Comput Syst Sci & Eng (2018) 2: 71–85

International Journal of
Computer Systems
Science & Engineering

The Optimization Reachability Query of Large scale Multi-attribute Constraints Directed Graph

Kehong Zhang and Keqiu Li

School of Computer Science and Technology, Dalian University of Technology, Dalian, China
E-mail: hawkzhang1@163.com, keqiu@dlut.edu.cn

Today, many applications such as social network and biological network develop rapidly,the graph data will be expanded constantly on a large scale. Some classic methods can not effectively solve this scale of the graph data. In the reachability query, many technologies such as N-Hop, tree, interval labels, uncertain graph processing are emerging, they also solve a lot of questions about reachability query of graph. But, these methods have not put forward the effective solution for the new issues of the multiattribute constraints reachability on directed graph. In this paper, TCRQDG algorithm effectively solves this new problem. Firstly it optimizes the multiattribute constraints with decision making technology; secondly the algorithm achieves fast and accurate query by integrating with the Create virtual vertex expand, conditions filtering, cycles contraction, interval label and other technology. TCRQDG algorithm can not only effectively solve the new problem, but also provide technical support for multiple constraints optimization decisions of network transmission, transport and logistics, software testing and other applications

Keywords: Multiattribute constraints; reachability; directed graph;interval labels;contraction

1. INTRODUCTION

Nowadays, the scale of graph data are constantly expanded by social networks, biological networks,semantic web, and the other real-world applications.As the main direction of graph data research, the reachability query has important practical value and theoretical significance. The researchers pay attention to improving the algorithm efficiency of graph reachability query. It is one of the hotspots in data research.

The graph reachability query expression is as follows, there are a directed graph G and the two vertices u, v in the graph G, if there is a path from u to v, that is to say that the two vertices u, v are reachability in the graph. This is a simple and difficult question. Due to the increase in the largescale graph data, the query of reachability will face new challenges, especially on the basis of graph of reachability. Many people put forward the concept of constraint reachability on graph query, the two vertices u, v are not only simple reachability but also meet the constraints condition of rules, distance, weight and so on. At present, there are much query algorithm of reachability.For example, transitive closure algorithm, the shortest path algorithm[32],N-Hop method [4, 1, 13, 23, 24, 28, 27, 31], tree method [16, 26, 9, 3],interval labels [10, 11, 21, 22, 29, 31], uncertain graph processing[7, 33, 26]etc. These algorithms mainly look for ways to reduce the consumption of time and space, at the same time, it also considers effective and accurate query results. Generally, the technical indicator of these algorithms mainly have three aspect: the first,query processing time;the second,whether it needs index and considers construction time and space size of the index;the third,whether it support dynamic update graph.

In real world, the reachability query is not only a simple judgment about the connection relationship, but also meeting certain constraints condition [12, 14, 17, 19, 15]. At present, it usually assumes that the edges weight is a single attribute cost. In fact, the relationship between the different entities can be examined from various angles, and the relationship representative of edge needs to use multiple attribute cost. For example, in the transportation network, the highway cost of the two cities is usually evaluated by the length, cost, congestion degree, security coefficient and so on. Therefore, in graph model description of the network, the cost of the edge should be multidimensional and thus any two vertices reachability evaluation is multidimensional.

In addition, the optimization reachability query of the multiple attributes decision making, has practical guiding significance in real life. For example, in the field of transport planning, it is necessary to consider the cost, time, safety factors. We also consider the driver's familiarity of road, psychological predisposition and other factors. The multiple attribute decision technology can effectively solve the problem by this time, the overall rating is selected and optimized on the basis of the comprehensive evaluation. Finally, the optimization results of users expectation will be obtained.

So, this article presented TCRQDG algorithm. It used decision making, create virtual vertex, conditions filtering, cycles contraction, interval labels and other technology to salve the multi-attribute constraints reachability on directed graph. In the experiment section, this article also compare with the new algorithm and classical algorithm. This algorithm effectively satisfy the reachability query of multiple attribute constraints.

1.1 Contribution

(1) At present, the research of multiple attribute constraints reachability is few, it mainly focuses on no constraint and single constraint. After the article studies the multiple attribute decision making method. Based on TOPSIS method, the algorithm considers the different attribute characteristics to build standardized decision matrix. The paper calculates multiple attribute comprehensive evaluation index with building standardized decision matrix, this comprehensive evaluation index determines the reachability of optimization. In this process, we must consider the correlation of multiple attributes and the importance of different attribute about decision-makers intentions;

(2) The reachability problem study builds an index with directed acyclic graph. This method essence will lead to the loss of much information on query. In order to avoid the loss of information, the article uses the annotation method of cycle vertex. This method can guarantee the information completeness, it also can achieve rapid reachability queries with interval label technology;

(3) The interval label query technology of largescale directed graph, its idea indicates that if $\mu \rightarrow \nu$ exist, then $L_\nu \subseteq L_\mu$. So far, this technology does not consider the constraints problem between one vertex and other vertex. In other words,the presentation of constraint reachability interval queries is almost none on directed graph. This article's method adds constraint conditions to interval technology, and uses multiple interval technology to realize constraints reachability query in directed graph;

(4) The interval label technology indeed accelerates the reachability query. The method needs to build multi labels to offset the exception of single label. The exception is that if $L_\mu = [s_\mu, e_\mu]$ includes $L_\nu = [s_\nu, e_\nu]$, but there are the unreachability problems of vertex u and v. In addition, many labels are needed to consume more time and space. In order to reduce the number of labels, this article uses constraint conditions to filtrate graph in initialized. If the weight of edge meets constraint condition, it will reserve, otherwise the edge will be removed. So, this method will reduce the graph complexity and the number of label, at the same time it will improve the efficiency of the query.

2. RELATED RESEARCH

2.1 The Basic Approach Research in Reachability

The paper puts forward an idea of k step reachability in query as discussed by J. Cheng et al [14], it mainly have two methods. The first method calculates minimum vertex cover of k step reachability; the second method randomly chooses path p with k step length, deletes the vertex set $\{v_0 \cdots v_k\}$ and its while connecting edge, merge $\{v_0 \cdots v_k\}$ into the set S, and repeating it until there is no path of k step length in graph. K. Xu et al [17] introduced a query with label constraint reachability. I.B. Dhia [12] puts forward the main method of dealing with the constraint conditions, it creates multiple set of each constraints condition, each query access multiple set, it will affect the efficiency of the query. As discussed by M. Qiao et al [19],it mainly introduced the method of query with constraints reachability, the first, the undirected graph is converted into a minimum spanning tree; the second, the minimum spanning tree is converted into index tree, it is based on the edge; the last, the query of constraints reachability is run by index tree. As discussed by R.M. Jin et al [25], its main idea is to set attribute affiliation, if a set A meets the nature of constraints reachability, its superset will meet the requirements of reachability, so the superset is astringed as minimum tag set.It is mainly through the sample drawn, setting up a weight estimation $\hat{X}_{e,n}$ and error bounds $\varepsilon_{e,n}$, these will be convenient queries of reachability.

Above-mentioned methods has really improved the query of reachability. But we think that there are some shortcomings, to sum it up, there are the following several aspects: the first, some methods in itself are the NP problems; the second, in dealing with the data structure of graph, the graph was simplified into a tree, or it creates the candidate set, the selected vertex set is removed from the original set. In spite of creating tree or candidate set, the information missing is a common problem;the last, from the above-mentioned method,the basis is undirected graph, all don't consider directed graph. But the constraints reachability problem of directed graph have essential difference with undirected graph. In other words, these methods of undirected graph are not sure effectively for directed graph.

2.2 The Research of Interval Label Technology for Reachability

In the reachability query of the directed graph, some methods are introduced, such as N-Hop, tree, interval label technology as discussed elsewhere et al [9, 21, 22, 29, 31]. The interval label technology is more effective than other technology. It mainly has two technologies with min-post label and pre-post label. The pre-post assigned interval label $L_\mu = [s_\mu, e_\mu]$ to each vertex, the s_μ is sequence number of preorder traversal from root vertex, e_μ is sequence number of postorder traversal. About min-post label, e_μ is sequence number of postorder traversal, but e_μ is a minimum sequence number of u's child vertices. If there are two interval labels $L_\mu = [s_\mu, e_\mu]$ and $L_\nu = [s_\nu, e_\nu]$, the $\mu \to \nu$ was true, then $L_\nu \subseteq L_\mu$.

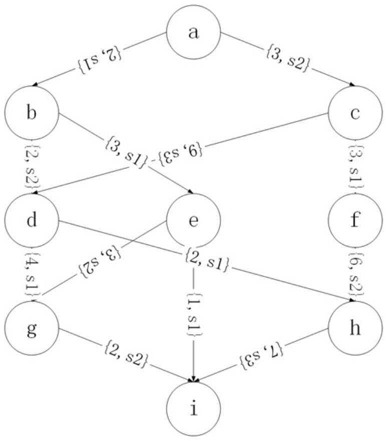

Figure 1 The multi-attribute directed graph.

However the interval label technology of min-post has an exception. The exception is that vertex u can not reach vertex v, but the label $L_\mu = [s_\mu, e_\mu]$ include $L_\nu = [s_\nu, e_\nu]$. As shown in Figure 2, there are label value exceptions of min-post technology after traversal the Figure.1. Such as the label value of vertex b is $L_b = [1, 6]$, the label value of vertex c is $L_c = [1, 8]$. The $c \to b$ is reachability by the principle of interval label, but it is not the true on the original graph.

For the exception problem, H Yildirim et al [11],[10] used multiple label technology to solve this abnormal information. At this point, there are $L_\mu = L_\mu^1, L_\mu^2, \cdots, L_\mu^d, L_\mu^i (1 \le i \le d)$ are the i times label value with after randomly traversal the DAG. If and only if L_μ^i ,$i \in [1, d]$, all are $L_\nu^i \subseteq L_\mu^i$,then $L_\nu \subseteq L_\mu$ is satisfied. It is reachability. Whereas it is unreachability.

The above method indeed solves some reachability problem of the directed graph. But there are also some shortcomings:

(1) The main method is that the directed graph is converted into minimum spanning tree, then interval label are set up by the minimum spanning tree. Strictly speaking, this is an approximate treatment method. There must exist artificial lack of information;

(2) All methods do not take the various conditions into directed graph, these lead to shortcomings in realistic constraint reachability query;

(3) The number of multiple interval label is established until there is not an exception. This will lead to setting up many labels, and it will consume too many resources.

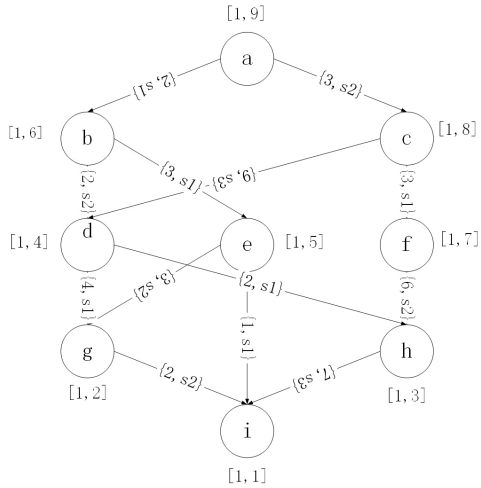

Figure 2 The interval label of the directed graph with Min-Post traversing.

2.3 The Research of Multiple Attribute Decision Making

Multiple attribute decision making generally includes three aspects about the standardization of the decision matrix, attribute weights, the comprehensive sequencing of scheme.According to their natures, attributes are divided into quantitative and qualitative style, according to its specific meaning, it can be divided into the efficiency, cost, fixed type, range, deviation, deviation interval.The attribute value of each decision scheme have exact numbers, interval number, fuzzy number, triangular fuzzy number, trapezoid fuzzy number, and so on.

Because the value of each property has a different dimension, the standardization process will become more essential step before attribute decision making, and different standardization process will directly affect the outcome of decisions. The commonly used method of multiple attribute decision making includes a simple linear weighting method, TOPSIS, AHP, ELECTRE, etc. TOPSIS method mainly constructs the positive ideal value and negative ideal value of multiple attribute decision making problems as discussed by C.L. Hwang [2]. The solution calculates the distance between each scheme value and ideal value, the evaluation basis is close to the positive ideal value and far away from the negative ideal value. The last, the sort of the scheme is evaluated by the synthesis score of two distance. TOPSIS is an effective method for multiple attribute decision making. It has many special advantages, for example intuitive geometric meaning, the raw data using more fully, less information loss, wide application range, etc. Since TOPSIS is put forward, its application range is widening. As discussed by V.P. Agrawal [30],it uses TOPSIS to select tools in agile manufacturing process. As discussed by G. Kim [8], they utilize artificial neural network technology to solve the TOPSIS attribute index weight. As discussed by C.T. Chen [4], it makes use of TOPSIS to solve the multiplayer and multi-criteria decision making problems in fuzzy environment. As discussed by M.A. Abo-Sinna et al [18], they expand it to multi-objective nonlinear programming problem.

3. PRELIMINARIES

There is a directed graph $G = (V, E, A, I)$, V is the set of vertices, E is the set of edges, $A \subset R$ is the multiple attribute values set of the real world, $A = (a_1, a_2 \cdots, a_i \cdots, a_n)$, a_i is the ith attribute values of edge. $I : E \rightarrow A$ is a function, it distributes a comprehensive evaluation index $I(e) \in R$ for each edge $e \in E$. A path $P(\mu, \nu) = \{\mu, \nu_1, \nu_2, \cdots, \nu_d, \nu\}$ from u to v, $\{\mu, \nu_1, \nu_2, \cdots, \nu_d, \nu\} \subseteq V$ and $\{(\mu, \nu_1), \cdots, (\nu_d, \nu)\} \subset E$, If e is a edge of path P, e belong to path P, it is marked as $e \in P$.

If each edge in the directed graph has multiple attributes, and each attribute has the constraint condition. So for each vertex, the different attribute of each outgoing edge or each constraint conditions constitutes the decision matrix. As shown in Eq. (1):

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1i} & \cdots & a_{1n} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ a_{i1} & \cdots & a_{ii} & \cdots & a_{in} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ a_{m1} & \cdots & a_{mi} & \cdots & a_{mn} \end{bmatrix} \tag{1}$$

In Eq. (1), $M = 1, 2, \cdots, m$ is subscript set of outgoing edge, $N = 1, 2, \cdots, n$ is subscript sets of attribute. Such as $a_{i1}, \cdots, a_{ii}, \cdots, a_{in}$ are all attribute values of the ith outgoing edge, and $a_{1j}, \cdots, a_{ij}, \cdots, a_{mj}$ are all value of the jth attribute. $A = \{a_{ij}\}_{m\times n}$ is the matrix of multiple attribute decision making problems.

Theorem 1 *There is a graph $G = (V, E, A, I)$, the query of multiple attribute constraints reachability is $q = (\mu, \nu, C)$,$\mu, \nu \in V$,$C = (c_1, c_2, \cdots, c_n)$ is multiple attribute constrained conditions, such as $c \geqslant x$, $c \leqslant y$ or $c \in [x, y]$, q is the query whether there is such a path $P(\mu, \nu)$, for $\forall e \in P(\mu, \nu)$. If all properties of each edge meet the constraint condition, the path $P(\mu, \nu)$ is reachability[19].*

Example 1: As shown in Figure 1, the first condition is numerical value. The second condition describes the state, the state value is $\{s_1, s_2, s_3\}$. For example,the first condition value is required less than or equal to 3,the second condition state is not s_3, the query judge whether the vertex A and vertex I is reachability, that is $q = (a, i, \leqslant 3, \neq s_3)$. Through searching we have found the graph has $P(a, i) = \{a, b, e, i\}$, and for $\forall e \in P(a, i)$,there are $\omega(e) \leqslant 3$ and $s \neq 3$, thus it is concluded that a and i was reachability under the constraints condition.

The above discussion focuses on edge's multiple attribute and ignores the vertex itself. If the value of vertex is considered, the query of graph will become complicated. In order to simplify the problem, this article uses the Create virtual vertex expand method as discussed by M. Qiao et al [19].

Create virtual vertex expand method: If some vertices of directed graph have multiple attribute, it is the method that these vertex are extended to the edge. First of all, we created each corresponding virtual vertex with the each vertex of multiple attribute value; the second, the multiple attribute value of original vertex is assigned to the edge of the original vertex and its expanding virtual vertex. The expanding virtual vertex instead of the corresponding original vertex as the starting vertex, the terminal vertex of ingoing edge keeps the same. For example, if the original path is $a \rightarrow b$, the vertex has multiple attribute value. At the moment, the corresponding virtual vertex a' is constructed for a. The attribute value of vertex a is translated into the $a \rightarrow b'$ edge's multiple attribute value. The edge $a \rightarrow b$ is translated into $a \rightarrow a'$ and $a' \rightarrow b$, the value of $a \rightarrow a'$ edge is the attribute value of vertex a. The attribute value of $a' \rightarrow b$ edge is the original edge $a \rightarrow b$ value. So the multiple attribute value of edge and vertex are simplified as single edge constraints problem. This method will improve the efficiency of the program.

Lemma 1 *If the vertices of original graph G have constraints attribute value, and there is a path that it meets Theorem 1. The constraint attribute value of vertices are expanded by Create virtual vertex expand method in G, there is still a path that it meets Theorem 1.*

Proof. *(1) The invariance of the constraint attribute value. This method is only equivalent value transfer from the original vertex to the edge of the original vertex and the newly virtual vertex, the size of the value didn't have any change. So the constraint attribute value doesn't not change.*

(2) The invariance of the connection. Because the Create virtual vertex expand method is only changed the starting vertex of the edge, the terminal vertex of ingoing edge did not change, the constraint attribute value of vertex are changed to the edge constraint of original vertex and the virtual vertex. If the vertex constraint attribute value of original graph G meets the constraint conditions, the edge of original vertex and the virtual vertex also satisfy the constraint conditions in extension graph G', so this method doesn't change the edge connectivity.

Through the above(1)and(2), Lemma 1 come into existence.

Example 2: As shown in Figure 3, not only each edge has attribute value, but also each vertex also has attribute value. This kind of situation is equivalent to a commodity logistics process, its speed and efficiency are not only affected by the condition of transport tool and route, but also the storage and handling capacity of each logistics center.

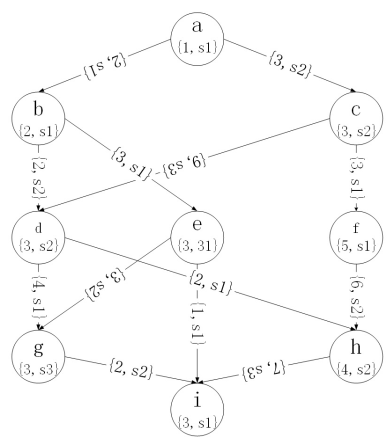

Figure 3 All of vertices and edges have multi-attribute.

Thus the virtual vertex is created for each corresponding vertex by the Create virtual vertex expand method. For example, the attribute value of vertex b is s_1 and 2, the virtual vertex b' is created, vertex a still connect the vertex b, Vertex $b's$ attribute value is translated into the attribute value of edge $b \to b'$, the edge's starting vertex from b is altered as vertex b'. So the expansion of vertex b is accomplished by the Create virtual vertex expand, similarly the other vertices are completed, the final results is shown in Figure 4.

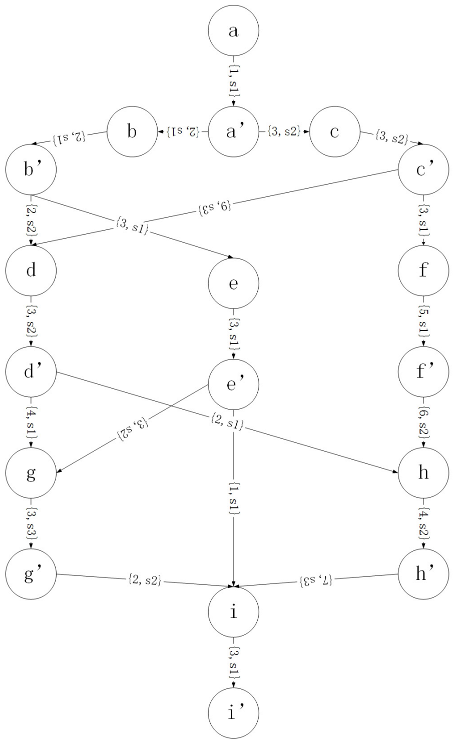

Figure 4 The expanding directed graph.

As shown in Figure 4, the query is whether the $a \to i$ can reach the constraint condition. The condition 1 is less than or equal to 3,the condition 2 state is not s_3,which is $q = (a, i, \leqslant 3, \neq s_3)$. By searching it, Figure 4 is found that there are $P(a,i) = \{a, a', b, b', e, e', i\}$,in addition, for $\forall e \in P(a,i)$, existing $attribute_1(e) \leqslant 3$ and $attribute_2(e) \neq s_3$. So, $a \to i$ was reachability under the condition of constraints. Because the technology of Create virtual vertex expand is easy to convert the constraints of vertices as the edge's constraints, so this article will focus on the constraints reachability of edge in the subsequent section.

3.1 The Interval Label

Theorem 2 *There are a directed graph $G = (V, E)$, Where $|V| = n$ vertex,$|E| = m$ edges, each vertex u is distributed an interval label $L_\mu = [s_\mu, e_\mu]$ by a certain order, the first distribution value is s_μ, the second distribution value is e_μ, the interval label of vertex u,v are L_μ and L_ν respectively. If there are $L_\nu \subseteq L_\mu (s_\mu \leq s_\nu$ and $e_\nu \leq e_\mu)$, the vertex u can reach v.*

The interval label has many method, but there are two basic methods, the first label technology is pre-post, the second label technology is min-post.

When the pre-post label technology traverses the directed graph, it accords to depth-first traversal from a certain vertex. When all the vertices are traversed, each vertex has two serial number values,the first value is first traversal sequence number value s_μ, the second serial number value is the last time backtracking u vertex's value e_μ, So the interval label of vertex u is $L_\mu = [s_\mu, e_\mu]$.

Example 3: The vertex set $\{a, b, c, d, e, f, g, h, i\}$ constitutes a directed graph in Figure 1, the pre-post label technology starts in depth-first traversal from vertex a. When all the vertices are traversed, the method will get all vertex's interval labels, as shown in Figure 5. At this time of the vertex b, its first traversal value is 2, the last time traverse value is 13, this constitutes interval label $L_b[2, 13]$. According to this principle [11, 20], the other vertex's interval label will be obtained. When the result is queried whether vertex b can reach the vertex i, only it needs to compare two interval labels value of the vertex b, i. Since the interval label of b is $L_b[2, 13]$, the interval label of i is $L_i[5, 6]$, because $(s_b = 2) < (s_i = 5)$ and $(e_i = 6) < (e_b = 13)$, so $b \to i$ is reachability.

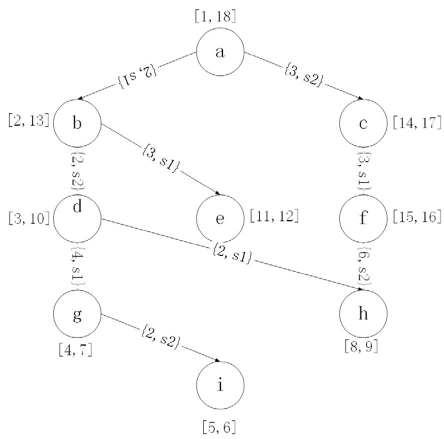

Figure 5 The interval label with Pre-Post technology.

Based on the directed graph traversal, the directed graph is reduced as directed spanning tree with pre-post label technology. Within the connection scope of the tree, it can quickly query reachability between vertices, but there is the phenomenon of information loss for directed graph. Such as shown in Figure 1, the vertex c can reach vertex d, but they are unreachability in Figure 5.

The interval label value of Min-Post technology are $L_\mu[_\mu, e_\mu]$, where s_μ is minimum value for the descendant vertex. It is divided into two kinds of circumstances, the first, the vertex is not a leaf vertex, there is $s_\mu = min\{s_v | v \in children(\mu)\}$; the second, if the vertex is a leaf vertices, there is $s_\mu = e_\mu$, e_μ was the last time traversal sequence number value of the vertex.

Example 4: The vertex set $\{a, b, c, d, e, f, g, h, i\}$ constituted a directed graph in Figure 1, the min-post label technology starts in depth-first traversal from vertex a. As shown in Figure 2, for vertex b, its minimum value of descendant vertex is 1, the last time traversing the value is 7, this constitutes interval label $L_b = [1, 7]$. According to this principle, the other vertex's interval label will be obtained. When the result is queried whether vertex b can reach the vertex i, only it needs to compare two interval labels value of the vertex a, i. Since the interval label of b is $L_b = [1, 7]$, the interval label of i is $L_b = [1, 1]$, because $(s_b = 1) \leq (s_i = 1)$ and $(e_i = 1) < (e_b = 7)$, so $b \to i$ is reachability. In a similar way, the interval label of c is $L_c = [1, 9]$, the interval label of d is $L_d = [1, 4]$, because $(s_d = 1) \leq (s_c = 1)$ and $(e_d = 4) < (e_c = 9)$, so $c \to d$ is reachability.

Based on the directed graph traversal, the interval label has abnormal phenomenon with min-post technology. Such as the label value of vertex b is $L_b = [1, 7]$,the label value of vertex c is $L_c = [1, 9]$. By Theorem 2 shows, $c \to b$ is reachability, but the actual situation is unreachability in the original graph.

3.2 The TOPSIS of Multiple Attribute Decision Making

TOPSIS is the ranking method of approximation to the ideal value.In firstly,the original matrix was normalized as normal matrix; the second, the positive and negative ideal value was calculated form normal matrix; the third, the distance between the positive and negative ideal value and scheme value was calculated; the finally, it is concluded that the proximity degree of the scheme and the positive ideal value. It is gist of performance evaluation for the evaluated object.

3.2.1 Steps of TOPSIS Analysis Method

The first step: The candidate edge information set is $P = [a_{ij}]_{m\times n}$, which meets the $1 \leq i \leq m$, $1 \leq j \leq n$, the number i is the serial number of edges and j is different attribute information of the same edge. The last, the result is decision matrix , as shown in Eq. (2).

$$P = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1i} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2i} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{ii} & \cdots & a_{in} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mi} & \cdots & a_{mn} \end{bmatrix} \tag{2}$$

The second step: Evaluation index system is divided into two major categories of cost and benefit type value. The benefit type value is the bigger and the better, the benefit type value is the smaller and the better. Because there are many difference in evaluation index, order of magnitude etc. These impact will result to some unreasonable phenomenon. So it needs to normalize the original matrix, the last, the normal matrix will be created.

$$R = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1i} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2i} & \cdots & r_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ r_{i1} & r_{i2} & \cdots & r_{ii} & \cdots & r_{in} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ r_{m1} & r_{m2} & \cdots & r_{mi} & \cdots & r_{mn} \end{bmatrix} \tag{3}$$

Some attribute values are bigger and better. For example, vehicle speed, the coefficient of road safety and so on, these attribute of traffic network are benefit type value. These class index standardization expression as shown in Eq. (4):

$$r_{ij} = \frac{a_{ij}}{\sqrt{\sum_{k=1}^{n} a_{ij}^2}}, i = 1, 2 \cdots m;\ j = 1, 2 \cdots n\,. \tag{4}$$

Some attribute values are smaller and better. For example, the distance, charges and so on, these attribute of traffic network are cost type value. These class index standardization expression as shown in Eq. (5):

$$r_{ij} = \frac{\frac{1}{a_{ij}}}{\sqrt{\sum_{k=1}^{n} \frac{1}{a_{ij}^2}}}, i = 1, 2 \cdots m;\ j = 1, 2 \cdots n\,. \tag{5}$$

In addition, interval-type data do not belong to the benefit type value and the cost type value. Such as road width, the putative interval $[a'_j, a^*_j]$,a'_j is lower limit and a^u_j is upper limit. These class index standardization expression as shown in Eq. (6):

$$r_{ij} = \begin{cases} 1 - (a'_j - a_{ij})/(a'_j - a^l_j) & a^l_j \leq a_{ij} \leq a'_j \\ 1 & a'_j \leq a_{ij} \leq a^*_j \\ 1 - (a_{ij} - a^*_j)/(a^u_j - a^*_j) & a^*_j \leq a_{ij} \leq a^u_j \\ 0 & other \end{cases} \tag{6}$$

The third step: The selection of standardized weights $W = \{\omega_1, \omega_2, \cdots, \omega_n\}$. At present there are a lot of determination method of attribute weights. If this method depends on the original data, these methods can be divided into three classes. The first class is the subjective weighting method. This method is according to the decision maker subjectively to distribute the attribute weights. The original data come from subjective judgment of experts. It has expert investigation method, analytic hierarchy process and least square method, etc. The second class is objective weighted model, the method based on objective information of decision matrix, the weight of attribute values needs to consider the variation degree and the correlation between attribute values. It doesn't depend on man's subjective judgment. Such as principal component analysis, entropy evaluation method, factor analysis, etc. Since the above two methods have advantages and disadvantages, so people also propose a third class weighting method, the comprehensive method of the combination of subjective and objective method.

According to different research purposes, using different methods to get the weights of different attributes. The weight vector of traffic information is set up by personal preferences, for example the driver wants personalized driving. On the other hand, the attribute weights of expert group information based on DELPHI method. The last, the product of weights and norm matrix are the weighted matrix. The process as shown in Eq. (7):

$$B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1i} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2i} & \cdots & b_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ b_{i1} & b_{i2} & \cdots & b_{ii} & \cdots & b_{in} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{mi} & \cdots & b_{mn} \end{bmatrix} \tag{7}$$

The fourth step: The attributes type of cost or benefit must be considered when this step obtains positive and negative ideal value of evaluation target by weighted matrix. The expression of positive and negative ideal solution are $O^+ = (b_1^+, b_2^+, \cdots, b_n^+)$, $O^- = (b_1^-, b_2^-, \cdots, b_n^-)$.

$$b_i^+ = \begin{cases} max(b_{ij}),\ j \in M^e & \text{benefit} \quad \text{type} \quad \text{value} \\ min(b_{ij}),\ j \in M^c & \text{cost} \quad \text{type} \quad \text{value} \end{cases} \tag{8}$$

The negative ideal value:

$$b_i^- = \begin{cases} min(b_{ij}),\ j \in M^e & \text{benefit} \quad \text{type} \quad \text{value} \\ max(b_{ij}),\ j \in M^c & \text{cost} \quad \text{type} \quad \text{value} \end{cases} \tag{9}$$

M^e is benefit type index, M^c is cost type index.

The fifth step: The Euclidean distance is calculated between the target value and the positive and negative ideal value.

$$S_i^* = \sqrt{\sum_{k=1}^{n}(b_{ij} - b_j^+)^2}, b = 1, 2 \ldots, n \tag{10}$$

$$S_i' = \sqrt{\sum_{k=1}^{n}(b_{ij} - b_j^-)^2}, b = 1, 2 \ldots, n \tag{11}$$

S^* is the distance between the target and the positive ideal value. S' is the distance between target and the negative ideal value.

The sixth step: The comprehensive evaluation index of each target is calculated by Eq. (12).

$$C_i^* = S_i'/(S_i' + S_i^*), i = 1, 2 \ldots, m \tag{12}$$

3.3 The Discussion of Comprehensive Evaluation Index

The Eq. (12) can tell that the value of the comprehensive evaluation index is a ratio. The distance between target value and the positive ideal value and negative ideal value decides the size of index value. C_i^* values can be divided into three cases to analyze.

(1) Assume S_i^* bigger and bigger, the S_i' value is smaller and smaller. So C_i^* values are getting smaller and smaller, until 0;

(2) Assume S_i' bigger and bigger,the S_i^* value is smaller and smaller. So C_i^* values are getting bigger and bigger,until 1;

(3) When the value of S_i' and S_i^* are equal, so the value of C_i^* is 0.5, the comprehensive evaluation index is critical state. If comprehensive evaluation index meets the $C_i^* > 0.5$, this is case of (2), it is hope result; If comprehensive evaluation index meets the $C_i^* < 0.5$,this is case of (1), it is not hope result. So the threshold value is $C_i^* = 0.5$, it can judge state of C_i^*.

4. THE DISCUSSION OF MULTIATTRIBUTE CONSTRAINT DIRECTED GRAPH

A multiconstraint and largescale directed graph, the reachability of query considers two major factors of direction and multiattribute constraint. In other words, if the interval label technology wants to traverse every vertex of the largescale directed graph, and assigned interval label to each vertex, it needs to consider both traversal sequence and multiple constraints condition in the process of traversal.The problem increases a lot of difficulty for the reachability judgment.Because of the multiple condition constraints, the reachability of original path becomes unreachability. Especially many complex factors, such as the correlation between each constraint conditions and the effect of constraint and so on, will lead to increased difficulty. If the query blindly uses the traditional interval label technology, it not only can't make judgment accurate, the result may be misleading sometime.

4.1 The Discussion of Spanning Subgraph and Interval Label

A constraint largescale directed graph, the reachability relationship mainly has two major factors of direction and constraint weights. If interval label technology needs to traverse every vertex of a largescale graph, and assign to the interval for each vertex, the establishment of interval needs to consider the traversal sequence and the constraint weights about traverse process. This problem increases much difficulty than only queried reachability of no weight directed graph. Because of the conditions constraint, the original reachable path maybe now become unreachable. If the query blindly uses interval technology, the result of query not only can't be accurately judged, but also may be misled in some cases.

From another perspective, after the constraint is considered, the directed graph can be simplified, the edge number of reachability graph will be reduced. That is to say, the graph G becomes the graph G', and the size of edges is decided by the nature of the constraint condition.

Definition 1 *For a non-empty edge set E' of graph G, the subgraph of G is consisted of the edge set E', the vertex with at least one edge associated in E' constitute the vertex set V'. The graph $G'(V', E')$ is called an edge induced subgraph of graph G, it is denoted by $G'[E']$.*

Definition 2 *The vertex set V constitute the graph G, the vertex set V' constitute the graph G'. If G' is subgraph of G, and $V = V'$, G' is called a spanning subgraph of G.*

Corollary 1 *Under constraint of the conditions, the two label values are equivalent that if the interval label technology create label values for spanning graph G' and graph G, where G' is spanning subgraph of G.*

Proof. *(1) By definition 2,$V' = V$ (V' is the vertex set of spanning graph G', V is the vertex set of graph G), so the number of vertex is same when label technology traverse the spanning G' and graph G.*

(2) Because of the constraint condition, if the weight of edge does not satisfy the constraint conditions ,the edge is equivalent to disconnect, while this disconnect edge constitutes set E", so $E' = E - E''$. By definition 1,2 the $G'(V, E')$ is a spanning graph of G. That is to say, when label technology sets up label value, it does not traverse the edge of

unsatisfied constraint condition,but it traverses all vertex of G'.

Composite (1),(2)the corollary 1 is correct.

For example, if conditional constraint value of Figure.1 is restricted as $\omega < 4$ and $S \neq S_3$, the edges e_{cd}, e_{dg}, e_{fh}, e_{hi} that do not meet the constraint condition are deleted, that is to say, the spanning graph of figure 1 is Figure 6.

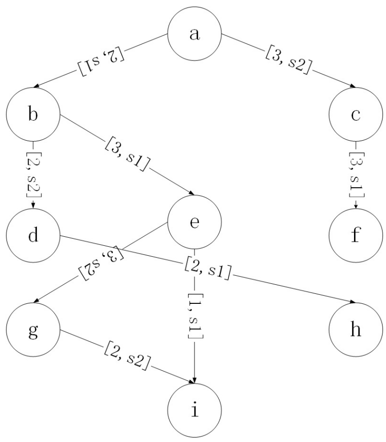

Figure 6 The spanning graph of condition constraint.

So the traversal of Figure 1 is turned into the traverse spanning graph of Figure 6, the label value is same as the traversing Figure 1, as shown in Figure 7.

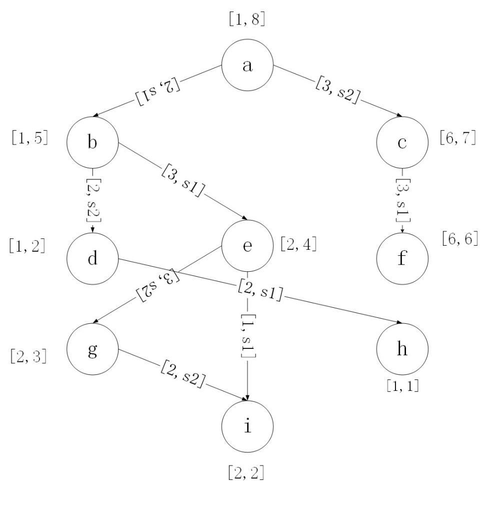

Figure 7 The interval label of spanning graph with Min-Post.

By comparing Figure 2 and Figure 7, after constraint value filtering, the anomaly value of min-post label technology reduce very rapidly. Because, the min-post technology is more effective in the tree than in directed graph. If the directed graph is simplified as the tree with constraint condition, it would have some advantages in reducing anomaly phenomena and searching the space.

4.2 The Constraints Conditions Discussion of Directed Graph

The directed graph with condition constraint mainly considers three aspects, they are constraint conditions, the constraints situation of the vertex and edge and the information completeness of directed graph. A detailed description is as follows:

(1) In multiple attribute constraint graph,the correlation and importance of attribute will cause great difficulties with reachability judgment. If the reachability judgment of every time considers the correlation and importance, it will affect the efficiency of reachability query. After this article fully considers these factors, the TOPSIS method was good choice of comprehensive decision-making and judgment for a variety of complex constraints, finally the result was a single composite scores. So the article only considers composite scores in reachability query, this will improve the efficiency of computing and the accuracy of the multiple attribute.

In another perspective, the constraints can simplify the original directed graph, it will lead to reducing the edge number of directed graph. So the graph G turn into G', and the edge number of G' is decided by the nature of multiple constraints. More rigorous conditions, the edge number of G' will become less. After considering the above situation, this paper designs three cases for the judgment of reachability:

(a) The multiple attribute value only meets condition constraints, in other words, this situation does not think about the correlation and weight of the attribute;

(b) This situation thinks about tendency in optimization judgment;

(c) Based on satisfaction the constraint conditions, this situation will find out a certain tendency of the optimization query.

(2) The attribute constraint problems, according to the different situation, the consideration factor of vertices and edges are different. For example, in the process of a large cargo transport, you only consider road traffic conditions, such as the bearing capacity of the bridge and traffic capacity of culvert, these will only consider edge constraints.But if we want to consider the processing capacity of goods transfer station and road traffic conditions at the same time, we should simultaneously consider the constraints of edge and vertex. So, the real algorithm design will consider that the vertex constraints is transformed as the constraints of the edge in these factors.

(3) In view of the constraint reachability of interval label technology, the general method is that the graph is are translated into the tree. But there is a problem, it is the loss of reachability information.So, with this problem considered, when the directed graph transformation is executed, the article tries to keep to the full of reachability information. So it can avoid the information loss of reachability with interval label technology.

5. THE CONSTRAINT REACHABILITY QUERY ALGORITHM OF DIRECTED GRAPH

In this paper, the path of the source to the destination was split into many sections with single nodes. Every outgoing edges of vertex got a comprehensive score by TOPSIS, the score was gist of the reachability judgment. So these will no longer consider multiattribute complex problem in the subsequent work, and simplify the calculation. The various attributes was fully considered in each section of path. For example, vehicle speed, safety coefficient, distance, costs, the driver's mentality tendentiousness and so on, all are the attributes of transportation network. The TOPSIS method synthesize these attributes as comprehensive evaluation index. Then the technology of circle contraction and interval label technology are used to realize the query of reachability. In addition, this paper considers the actual situation, the reachability query was divided into constraints optimized process and simple condition constraints. If user needs better cost performance and satisfied result, then they should choose optimized query. If the user just wants to know whether the path reachability in basic condition, the condition constraints query was a choice. So the overall structure of TCRQDG was shown in algorithm 1:

Algorithm 1 Overall structure of TCRQDG.

Input:
large-scale directed constraint graph;

Output:
The results of reachability;

1: If the graph need expanding,it goto step 2.Otherwise it goto step 3;
2: Algorithm 2: Create virtual vertex expand;
3: Algorithm 3: Constructing filter graph with condition;
4: If the comprehensive evaluation index was needed, it goto step 5.Otherwise it goto step 6;
5: Algorithm 4: Synthesis evaluation with condition;
6: Algorithm 5: Loop and contracting;
7: Algorithm 6: Multi-label set up;

5.1 The Pretreatment of The Original Graph

Under normal circumstances, the problems of the constraint reachability have two cases in directed graph:

(1) The edges constraints is considered in the reachability query of the graph;

(2) The reachability query simultaneously considered the constraint of edge and vertex in the graph.

For the first case, the people pay close attention to it, then the original graph does not make any preprocessing, the query handling only considered the edge constraints. But the second case is more complicated, the query not only considers the constraint of edge, but also thinks of the vertex constraint. The problem is more complicated than the first case, at the same time it is difficult to implement. So the article adopts the method of Create virtual vertex expand method: The vertex with constraint value is extended to the edge. First of all, the each vertex with constraint weights is created the corresponding virtual vertex; the Second, the each constraint weight of original vertex is assigned to the edge of the original vertex and its expanding virtual vertex. The expanding virtual vertex instead of the corresponding original vertex is regarded as the starting vertex, the terminal vertex of ingoing edge stays the same. The real example was shown in Figure.3,4, the description is shown in algorithm 2. Through the conversion of algorithm 2, the multiple factors of directed graph are simplified to the edge processing, it will improve the efficiency of query processing.

Algorithm 2 Create virtual vertex expand method.

Input:
large-scale directed constraint graph;

Output:
The expanding graph with creating virtual vertex;

1: The algorithm input directed graph G that the vertices and edges have constrain weight;
2: If directed graph was forest,firstly,the algorithm sets up a virtual root vertex, the weight value of virtual root vertex is 0;secondly,the algorithm create each edge for the virtual root vertex and the root vertex of each tree,the edge constraint weights value is 0;
3: Starting from the root vertex traversal;
4: If the starting vertex is the virtual root vertex, then the root vertex do not be made any processing,the child vertex is directly traversed.The algorithm goto step 6,otherwise,it goto the step 5;
5: If the graph is the original graph, the traverse will directly start from the root vertex v;
6: For each vertex v, the algorithm creates a corresponding virtual vertex v', and connecting vertex v and v', the weight of edge vv' is the weight of vertex v;
7: The starting vertex v of out edge is modified v', the terminal vertex of income edge stay the same;
8: Repeat step 6 and step 7,until all vertex of directed graph G are traversed;
9: Output expanding directed graph G';

5.2 According to Constraint Conditions Filtering the Expanding Directed Graph

The main element of condition graph is the constraint condition, the constraint situation thinks of vertices and edges, information completeness of directed graph, etc. As constraint problem, if the edge weight doesn't meet the constraint condition, it will ultimately be deleted by the reachability query of graph. If the filter is put in later, it will increase the consumption of time and space, while the consumption has not contributed to the query. But if the filter is put in the front, it will have two benefits:

(1) this method will reduce the consumption of time and space, improve the efficiency of the query;

(2) The filter will help to reduce the number of multi-interval label.

The solution of this article is to simplify the directed graph according to the constraint conditions. Specific approach is described below, when the each edge is traversed, the first, the edge weight is judged whether the value can satisfy the constraint condition. If the edge weight can satisfy the constraint condition, this edge retention. If the edge weight does not satisfy the constraint conditions, then the edges and their weights are deleted, then the edge and its weights are removed from expanding directed graph. It has a realistic basis, for example, the large equipment transportation passes by a bridge, if the bridge allows maximum weight that is 100 tons, and if the large equipment weight is 150 tons. The bridge is not suitable for the large equipment, the line must be rerouted. The simplifying directed graph by conditions reduces a large number of error message and exception information, this solution will effectively promote the query efficiency of interval label technology. The specific process as shown in algorithm 2, because there are three constraint conditions, the algorithm uses constraint condition $\omega_{edge} \leq \omega_c$ to illustrate. The rest of the two methods can be used by simple adjustment.

Algorithm 3 Constructing filter graph with condition.

Input:
Expanding directed graph;

Output:
Simplifying directed graph;

1: The algorithm inputs the expanding directed graph;
2: Starting from the root vertex, the algorithm traverses each edge;
3: If edge attribute value satisfies constraint condition, the edge information and constraint conditions are reserved in graph, it goto step 5. Otherwise the algorithm will enter the step 4;
4: Because the weight of edge doesn't meet the constraint condition, the edge and weight are deleted;
5: If all edges of graph are traversed, the algorithm will enter the step 6, otherwise, it will goto step 2;
6: Output simplifying directed graph;

Algorithm 3 simply iterates through each edge, so the algorithm complexity is $O(m)$, the m is the number of edges of graph.

For example, if the constraint condition is $\omega < 4$ and $s \neq s_3$, because the edges $e_{cd}, e_{dg}, e_{fh}, e_{hi}$, does not satisfy constraint condition, these edge will are deleted. The spanning graph as shown in Figure.6.

5.3 Calculating the Comprehensive Evaluation Index

This article evaluates various constraint conditions of the directed graph by TOPSIS. The method produces comprehensive score for every edge. The size of the score reflects the merits of the edge scheme. Because the different conditions have many value type, for example, cost and benefit type value, interval number and single number type value, etc. These different conditions will affect the calculation of comprehensive score. The article uses the theory of TOPSIS to deal with these data, its purpose is to reduce the effect of different data type. Finally, the comprehensive evaluation index is calculated by normalization matrix. Based on the discussion of comprehensive evaluation index, the threshold is $t = 0.5$. The $C^* \geq 0.5$ is desirable result,and the $C^* < 0.5$ is not good value. The directed graph can be reduced by threshold filter.

Algorithm 4 Comprehensive evaluation with attribute.

Input:
The simplification directed graph;

Output:
The comprehensive evaluation index of graph;

1: The algorithm determines the path information set;
2: The path information set is normalized by different attribute;
3: It calculates the weight judgment matrix with the normalization weights;
4: According to weight judgment matrix, this method obtains the positive and negative ideal solution;
5: The Euclidean distance is calculated between each target and the ideal value;
6: Each Euclidean distance is used to calculate the target of comprehensive evaluation index;
7: If the directed graph needs to optimize with comprehensive evaluation index, then it executes the step 8,otherwise quit;
8: The comprehensive evaluation index are filtered by the threshold;

The comprehensive evaluation index is calculated by TOPSIS analysis method, the multiple attribute graph will be reduced only a single index value. So according to the size of the comprehensive evaluation index, it can judge an optimal choice of the outgoing edges.

5.4 The Circle Judgment in The Spanning Graph

The general algorithm is to break here, its purpose is to translate the graph into a directed acyclic graph, but broken circle has loss of the graph information. In other words the directed acyclic graph is an approximate representation of the original graph, it can't completely show original graph information. This article method is circle marking. Firstly, the method is looking for circles, after that each circle is respectively marked with new label. And vertices of each circle are contracted as a virtual vertex, the new label value of virtual vertex is the same as corresponding circle label value. The ingoing edge and outgoing edge of circle vertices are connected to this virtual vertex, and remain the same. Then the label technology creates interval label in contraction graph. Finally the contraction graph returns to the original spanning graph, and the vertex interval label of the same circle is marked as the same value, vertex interval label value is the value of corresponding virtual vertex. As shown in Figure 8, 9, 10, 11.

The Figure 11 shows that the interval label value of vertex f and h are consistent, but this does not affect the original reachability of figure 8, because the information of original graph is preserved by this method, so there is not any artificial information missing. As shown in algorithm 5.

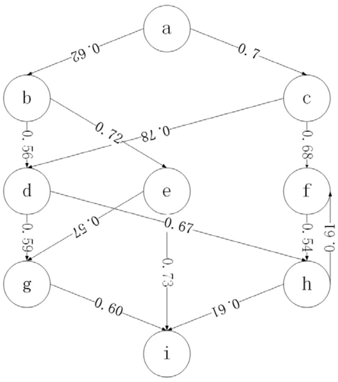

Figure 8 The directed graph of having a circle.

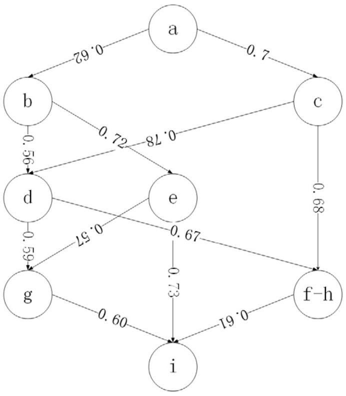

Figure 9 The directed graph with circle shrinkage.

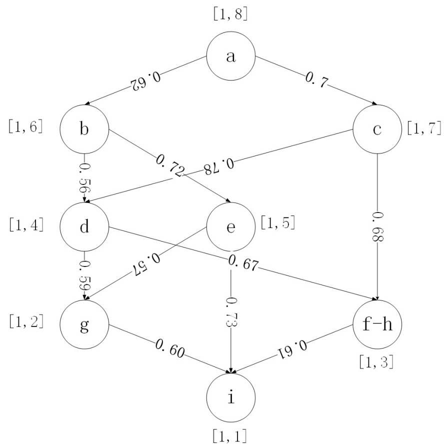

Figure 10 The directed graph with Min-Post label.

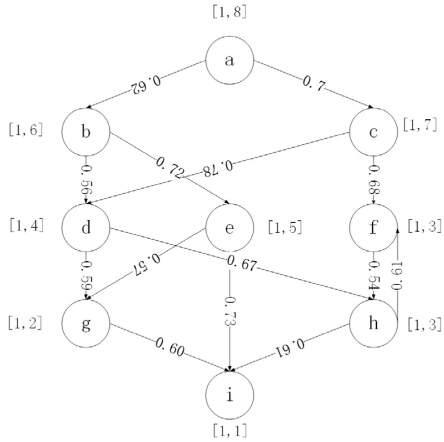

Figure 11 The directed graph with the Min-Post label after recovering the circle.

Algorithm 5 Loop and contracting.

Input:

The comprehensive evaluation index of graph;

Output:

The contraction directed graph;

1: The algorithm input the comprehensive evaluation index of graph;
2: Starting from the root vertex,this algorithm traverses each edge of the simplifying directed graph, and the root vertex is put into the stack;
3: If the current traversal vertex has not adjacent child vertices that these vertices is not in the stack, this shows that there are not a circle, the adjacent child vertex will be traversed;
4: If adjacent vertex of current traversal vertex is in the stack, this shows that there are a circle, then all vertex of the circle are made into the same label;
5: The same label vertex of circle is contracted as a virtual vertex,the outgoing edge and ingoing edge of all circle vertex are changed into the virtual vertex;
6: When all vertices of simplifying directed graph are traversed,this algorithm will end;

5.5 Creating the Multi-interval Label

There are two main forms about the interval label technology, the first is pre-post technology, the second is min-post technology, the other technologies are the extension of the two technologies. After directed graph traversal, pre-post label technology can quickly query the reachability of arbitrarily two vertices, but this technology has the phenomenon of information loss in the directed graph. Such as shown in Figure 1, the vertex c and d are reachability, but they are unreachability in Figure 5.

Min-post interval label is $l_\mu = [s_\mu, e_\mu]$, the value s_μ is the minimum value traversal of child vertex of vertex u. It is composed of two cases, firstly, if child vertex is not minimum vertex, $s_\mu = min\{s_\nu | \nu \in children(\mu)\}$; secondly, if child vertex is the minimum vertex, at this time $s_\mu = e_\mu$, the sequence number value is last time traversing value of the vertex u. The technology of min-post interval label will produce some exceptional information, this abnormal information is mainly that the interval label contain some vertices reachability information, but these vertices are unreachability in the original graph. For this kind of situation, as discussed by [11], they used multi-interval label technology to solve the problem, the detailed description is that there are $L_\mu = L_\mu^1, L_\mu^2, \cdots, L_\mu^n$, then $L_\mu^i (1 \le i \le d)$ is the interval label of vertex u of the i times random traversal DAG. If and only if for all $i \in [1, d]$,there are $L_\nu^i \subseteq L_\nu^i$, the $L_\nu \subseteq L_\nu$ is established, these show that the vertex μ and ν is reachability. If an interval label exist in $L_\nu^i \subsetneq L_\nu^i$, this prove that the vertex μ and ν is unreachability. So multiinterval label technology avoid information missing of the pre-post technology. But the technology is not concerned about the problem of constraints and information integrity, so this article uses the min-post label technology to create label index for the filter directed graph. When the interval label is set up in each time, according to the third step, the virtual vertex revert to vertices of the correspondence circle, but these vertices have the same label.

After the Figure 1 is dealt with by min-post label technology, as shown in Figure 2, there are the phenomenon of abnormal information, such as the label value of b is $L_b = [1, 6]$, The label value of c is $L_c = [1, 8]$. By the definition of interval label, $c \rightarrow b$ can reach, but in fact they can't reach on original graph.

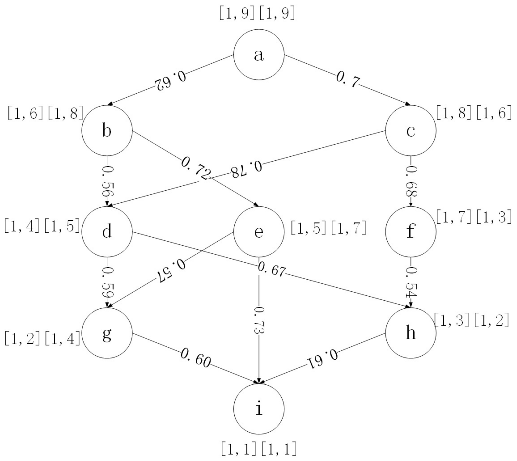

Figure 12 The multi-interval label of Min-Post directed graph.

As shown in Figure 12, the label of b are respectively $L_b^1 = [1, 6]$ and $L_b^2 = [1, 8]$, the label of c are respectively $L_C^2 = [1, 8]$ and $L_C^1 = [1, 6]$. Although ,$L_b^1 \subset L_C^1$ but $L_b^1 \not\subset L_C^1$, the $c \rightarrow b$ was wrong. So this method overcomes the abnormal phenomenon.

5.6 The Query of Constraint Reachability

The query of constraint reachability considers the actual situation in a directed graph, for example social networks, the query is that whether the two people are classmates, then everyone is represented as a vertex. If the query vertices of two people can reach, this will show that the two people are classmates. Otherwise, they are not classmates. For other situation, the query also wants to know that whether each relationship between two people is classmate in the people list. Under this condition, the query efficiency is very low with a single vertex pair. At this time, the query algorithm can consider the method of batch, all case of the query one-time are imported, and the result can be satisfying and dissatisfying the constraint condition respectively, so this will improve the efficiency of the work.

Algorithm 6 Multi-label set up.

Input:
The contraction directed graph;

Output:
The directed graph of multi-interval label;

1: Starting from the root vertex, the algorithm traverses each edge of the contracting directed graph;
2: If the vertex of traversing edge has successor, it will continue to traverse the following vertices. When the vertex $v_{\text{min-children}}$ has no posterity, then $s_v = e_v$, $v_{\text{min-children}}$ is the smallest subsequent vertices, the label is $L^i[s_v, e_v]$;
3: From the vertex $v_{\text{min-children}}$ backtracking, if the current vertex of the backtracking has no other posterity vertex t, its label is $L^i[s_v, e_v + 1]$,if there are other posterity vertices, the label is $L^i[s_v, e_{\text{last}}]$ .The s_v is the smallest number of posterity vertex, and e_{last} is the last number of backtracking;
4: If the process of backtracking returns to the root vertex and the root vertex has no other posterity vertices without the traverse, the creation of interval label is ended;
5: Starting from the root vertex, the algorithm traverses each edge of contraction directed graph again. But the starting edge is different from previous outgoing edges of root vertex, and it goto the step 2;
6: The method check circle mark at this time, if the mark is empty, that the graph has no circle, then it goto the step 8. If the mark is not null, then it goto step 7;
7: According to the circle mark, the virtual vertex reverts to former vertex of circle contraction, and the label value of virtual vertex is assigned to all vertex in the circle;
8: If there are no other circles, then it goto step 9. If there are other circles, then it goto step 7;
9: The algorithm end;

6. EXPERIMENTAL ANALYSIS

The operating system of the experiment is windows 7 service pack 1,CPU is intel core i3-2310M, the memory is 4G, the experimental data come from San Francisco Road Network [6] and Beijing road network text data [5].

The two data sets have been optimized through the following ways: Firstly, the N vertices are generated by each intersection information, afterwards the M edges are generated by connecting between each intersection; Secondly,the constraint conditions include road width, speed, road safety, distance, etc. Such as the road width is interval data, speed and road safety are benefit type data, distance and charge are cost type data. The experiment integrates these constraint conditions to produce data. At last, the San Francisco Road Network has 174955 vertices and 223001 edges, the Beijing road network has 171504 vertices and 433251 edges.

6.1 The Data Preprocessing of Modified TOPSIS

Each edge of road network was evaluated by modified TOPSIS. Secondly,the road network is screened with the threshold $t \geq 0.5$ of comprehensive scores. After screening, the retained edges have higher score in multiattribute conditions graph, it will contribute to make the better judgment of constrain reachability. So it has practical guiding significance. Finally, the compared result of the new number and the original number of edges were shown in Figure 13.

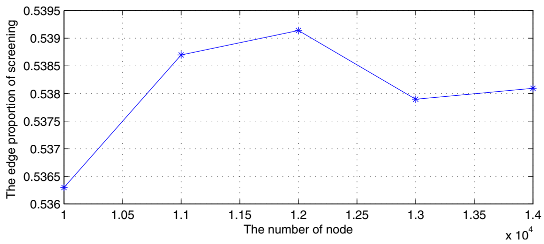

Figure 13 The ratio of the new number and the original number of edges.

The Standard Deviation of screening ratio is 0.001083, the value range of ratio is 0.002839. These data reflect the discrete degree of screening that it is small, the reality data deviate few from the average value. The mean is 0.538, the median is 0.5381. The mean and the median are almost unanimously. These values reflect the screening value is reasonable.In addition, the number of edge is almost half than screening before. So the workload is reduced much, the result will improve the working efficiency.

Every process of decision making needs TOPSIS method for comprehensive evaluation, so the TOPSIS will affect the query time. This article constantly adds nodes and edges in experiment, its purpose is to analyze the influence of data scale. The Detailed was show in Figure 14.

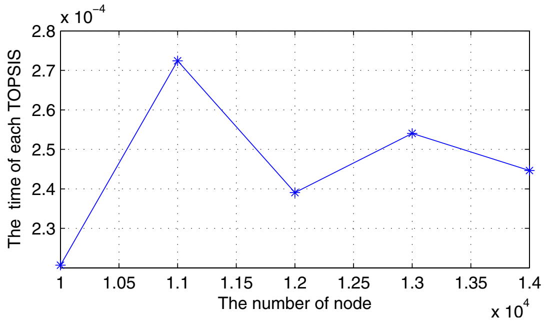

Figure 14 The time of each TOPSIS.

After the experiment, the process of TOPSIS is microsecond time, the standard deviation of TOPSIS time is 0.00001907 second, the value range of time is 0.00005175 second. The each decision time is a little different, the cause may be the number of edges, but the overall difference only is a few microseconds. In addition, the average is about 0.0002462 seconds, the median is 0.0002446 seconds, thus it can be seen that the time of single TOPSIS has not obvious change with the increase of data scale.

6.2 The Comparison of TCRQDG Algorithm and Classical Floyd Algorithm

Floyd algorithm is a classical algorithm of the shortest path, it can query the shortest distance of all vertex pairs. Of course, it can also query the reachability of vertices, and TCRQDG algorithm can also query a batch of the reachability of vertices. This article uses the same vertices and edges to compare, especially when the data volume is very larger, obviously it will show the difference efficiency between two kinds algorithms.

In the experimental process, when the data sets have 500 to 1000 vertices, the time consumption of Floyd algorithm are between 50580.9 milliseconds to 196251 milliseconds, while the total time consumption of TCRQDG algorithm is between 31 milliseconds to 67 milliseconds, where the time of creating index is between 9 milliseconds to 12 milliseconds, the query time is between 22 milliseconds to 54 milliseconds. This experiment makes base 10 logarithmic transformation of the original data, so it can easily compare with the Floyd algorithm and TCRQDG algorithm in a figure.

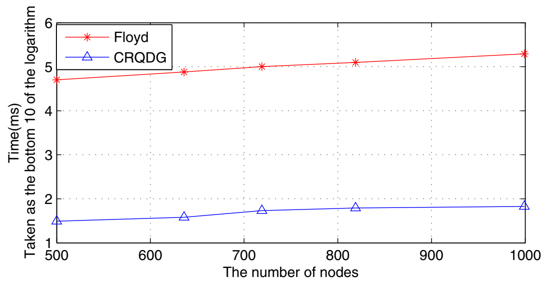

Figure 15 The comparison of the query time on Floyd and TCRQDG.

The Figure 15 shows that the time consumption of Floyd and TCRQDG is not an order of magnitude. In addition, with the increase of the data amount, the time consumption has the trend of increasing. But the growth trend of Floyd algorithm is quicker than TCRQDG algorithm, it is also verified by experiments. The reason is that the size of experiment data is chose between 500 and 1000 vertices.

6.3 The Comparative Analysis of Creating the Label Under Different Conditions

Because TCRQDG algorithm is the reachability query of condition constraint, so different constraint conditions may lead to the different time consumption of indexing and querying, for comparative analysis, this paper obtained data for 11k to 16k vertex number and respectively set conditions 1 and 2.However,the conditions 1 is $\omega_1 \in [\omega_{\min 1}, \omega_{\max 1}]$, the conditions 2 is $\omega_2 \in [\omega_{\min 2}, \omega_{\max 2}]$. In addition, when setting the conditions, the condition 2 is more strict than condition 1,$[\omega_{\min 1}, \omega_{\max 2}] \subset [\omega_{\min 1}, \omega_{\max 1}]$, the $\omega_{\max 1}$ is twice than the $\omega_{\max 2}$.

Under different conditions, TCRQDG algorithm may be different in creating index time. If the condition is strict, the filtering process will delete more edge of the dissatisfying condition by the algorithm 3. So the filtering will reduce a lot of work in a subsequent circle mark, creating labels and query process, in order to improve the efficiency of reachability query.

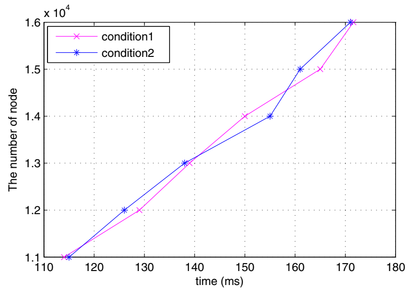

Figure 16 The comparison of creating time under different conditions.

Through the comparative analysis of Figure 16, the number of vertices in different conditions and the time of creating label is not the same. Because the conditions 2 is more strict than conditions 1. So it can be observed from the Figure 16:

(1) when the condition 2 creates label, the time consumption is less than condition 1 in each number set, which the conditions filtering process played a good role. On the one hand, this method reduced the unnecessary work, on the other hand it also meets the requirements of the constraint;

(2) Whatever the conditions, with the increasing of the amount of data, the time of creating labels also increases, it also accords with scientific rules.

6.4 The Comparative Analysis of Query Under Different Condition

The same amount of data, through the filtering of different constraint conditions, the time of creating index is different, the number of edge will also vary as filtering of the constraint condition, then the query time of batch is different.

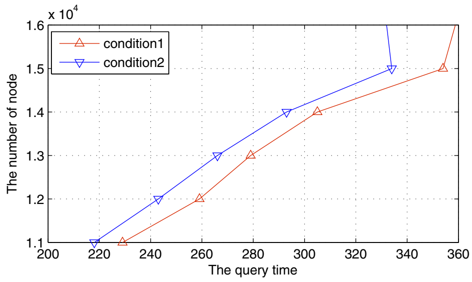

Figure 17 The comparison of query time under different condition.

Through Figure 17, the reachability query time is different under two kinds of condition. The query time of condition 1 is more than the condition 2, the reason is that the conditions 2 is more strict than condition 1. After filtering the condition, the amount of effective edge will decrease, at the same time the query time will reduce, in order to improve the efficiency of the query.

As shown of Figure 18, the total using time under different conditions is different, the total time of condition 1 is much more than condition 2, this is also related to the filtering conditions. If the filtering conditions is more strict,the amount of effective edge is less, the time of creating index and query will be reduced.

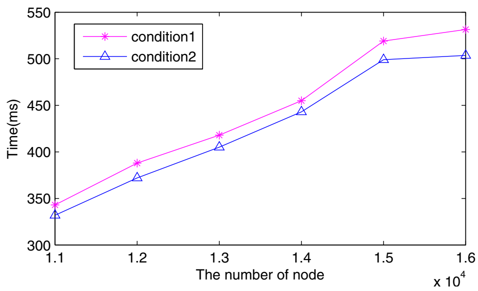

Figure 18 The comparison of the total time in different conditions.

6.5 The Comparative Analysis of Effective Query Vertex for TCRQDG and Classical Algorithm

Because TCRQDG algorithm is the reachability query of condition constraint. After being filtered, the more strict constraint conditions will lead to the less efficient vertex number. Some classic algorithms query all the amount of data vertex. For the effective vertex number of the actual query, TCRQDG algorithm is reduced a lot. So it improves the efficiency of the query.

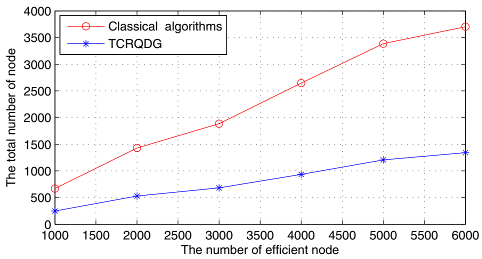

Figure 19 The comparison of the effective vertex number.

As shown of Figure 19, under the same limit condition, the effective vertex number of queries on some classic algorithm is much larger than the TCRQCD algorithm. With the increase of the vertex number, the gap is more and more obvious. So the condition filter of this article does great help to improve the efficiency of the query.

7. CONCLUSION

This paper introduces a query algorithm of TCRQCD, it can effectively solve the new problem that the reachability query requires multiple attribute constraints in largescale directed graph. The method translates the multiple attribute values as comprehensive evaluation index, this will simplify condition of the directed graph. The follow step will accelerate the query by circle contracting, release and interval label technology. So, the TCRQDG algorithm will also provide effectively technical support for network transport, logistics transportation, social networks, and software testing. At the same time it also creates a solid foundation for the query of parallel technology, the dynamic and uncertainty graph.

8. ACKNOWLEDGMENTS

This work supported by the National Science Foundation for Distinguished Young Scholars of China (Grant No. 61225010) and Lanzhou science and technology plan projects (Grant No. 2014-1-89 and Grant No. 2014-2-35).

REFERENCES

1. S. A.D.Zhu, W.Q.Lin and X.K.Xiao(2014). Reachability queries on large dynamic graphs: A total order approach. In *Proceedings of Special Interest Group on Management Of Data(SIGMOD'2014)*, pages 1323–1334, Snowbird,USA. ACM.
2. C.L.Hwang and K.Yoon(1981). *Multiple Attribute Decision Making: Methods and Applications*. Springer-Verlag, New York.
3. N. L. C.Peng, Y.S.Tan. Approximation algorithms for inner-node weighted minimum spanning trees. *Computer System Science and Engineering(CSSE)*, 24(3):189–195.
4. C.T.Chen(2000). Extension of the topsis for group decision making under fuzzy environment. *Fuzzy Sets and Systems(FSS)*, pages 1–9.
5. Datatang.com(2013). Beijing new road network.
6. Datatang.com(2013). San francisco road network.
7. G.Fishman(1986). A comparison of four monte carlo methods for estimating the probability of s-t connectedness. *IEEE Transactions on Reliability(TR)*, 35(2):145–155.
8. C. G.Kim and K.P.Yoon(1997). Identifying investment opportunities for advanced manufacturing system with comparative-integrated performanc measurement. *Int J of Production Economics(PE)*, 50(1):23–33.
9. J. P. H.Wang, H.He and J.X.Yu(2008). Dual labeling: Answering graph reachability queries in constant time. In *Proceedings of IEEE International Conference on Data Engineering(ICDE '2008)*, Atlanta,USA. IEEE.
10. V. H.Yildirim and M.J.Zaki. Dagger: A scalable index for reachability queries in large dynamic graphs.
11. V. H.Yildirim and M.J.Zaki(2012). Grail: a scalable index for reachability queries in very large graphs. *The VLDB Journal(VLDB)*, 21(1):509–534.
12. I.B.Dhia(2012). Access control in social networks : A reachability-based approach. In *Proceedings of International Conference on Extending Database Technology(EDBT '2012)*, pages 227–232, Berlin,Germany. EDBT/ICDT.
13. J.Cai and C.K.Poon(2010). Path-hop efficiently indexing large graphs for reachability queries. In *Proceedings of ACM International Conference on Information and Knowledge Management(CIKM '2010)*, pages 119–128, Toronto,Canada. ACM.
14. H. H. J.Cheng, Z.C.Shang and J.X.Yu(2012). Kreach:who is in your small world. *The VLDB Journal(VLDB)*, 5(11):1292–1303.
15. S. J.S.Lee. An improved path finding under multiple exits with soa in a double-layered manet. *Computer System Science and Engineering(CSSE)*, 27(2):149–165.
16. K.D.Saikat and H.Jamil(2010). A hierarchical approach to reachability query answering in very large graph databases. In *Proceedings of ACM International Conference on Information and Knowledge Management(CIKM '2010)*, pages 1377–1380, Toronto,Canada. ACM.
17. J. L. Y. K.Xu, L.Zou and D.Y.Zhao(2011). Answering label-constraint reachability in large graphs. In *Proceedings of ACM International Conference on Information and Knowledge Management(CIKM '2011)*, pages 1595–1600, Glasgow,UK. ACM.
18. M.A.Abo-Sinna and A.H.Amer(2005). Extensions of topsis for multi-objective large-scale nonlinear programming problems. *Applied Mathematics and Computation(AMC)*, 162(1):243–256.
19. L. M.Qiao, H.Cheng and J.X.Yu(2013). Computing weight constraint reachability in large networks. *The VLDB Journal(VLDB)*, 22(3):275–294.
20. P.F.Dietz(1982). Maintaining order in a linked list. In *Proceedings of ACM Symposium on Theory of Computing(STOC '1982)*, pages 122–127, San Francisco,USA. ACM.
21. A. R.Agrawal and H.V.Jagadish(1989). Efficient management of transitive relationships in large data and knowledge bases. In *Proceedings of Special Interest Group on Management Of Data(SIGMOD '1989)*, pages 253–262. ACM, 1989.
22. N. R.Jin, Y.Xiang and H.Wang(2008). Efficiently answering reachability queries on very large directed graphs. In *Proceedings of Special Interest Group on Management Of Data(SIGMOD '2008)*, pages 595–608, Vancouver,BC,Canada. ACM.
23. S. R.Jin, N.Ruan and J.Y.Xu(2014). Scarab: Scaling reachability computation on large graphs. In *Proceedings of Special Interest Group on Management Of Data(SIGMOD '2012)*, pages 169–180, Scottsdale,USA,. ACM.
24. R.M.Jin and G.Wang(2013). Simple, fast, and scalable reachability oracle. In *Proceedings of VLDB Endowment(VLDB '2013)*, pages 1978–1989. VLDB.
25. H. N. R.M.Jin, H.Hong and Y.Xiang(2010). Computing label-constraint reachability in graph databases. In *Proceedings of Special Interest Group on Management Of Data(SIGMOD '2010)*, pages 123–134, Indianapolis,USA. ACM.
26. Y. R.M.Jin, N.Ruan and H. X. Wang(2011). Path-tree: An efficient reachability indexing scheme for large directed graphs. *ACM Transactions on Database Systems(TDS)*, 36(1).
27. A. R.Sehenkel and G.Weikum(2005). Efficient creation and incremental maintenance of the hopi index for complex xml document collections. In *Proceedings of IEEE International Conference on Data Engineering(ICDE '2005)*, Tokyo,Japan. IEEE.
28. A. R.Sehenkel and G.Wezkum(2004). Hopi:an efficient connection index for complex xml document collections. In *Proceedings of International Conference on Extending Database Technology(EDBT '2004)*, pages 1978–1989, Heraklion,Greece. EDBT.
29. S.Trissl and U.Leser(2007). Fast and practical indexing and querying of very large graphs. In *Proceedings of Special Interest Group on Management Of Data(SIGMOD '2007)*, pages 595–608, Beijing,China, 2007. ACM.
30. V. V.P Agrawal and S. Gupta(1991). Computer aided robot selection: The multiple attribute decision making approach. *Int J of Production Research(PR)*, 29(8):1629–164.
31. Y.Chen and Y.Chen(2008). An efficient algorithm for answering graph reachability queries. In *Proceedings of IEEE International Conference on Data Engineering(ICDE '2008)*, Cancun,Mexico. IEEE.
32. H. C. . Y.X.Tang, Y.Q.Zhang. A parallel shortest path algorithm based on graph-partitioning and iterative correcting. *Computer System Science and Engineering(CSSE)*, 24(5):351–360.
33. Y.Yuan and G.R.Wang(2010). Answering probabilistic reachability queries over uncertain graphs. *Chinese Journal of Computers(CJC)*, 33(8):1378–1386.