A Multi-dimension Weighted Graph-Based Path Planning with Avoiding Hotspots

Shuo Jiang[1,3](✉), Zhiyong Feng[1,3], Xiaowang Zhang[2,3], Xin Wang[2,3], and Guozheng Rao[2,3]

[1] School of Computer Software, Tianjin University, Tianjin 300350, China
{jiangshuo, zyfeng}@tju.edu.cn
[2] School of Computer Science and Technology, Tianjin University, Tianjin 300350, China
{xiaowangzhang, wangx, rgz}@tju.edu.cn
[3] Tianjin Key Laboratory of Cognitive Computing and Application, Tianjin 300350, China

Abstract. With the development of industrialization rapidly, vehicles have become an important part of people's life. However, transportation system is becoming more and more complicated. The core problem of the complicated transportation system is how to avoid hotspots. In this paper, we present a graph model based on a multi-dimension weighted graph for path planning with avoiding hotspots. Firstly, we extend one-dimension weighted graphs to multi-dimension weighted graphs where multi-dimension weights are used to characterize more features of transportation. Secondly, we develop a framework equipped with many aggregate functions for transforming multi-dimension weighted graphs into one-dimension weighted graphs in order to converse the path planning of multi-dimension weighted graphs into the shortest path problem of one-dimension weighted graphs. Finally, we implement our proposed framework and evaluate our system in some necessary practical examples. The experiment shows that our approach can provide "optimal" paths under the consideration of avoiding hotspots.

Keywords: Path planning · Avoiding hotspots · Multi-dimension weighted graph · Shortest path problem

1 Introduction

1.1 Path Planning

Path planning is a sequential algorithm based on existing nodes, edges and weights according to a certain method. These nodes, edges and weights are data in graph model, which can represent different things in different situations, such as obstacles, hotspots and so on. Path planning technology has been applied extensively in many domains since it was proposed [1]. There are plenty of applications in frontier domains: route planning of unmanned aerial vehicles, robot path planning and space path planning of rocket launch. This technology not only speeds up the progress in frontier domains, but also becomes an integral part of our daily life [12]. For example, GPS navigation helps

H. Chen et al. (Eds.): CCKS 2016, CCIS 650, pp. 15–26, 2016.
DOI: 10.1007/978-981-10-3168-7_2

us plan path while we are driving. The application of the technology in business and management domain is logistics, that is, resources dispatch in a reasonable way. Generally speaking, the problems which can be translated into graph models can be translated into nodes, edges and weights, and we can use path planning to solve them [2].

1.2 Avoiding Hotspots

Avoiding hotspots is a way to use existing data to deal with hotspots, thus making the overall planning immune to the effects of hotspots. Avoiding hotspots is not eliminating hotspots. What we avoid is the effect and damage caused by hotspots. That is to reduce the occurrence probability of hotspots. Therefore, avoiding hotspots can be used in path planning, especially vehicle routing problem.

Vehicle routing problem (VRP) was proposed firstly in 1959. It means a distribution center which provides different numbers of cargoes to a certain number of customers in a city or an area. The most important part is to plan an optimal path, the goal of which is to reach the highest economic benefit under the precondition that the requirements of customers must be met. There are some requirements in path planning such as the shortest path, the shortest time or the least oil consumption.

The loss is not only in economy, but also in environment. Fuel consumption causes air pollution. Traffic jam happens frequently in our daily life. The probability of traffic accidents is still rising. Thus, it is of great importance to avoid hotspots.

1.3 Related Works

Since the weighted graph was proposed, plenty of applications have been generated. After researching the current situation of path query of weighted graph, we reach some conclusions as follows: [3] proposes a new optimal route search model for public transit based on directed weighted graph. This model cannot only allow users to set their ideal maximum walking distance, but also meet the requirements of personalized query by using the flexible weighting graph method, especially the strong expression ability for multi-object query. Fire brigades are always required to reach the field of fire in the shortest time. Therefore, selecting the appropriate path can effectively reduce the loss of casualties and property. [4] establishes a model of multi-stage weighted directed graph aimed at this problem. Multi-stage weighted directed graph is a common graph, which can translate a lot of practical matters, such as transportation, engineering, and management, into the shortest path problem. [5] establishes a general weighted graph for transportation. It is a math model which combines network analysis and linear programming theory. This model solves the practical matter caused by network complexity, path diversity and load capacity.

There is a query of weighted graph based on regular expression in [6]. The author characterizes the query of weighted graph and proposes the algorithm. This query can be embedded effectively in XML query language. [7] proposes a query of weighted regular expression. This query can allow users to define priority of weight and be connected naturally with link information of quantitative database. The authors also propose a distribution algorithm to calculate this query. This algorithm can also solve the multi-source shortest path problem in case that we do not know the complete graph. In order to query and analyze graph database by the method of aggregate function and order, [8] extends a previous graph query language. This language can support query probability graph in that way. [13] presents a SPARQL-based querying language named pSPARQL for probabilistic RDF graphs.

We can see that there is a preliminary study on weighted graph from the research above. Not only the language is of normalization and flexibility, but also the algorithm for the weighted graph is of efficiency. However, there is still something that can be extended in the weighted graph to make it more effective than previous ones. We can see that the previous studies on weighted graph only focus on one-dimension weight, while the traffic environment is complex in the real world, which means one-dimension weight cannot describe the information exactly. As we know, many cases are composed of factors influenced with each other. Therefore, it is unreasonable to calculate weighted graph with one-dimension weight.

We can also see from the research on related works that the weighted graph systems can not meet all the requirements of customers. The problems we face every day are all in characteristics for ourselves, as a result, the previous one-dimension weighted graph models can only solve a little part of the problem. Therefore, there is a lack of model or an aggregate function which users can define for their own demands to solve problems.

1.4 Overview

The overview of the paper is as follows: we focus on complex traffic environment with multi-dimension weighted graph; then we establish a model according to the specific circumstances and requirements; and we define an aggregate function which can translate multi-dimension weighted graph into one-dimension weighted graph; finally we use Dijkstra algorithm, a classical path planning algorithm, to solve the problem and propose a good plan.

The overall structure of this paper is as follows: Sect. 1 mainly introduces the related work, the lack of them, and how we deal with the lack through our innovations of this paper; Sect. 2 introduces related concept of graph; Sect. 3 introduces the multi-dimension weighted graph and the aggregate function; Sect. 4 simulates a specific case, then we show how to use our method to solve it; Sect. 5 shows the whole framework, related experience and the efficiency of this framework; Sect. 6 concludes this paper and the future work.

2 Graph

2.1 Basic Definitions

Graph is a math object to describe the relationship among objects. Assume graph G is an ordered two tuple (V, E), and V represents a set of vertices, then we can use $V(G)$ to represent a set of nodes; E represents a set of edges. Similarly, we can use $E(G)$ to represent a set of edges. Note that E and V do not intersect. The elements of E are all two tuple, which are noted by (x, y), and $x, y \in V$ [9].

Path is a sequence from one node to another. For example, assume that a path P is v_0, $(e_1, v_1, e_2, v_2, \ldots, e_k, v_k)$ and the length of this path is k. There is a pair (v_{i-1}, v_i), which is an edge from v_{i-1} to v_i. If the starting node and the ending node is the same, then we say this path is close. Otherwise, we say it is open.

Graphic model is a structure model, whose function is to describe a system. Constituted by nodes and edges, it can represent everything in the real world, so it can be used to describe the relationship among all objects. Therefore, a graphic model is a good tool for modeling, and it proposes a good way to deal with complex systems.

2.2 Directed Graph

Directed graph is a subclass of graph. Every edge is directed in directed graph. Directed graph is an ordered pair. Assume there is a directed graph D, and the ordered pair is (V, E). Then V is a nonempty set constituted by nodes of D. The elements in V are vertexes. E is a set of edges of D constituted by V.

Every element in the edge of directed graph is an ordered pair. Assume that an ordered pair is $<u, v>$ in directed graph D, which we say is a directed edge. u represents the starting node of the edge, while v represents the ending node of the edge. Therefore, $<u_i, v_i>$ and $<v_i, u_i>$ represent two different edges.

2.3 Undirected Graph

Undirected graph is a subclass of graph, however, different from directed graph. Every edge in undirected graph is undirected, and it is represented by unordered pair. Assume that an undirected graph $G = \ <V, E>$. V is a nonempty set constituted by nodes. E is a set of unordered two tuple constituted by the elements in V, and it is a set of edges. Intuitively, if all edges in a graph are undirected, then the graph is undirected. Unordered pair is usually noted by round brackets. Contrary to directed graph, there are no starting node s and ending node s in undirected graph. That is, the two unordered pairs (v_i, v_j) and (v_j, v_i) present the same edge.

2.4 Weighted Graph

Weighted graph is also a subclass of graph, but it is different from the previous two graphs for the reason that every edge in weighted graph is assigned with a value. This value is the weight of this edge. Weight can take a certain value to represent other objects, such as cost, probability and so on. Broadly speaking, weight in the weighted graph is usually single.

Assume there is a weighted graph $G = \ <V, E, W>$. V is a nonempty set constituted by nodes, then it is a set of nodes of G. E is a set of two tuple constituted by the elements in V, then it is a set of edges. W represents weight. If E is constituted by a set of unordered two tuples, then the weighted graph G is an undirected weighted graph.

Otherwise, it is directed weighted graph. The study of this paper focuses on undirected weighted graph.

3 Extension of Weighted Graph

We can see that there is usually a single weight in weighted graph from the previous research. However, objects are usually affected by more than one factor in the real world. For example, when a user needs a path to reach the destination in the shortest time, we should consider about the length, the probability of traffic jam and the degree of traffic jam. Not only that, different people will have different requirements for the problem of path planning. Some people need the shortest time to reach the destination, while some people need the shortest length to reach there. Therefore, faced with many different requirements, we cannot use the single weight to solve those problems, but need to define a multi-dimension weighted graph to create different models to solve different practical problems.

3.1 Multi-dimension Weighted Graph

Multi-dimension weighted graph is an extension of weighted graph. From the Sect. 2 we have already known that weighted graph G can be represented as follows:

$$G = \ <V, E, W> \tag{1}$$

Multi-dimension weighted graph is not a single weight on every edge. Assume a graph G_1 is a multi-dimension weighted graph. Then it can be represented as follows:

$$G_1 = \ <V, E, (w_1, w_2, \ldots)> \tag{2}$$

Every weight in multi-dimension weighted graph is related to path planning, since the study is based on path planning. Here shows an example based on the G_1.

Assume that G_1 represents a graph of probability of traffic jam. Then V represents a set of location in a city; E, represents a set of roads; $w_1, w_2, \ldots$ represents a set of attributes of the roads. In other words, they are the factors which can affect the path planning. There are three weights $\mathrm{w}_1, \mathrm{w}_2$ and w_3, where w_1 represents the length of every road; w_2 represents the degree of traffic jam; w_3 represents the probability of traffic jam of every road.

Therefore, we can connect the related factors to solve the problem in the real world more exactly. Then we will introduce the aggregate function $f(x)$ to deal with these weights.

3.2 Aggregate Function

Aggregate function $f(x)$ can calculate several weights, and obtain the functional results. That is, we can use aggregate functions to translate several weights into one weight.

There are several common aggregate functions in Excel, such as addition, subtraction, multiplication, division and averaging. For example, the addition aggregate function:

$$f(w_1, w_2, \ldots) = w_1 + w_2 + \cdots \tag{3}$$

The common aggregate functions are too restricted, which can only calculate common data. Users usually face difficult situations, and these aggregate functions can not deal with them well. Therefore, we need to propose aggregate functions for users to calculate special problems for their own demands. If we have the aggregate function which is defined by ourselves, then we can translate the multi-dimension weighted graph $G_1 = \ <V, E, (w_1, w_2, \ldots)>$ into one-dimension weighted graph $G = \ <V, E, w_f>$ by the aggregate function $f(w_1, w_2, \ldots)$, and $w_f = f(w_1, w_2, \ldots)$.

4 Application of Multi-dimension Weighted Graph

We focus on a problem of navigation based on a graph of probability of traffic jam to introduce the two concepts proposed in the third chapter in detail.

4.1 Graph of Probability of Traffic Jam

We establish a graph of probability of traffic jam in order to solve the problem of path planning for those people who are in emergency. This model can reduce the risk to meet the traffic jam. This model not only has the basic information of roads and locations, but also has the attributes which will affect the traffic jam for every road. Graph of probability of traffic jam is a multi-dimension weighted graph. We define it as follows:

$$G = \ <V, E, (w_l, w_h, w_p)> \tag{4}$$

- V represents the locations in the city. We note $A, B, C, \ldots$ to represent them.
- E represents the roads in the city. We note $a, b, c, \ldots$ to represent them.
- (w_l, w_h, w_p) represents three-dimension weights, where w_l represents length of road. We note L, and $L \in (0, +\infty)$; w_h represents the degree of traffic jam. We note H, and $H = \{1, 2, 3\}$ (1 represents a weak degree, 2 represents a common degree, 3 represents a strong degree); w_p represents the probability of traffic jam. We note P, and $P \in [0,1]$.

We study the case of traffic jam in the real world, then we define the following aggregate function $f(x)$:

$$f\left(w_l, w_h, w_p\right) = w_l \times (w_h \times w_p + 1) \tag{5}$$

Then we use the above function to calculate the three-dimension weights to obtain the result. We will establish a graph model to show how to obtain this result.

4.2 Data and Results

We establish 5 nodes and 6 edges. The detail data and the graph model are as follows:

- $V = \{A, B, C, D, E\}$,
- $E = \{a, b, c, d, e\}$,
- The set of three-dimension weights of the 6 edges is W = {(12, 1, 0.1), (11, 2, 0.5), (1, 3, 0.8), (6, 2, 0.3), (15, 2, 0.2), (5, 2, 0.6)}

Figure 1 shows the graph model which stores the above data.

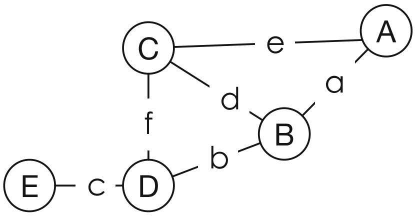

Fig. 1. The graph model

First, put W into the aggregate function $f(x)$, which we have defined before.

For example, w_a = (12, 1, 0.1), then according to the $f(w_l, w_h, w_p) = w_l \times (\mathrm{w_h} \times \mathrm{w_p} + 1)$, $\mathrm{w_{fa}} = 12 \times (1 \times 0.1 + 1) = 13.2$. Then we deal with the result by rounding to get the integer 13. After calculating the three-dimension weights by aggregate function, we get the final w_f = $\{13, 22, 3, 9, 21, 11\}$. Finally we calculate the final result w_f with Dijkstra algorithm [10] to get the final value from every node to other nodes. Table 1 shows the case from node A to other nodes.

Table 1. Result of Dijkstra (A) of graph of probability of traffic jam

B	C	D	E
13	21	32	35

From the aggregate function we can see that the bigger the value is, the higher risk to meet the traffic jam will be.

5 Experiments

5.1 Framework

We show the whole framework in our architecture [11]. First, according to the special problems and different requirements, we establish the suitable models with the related factors, which will affect the result in the real world. Then, consider the relationship among these weights to define the aggregate function $f(w_1, w_2, \ldots)$. After that, put the

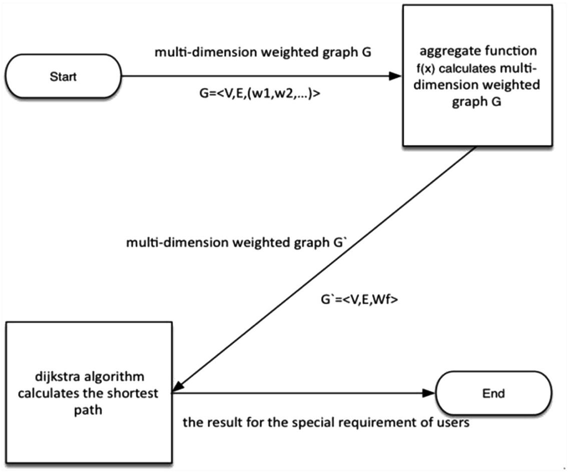

Fig. 2. The framework

weights of multi-dimension weighted graph in the aggregate function to get the final result of weight. This process can realize the translation from multi-dimension weighted graph to one-dimension weighted graphs. Finally, we use the Dijkstra algorithm to get the result which we need. Figure 2 shows the framework.

We will put some more examples to show that the model can solve a lot of practical problems.

5.2 Efficiency

According to the graph of probability of traffic jam, we test the efficiency with the following number data size 10, 20, 30, 40 and 50 and time the corresponding time of program running. Then Fig. 3 shows the result of efficiency.

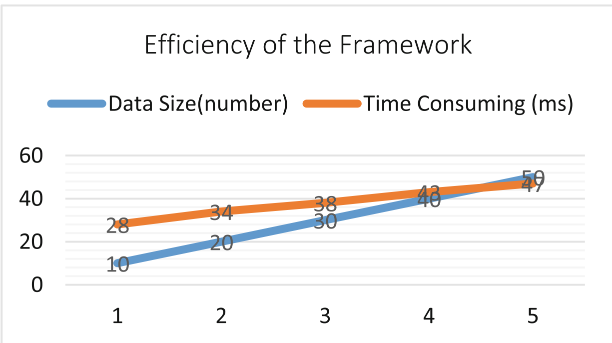

Fig. 3. The efficiency of the framework

From the result we can see that the slope of rising data size is bigger than the slope of rising time consuming. Therefore, the efficiency of the model plays an important role in the age of big data.

5.3 Graph of Traffic Accidents

We want to choose a safer path rather than the shortest path when we go to a dangerous place. According to the traffic accidents, we define the following model:

$$G = \ <V, E, (w_l, w_q, w_v, w_k)> \tag{6}$$

- V represents the locations in the city. We use A, B, C, …to represent them.
- E represents the roads in the city. We note a, b, c, … to represent them.
- (w_l, w_q, w_v, w_k) represents four-dimension weights, where w_l represents length of roads, and we note $L \in (0, +\infty)$; w_q represents the traffic volume. The more the volume is, the higher risk of accidents will be. We note $q = \{1, 2, 3\}$ (1 represents a small volume, 2 represents a middle volume, 3 represents a large volume); w_v represents the maximum speed (the faster the speed is, the more dangerous it will be), and we note $V \in (0, +\infty)$; w_k represents the risk factor, and we note $K \in (0,1)$.

According to the research of accidents, we define the following aggregate function to deal with the four weights:

$$f\left(w_l, w_q, w_v, w_k\right) = \frac{w_l * w_q}{100} * \frac{w_v}{(1 - w_k)} \tag{7}$$

We still use the earlier graph to make the experiment. We establish the following data:

- $V = \{A, B, C, D, E\}$,
- $E = \{a, b, c, d, e\}$,

And the set of four-dimension weights of the 6 edges is W = {(12, 1, 80, 0.1), (11, 2, 70, 0.5), (1, 3, 90, 0.8), (6, 2, 60, 0.3), (15, 2, 75, 0.2), (5, 2, 80, 0.6)}. First, put W into the aggregate function $f(x)$. After calculating the four-dimension weights by aggregate function, we get the final w_f = {10, 30, 13, 10, 28, 20}. Finally we calculate the final result w_f with Dijkstra algorithm to get the final value from every node to other nodes. Table 2 shows the value of risk from node A to other nodes.

Table 2. The result of Dijkstra (A) of graph of traffic accidents

B	C	D	E
10	20	40	53

According to the aggregate function we can see that the bigger the value is, the higher risk to meet the traffic accidents will be.

5.4 Graph of Traffic Cost

According to the framework, we establish a model for the user who care about the traffic cost. we define the following model:

$$G = \; < V, E, (w_l, w_x, w_e) > \tag{8}$$

(w_l, w_t, w_e) is three-dimension weight, where w_l represents length of roads, and we note $L \in (0, +\infty)$; w_t represents cost of consumed fuel of per kilometer, and we use $X \in (0, +\infty)$; w_v represents the maximum speed (the faster speed, the more dangerous), and we use $E \in (0, +\infty)$;

According to the research of cost, we define the following aggregate function to deal with the three weights:

$$f(w_l, w_t, w_e) = w_l \times w_x + w_e \tag{9}$$

We still use the earlier graph to make the experience. We establish the following data:

- $V = \{A, B, C, D, E\}$,
- $E = \{a, b, c, d, e\}$,
- The set of three-dimension weight of the 6 edges is $W = \{(15, 4, 20), (11, 2, 25), (8, 3, 15), (33, 2, 28), (15, 2, 22), (40, 2, 30)\}$.

First, put W into the aggregate function $f(x)$. After calculating the three-dimension weight by aggregate function, we get the final $w_f = \{80, 47, 39, 94, 52, 110\}$. Finally we calculate the final result w_f with Dijkstra algorithm to get the final value from every node to other nodes. Table 3 shows the value of cost from node A to other nodes.

Table 3. The result of Dijkstra (A) of graph of traffic cost

B	C	D	E
80	52	127	166

According to the aggregate function we can see that the bigger the value is, the higher cost spending on the path will be.

5.5 Graph of Traffic Time

According to the framework, we establish a model for the user who care about the traffic time. We define the following model:

$$G = \; < V, E, (w_l, w_v, w_d) > \tag{10}$$

(w_l, w_v, w_d) is three-dimension weight, where w_l represents length of roads, and we note $L \in (0, +\infty)$; w_v represents cost of consumed fuel of per kilometer, and we note $V \in (0, +\infty)$; w_d represents the value of traffic jam. As we know, the traffic time is related to the case of traffic jam. Therefore, we will use the previous result in this model. We note $D = \{13, 22, 3, 9, 21, 11\}$;

According to the research of cost, we define the following aggregate function to deal with the three weights:

$$f(w_l, w_v, w_d) = \frac{w_l}{w_v} * \frac{w_d}{(w_d - 1)} \tag{11}$$

We still use the previous graph to make the experience. We establish the following data:

- $V = \{A, B, C, D, E\}$,
- $E = \{a, b, c, d, e\}$,
- The set of three-dimension weights of the 6 edges is W = {(120, 40, 13), (110, 80, 22), (100, 20, 3), (60, 15, 9), (150, 70, 21), (50, 10, 11)}.

First, we put W into the aggregate function $f(x)$. After calculating the three-dimension weights by aggregate function, we get the final $w_f = \{3, 1, 7, 4, 2, 5\}$. Finally we calculate the final result w_f with Dijkstra algorithm to get the final value from every node to other nodes. Table 4 shows the value of time from node A to other nodes.

Table 4. The result of Dijkstra (A) of graph of

B	C	D	E
3	2	4	11

According to the aggregate function we can see that the bigger the value is, the higher time spending on the path will be.

6 Conclusions

Path planning is a problem which we are always researching and probing. Although the applications of path planning are emerging in an endless stream, there is no suitable application for general users for their personal requirements. This paper establishes a multi-dimension weighted graph to exactly realize the simulation of the practical problems in the real world. We put the factors which will affect each other together to constitute the multi-dimension weighted graph, then according to the relationship among the weights to define aggregate function, which can calculate the factors to meet the different requirements of different users.

This paper establishes a framework by the combination of multi-dimension weighted graphs and aggregate functions. Then we simulate a graph of probability of traffic jam to show the process of this framework. We improve the previous related works based on weighted graph with only one-dimension weight and imperfect aggregate functions. Finally, we make it more suitable to solve the problem of path planning in the real world.

We put some other examples such as the graph of traffic accidents, the graph of traffic cost and the graph of traffic time. Intuitively, the framework can solve a lot of problems, and it can regard the previous result as a factor in this model. We also define the corresponding aggregate function to calculate the three examples above.

We will improve the second processing module, in which we will use another algorithm to deal with the final weight in our future work. We hope the framework can solve a lot of practical problems beyond the path planning.

Acknowledgements. We would like to thank Yaqi Chen for previous survey and useful comments. This work is supported by the program of the National Key Research and Development Program of China (2016YFB1000603) and the National Natural Science Foundation of China (NSFC) (61502336, 61373035). Xiaowang Zhang is supported by Tianjin Thousand Young Talents Program.

References

1. Stiles, P., Glickstein, I.: Route planning. In: IEEE, pp. 420–425 (1991)
2. Zhang, G., Hu, X., Chai, J., Zhao, L., Yu, T.: Summary of path planning algorithm and its application. Mod. Mach. **5**, 85–90 (2011)
3. Yao, C., Li, X., Shen, L.: Weighted directed graph model for searching optimal travel routes by public transport. Appl. Res. Comput. **30**(4), 1058–1063 (2013)
4. Hao, R.: Fire rescue based on shortest route model and its solution strategies. China Sci. Technol. Inf. **19**, 29–30 (2010)
5. Feng, M.: The transportation problem based on general weighted graph. Math. Pract. Theory **38**(9), 131–135 (2008)
6. Flesca, S., Furfaro, F., Greco, S.: Weighted path queries on semistructured databases. Inf. Comput. **204**(5), 679–696 (2006)
7. Stefanescu, D., Thomo, A.: Enhanced regular path queries on semistructured databases. In: Grust, T., et al. (eds.) EDBT 2006. LNCS, vol. 4254, pp. 700–711. Springer, Heidelberg (2006). doi:10.1007/11896548_53
8. Dries, A., Nijssen, S.: Analyzing graph databases by aggregate queries. In: MLG 2010, pp. 37–45, July 2010 (2012)
9. Diestel, R.: Graph Theory, 4th edn. Tsinghua University Press, Beijing (2013)
10. Cormen, T.H., Leiserson, C.E.: Introduction to Algorithm, 2nd edn. Machinery Industry Press, Cambridge (2006)
11. Hellings, J., Kuijpers, B., Van den Bussche, J., Zhang, X.: Walk logic as a framework for path query languages on graph databases. In: Proceedings of ICDT 2013, Genoa, Italy, pp. 117–128. ACM (2013)
12. Zhang, X., Van den Bussche, J.: On the power of SPARQL in expressing navigational queries. Comput. J. **58**(11), 2841–2851 (2015)
13. Fang, H., Zhang, X.: pSPARQL: a querying language for probabilistic RDF (extended abstract). In: Proceedings of ISWC Posters and Demos 2016, Kobe, Japan (2016)

Springer

http://www.springer.com/978-981-10-3167-0

Knowledge Graph and Semantic Computing: Semantic, Knowledge, and Linked Big Data
First China Conference, CCKS 2016, Beijing, China, September 19-22, 2016, Revised Selected Papers

Chen, H.; Ji, H.; Sun, L.; Wang, H.; Qian, T.; Ruan, T. (Eds.)

2016, XIII, 250 p. 50 illus., Softcover
ISBN: 978-981-10-3167-0