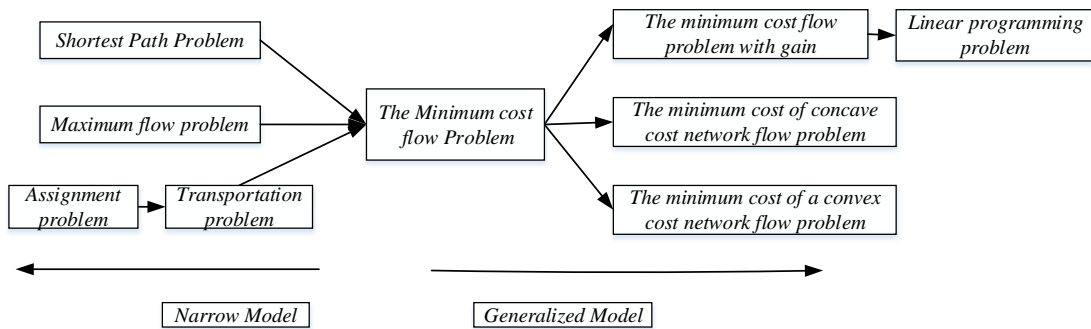


Figure 5. Model of Minimum Cost Flow Problem

In Figure 2, let weights stands for maximum capacity, introduce Ford-Fulkerson label algorithm to develop programs to ensure perfect section from V_I to V_{II} .

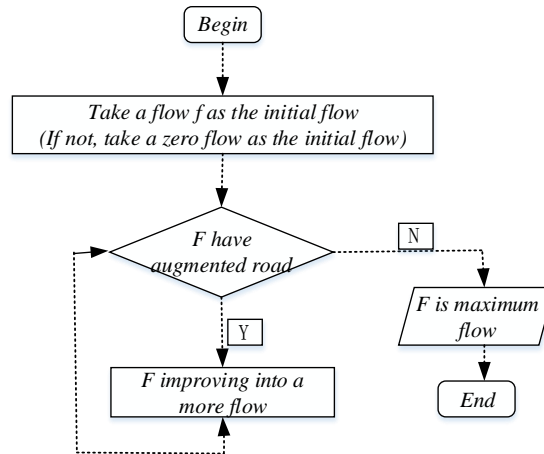


Figure 6. Processing Flow of Seeking Maximum Flow

Let $n=11$, maximum traffic $w=252$, Label $e=[12 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$, maximum flow

$$q = \begin{bmatrix} 0 & 0 & 54 & 62 & 0 & 136 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 54 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 62 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 136 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 61 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 115 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (3)$$

Further considerations also involves flow fee, which increases the dimensional computing for current systems. For example, in a standard transportation problem, it often requires to complete certain transport tasks in certain premise cargo^[15]. Therefore, we introduce Busacker-Gowan iterative algorithm for minimum cost of flow, its main steps are as follows (Let network $G=(V, E, C)$, taking the initial feasible flow f is zero flow):

Step 1 : Construct a directed weighted graph $G_f=(V, E_f, F)$, for any $v_i v_j \in E$: when $f_{ij}=0$, $v_i v_j \in E_f, F(v_i v_j)=b_{ij}$; when $f_{ij}=C_{ij}$, $v_j v_i \in E_f, F(v_j v_i)=-b_{ij}$; when $0 < f_{ij} < C_{ij}$, $v_i v_j \in E_f, F(v_i v_j)=b_{ij}$; $v_j v_i \in E_f, F(v_j v_i)=-b_{ij}$.

Step 2 : Calculate the shortest path u from v_s to v_t , if there exist u , turn to *Step 3*; otherwise current f is the final results.

Step 3 : Renew flow. Let $\delta_{ij}=C_{ij}-f_{ij}, v_i v_j \in u^+, \delta_{ij}=f_{ij}, v_i v_j \in u^-, \delta=\min\{\delta_{ij} | v_i v_j \in u\}$. Redefine flow $f=\{f_{ij}\} = f_{ij}+\delta, v_i v_j \in u^+, f_{ij}-\delta, v_i v_j \in u^-$, or f_{ij} . If the resulting maximum flow W_f no less than predetermined flow value, then reduce δ with appropriate value to ensure W_f equal to the predetermined value W_{f0} , f is the minimum flow, process terminated, otherwise turn to step 1.

We assume unit flow on the arc of the road network in the cost matrix is:

$$A_3 = \begin{bmatrix} 0 & 32 & 65 & 29 & 0 & 82 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 41 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 24 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 30 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 40 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 26 & 0 & 0 & 0 & 0 & 0 & 81 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 41 & 29 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 76 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 34 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 75 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (4)$$

From preliminary matrix A_2 and A_3 , the maximum flow $wf = 135$, the minimum cost $z wf = 48468$, The maximum flow matrix with minimum cost is:

$$f = \begin{bmatrix} 0 & 0 & 24 & 29 & 0 & 82 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 24 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 29 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 81 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 30 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 54 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (5)$$

4. Results Test with Topological Sorting of Transport

Topological sorting is a method to determine whether there is a circuit in the map. In terms of graph, topological sorting is a method to directed graph without circles, it could not be used in undirected graph or circled graph. Sorting algorithm is achieved in the following steps^[16-17]:

- Step 1: Select a zero-degree vertex and output from AOV network;
- Step 2: Delete the vertex and all edges emanating from the vertex of AOV network.
- Step 3: Repeat steps 1 and 2, until no vertex-degree zero.

Conducted in accordance with the above topological sorting method, and the results are bound to limited in two cases:

Scenario 1: All vertices are output, the entire topological sorting completed;

Scenario 2: Some vertexes are not output, but the rest of the figure no penetration of vertices 0, so that topological sort cannot be continued, which indicates that this graph is acyclic graph^[19].

From Figure 2 in subsection 2 of section 2, $V_1V_2V_3V_4V_5V_6V_7V_8V_9V_{10}V_{11}$ is a topological sequence from directed acyclic graph, there may be more than one sequence, and the topological sorting process is run as follows:

- Step 1: Select a zero-degree vertex V_1 , delete V_1 and each edges issued from it.
- Step 2: Select a zero-degree vertex V_6 , delete V_6 and each edges issued from it.
- Step 3: Select a zero-degree vertex V_4 , delete V_4 and each edges issued from it.
-
- Step 10: Select a zero-degree vertex V_8 , delete V_8 and each edges issued from it.

Step 11: Select a zero-degree vertex V_{11} , delete V_{11} .

The final topological sequence obtained is $V_1V_6V_4V_5V_7V_9V_{10}V_2V_3V_8V_{11}$, there are two paths from the source vertex V_1 to V_3 with the length 115 and 147 respectively, and as a result, V_3 is the first occurrence with time mark 147. Similarly, we can find the longest travel distance and cost corresponding to the needs of the project, to make AOE web project completed efficiently.

5. Conclusions

The optimization model on path scheduling based on Graph Theory does achieved accuracy and adaptive ability in real scenarios, especially for middle/small scale array. Improvements based on classical algorithms can maintain core indexes and provide clear description in geometrical language.

Dijkstra algorithm and Warshall-Floyd algorithm are classical methods related to this problem without consideration about complex weights values^[18]. According to graph description, we realized the problem reconstruction with a clear visual reflection and analysis convenience, meanwhile, adjacent matrix and incident matrix could be well defined and interpreted consistent with real backgrounds. Furthermore, through the analysis methods aforementioned above, we came up with the idea about path optimization form global view, which includes the weights of expenses form distance, time and extra factors.

6. Related Works

In most theoretical studies, the maximum flow problem included research on the flow capacity, but failed to pay attention to the cost of circulation, in fact, it is an important factor in real applications^[19-21]. If we only consider the unit transport costs of goods, then the question turns to be shortest path problem. Thus, the shortest path problem is a special case of the global scenario, it is necessary corollary idea of high-dimensional solution to the problem, which is the focus of our future research.

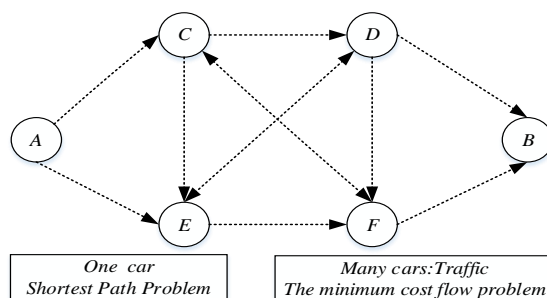


Figure 8. Relationship between the Minimum Cost and Shortest Distance

Acknowledgements

The project is funded by National Natural Science Foundation of China (Grant No. 11561017); National Natural Science Foundation of China (Grant No. 61440019; Grant No. 11261015); Graduate Student Innovation Research Project of Hainan Province (NO.Hys2015-38).

References

- [1] Q Gao, "MapX-based navigation software design and implementation", Beijing Jiaotong University, Beijing(2008).

- [2] H. Wang, Q. Huang, C. Li and B. Zhe, "Editor, Graph Theory Algorithms Based on MATLAB", Beijing University of Aeronautics and Astronautics Press, Beijing (2010).
- [3] G. suixiang, "Graph Theory and Network Flow Theory", Higher Education Press, Beijing (2010).
- [4] Guiping Wang, Yan Wang and Jiachen Ren, Editor, Graph Algorithm Theory, Implementation and Applications, Peking University Press, Beijing (2011)
- [5] B Zhang, J Lv, *et al.* "An uncertain figure most reliable solution for the maximum flow problem [J]", 7(2014).
- [6] J Du, "Operational Research. Zhejiang University Press", Zhejiang (2010).
- [7] X. Wang, G. Zhang, J. Zhao, *et al.* "A modified membrane-inspired algorithm based on particle swarm optimization for mobile robot path planning [J]", International Journal of Computers, Communications & Control (IJCCC), 10(2015)
- [8] L Li, "Selection of the optimum conditions under generalized cost multimodal transport modes [M]", 2(2006)
- [9] Z. Zhang *et al.* "Proficient in MATLAB 6.5 version [M]", Beijing University of Aeronautics and Astronautics Press, Beijing (2003)
- [10] K. V. Raemdonck, C. Macharis and O. Mairesse, "Risk analysis system for the transport of hazardous materials [J]", Journal of Safety Research, 45, pp. 55-63(2013).
- [11] M.R. Hassan, "An efficient method to solve least-cost minimum spanning tree (LC-MST) problem [J]", 24: pp. 101-105(2012).
- [12] S. Ebrahimi, P. Payvandy, "Efficient constrained synthesis of path generating four-bar mechanisms based on the heuristic optimization algorithms [J]", Mechanism and Machine Theory, 85: pp. 189-204, (2015).
- [13] W. Bi, W. Zang, T. Liu, "Three-Dimensional Terrain Modeling and Path Optimization Based on Google Earth and ACIS", Human Centered Computing, Springer International Publishing, (2015), pp. 872-879.
- [14] R T Esfahani, Z Zojaji, "Optimization of finite element model of laser forming in circular path using genetic algorithms and ANFIS [J]" Soft Computing, (2015), pp. 1-15.
- [15] Y. Chen, S. Shen, T. Chen, *et al.* Path Optimization Study for Vehicles Evacuation based on Dijkstra Algorithm[J]. Procedia Engineering, 71, (2014), pp. 159-165
- [16] X. Zhang, Q. Wang, A. Adamatzky, *et al.* A biologically inspired optimization algorithm for solving fuzzy shortest path problems with mixed fuzzy arc lengths [J]. Journal of Optimization Theory and Applications, vol. 163, no. 3, (2014), pp. 1049-1056
- [17] W U Fan, P. Li, "Localization Algorithm Based on Node Density and Path Optimization for Wireless Sensor Networks [J]", Chinese Journal of Sensors and Actuators, 11: 018(2014).
- [18] Q. Wang, A. Zhang, L. Qi, "Three-dimensional path planning for UAV based on improved pso algorithm[C]", The 26th Chinese. IEEE Control and Decision Conference (2014 CCDC), (2014), pp. 3981-3985.
- [19] B. Vallade, T. Nakashima, "Improving Particle Swarm Optimization Algorithm and Its Application to Physical Travelling Salesman Problems with a Dynamic Search Space [M]", Applied Computing and Information Technology. Springer International Publishing, (2014), pp. 105-119.
- [20] F. Neumann, C. Witt, "Bioinspired Computation in Combinatorial Optimization: Algorithms and Their Computational Complexity[C]", Proceedings of the 15th annual conference companion on Genetic and evolutionary computation. ACM, (2013), pp. 567-590.
- [21] V. Roberge, M. Tarbouchi, G. Labonté, "Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning [J]", Industrial Informatics, IEEE Transactions on, vol. 9, no. 1, pp. 132-141(2013).

Authors



Biyuan Yao. Master of Hainan University, her research interest covers Graph Theory with Applications.



Jianhua Yin. Professor of Hainan University, MS supervisor. His research interest: Graph Theory with Applications.



Hui Zhou. Hui Zhou joined Hainan University as a staff college since May 2011. His research interests include computer network, digital tourism, and cluster



Wei Wu. Research associate of Institute of Deep-sea Science and Engineering, CAS. His research interest covers data and knowledge engineering.

