氏　　　名	韋　于思
学位の種類	博士（工学）
学位記番号	総博甲第９４号
学位授与年月日	平成２６年３月２５日
学位授与の要件	学位規則第４条第１項
文部科学省報告番号	甲第５２０号
専　攻　名	電子機能システム工学専攻
学位論文題目	Acceleration of Thorup Shortest Paths Algorithm by Modifying Component Tree Demonstrated with Real Datasets （コンポーネント木の改良による Thorup 最短経路アルゴリズムの高速化と実データによる検証）
論文審査委員	主査　島根大学教授　田中　章司郎
	島根大学教授　岡本　　覚
	島根大学教授　濵口　清治
	島根大学准教授　廣冨　哲也

論文内容の要旨

This study provides an improved Thorup algorithm which modified the component tree of the original Thorup algorithm to make it able to maintain the tentative distances of the vertices without constructing a unvisited structure, in order to reduce the time cost of pre-indexing and calculating shortest paths in practice. The experiment indicated, comparing to the array-based Dijkstra, Fibonacci-based Dijkstra and the original Thorup algorithm, reduction by 17.3%, 78.7% and 87.7% of the time cost, respectively.

The single source shortest path (SSSP) is the problem of finding the shortest path from a source vertex to every other vertex. It has been applied in many fields such as navigation [1], keyword searching [2], computer network [3], and it is widely used for finding the optimal route in a road network. SSSP problem is described as the following throughout this research. Given a graph $G=(V,E)$ and a source vertex $s \in V$, suppose s can reach each vertex of the graph, then find the shortest path from s to every vertex $v \in V$, in which V and E represent the vertices and edges of G [4]. The m and n mentioned in the rest of this study represent $|E|$ and $|V|$, respectively. Use $\mathrm{D}(v)$ to represent the tentative distance from the source vertex to v, and use $\mathrm{d}(v)$ to represent the ensured shortest distance from the source vertex to v, and let $\mathrm{L}(v, w)$ represent the positive integer weights of edge(v, w). At the beginning, $\mathrm{D}(v) = \infty$ for every vertex except the source vertex, $\mathrm{d}(s) = 0$. The length of a shortest path should be the sum of the weights of each edge of the shortest path.

One of the most influential shortest path algorithms is Dijkstra [5][4][6] which is proposed in 1959. Yen' s paper [7] is considered to be the first one which implements Dijkstra with an array. In detail, an array is used to record the tentative distance of each vertex which is adjacent to visited vertices. Every time when a vertex is visited, the distances of the vertices which are adjacent to the vertex will be recorded in an array, and then, the vertex which has the shortest distance will be taken to try to relax the vertices which are adjacent to this vertex. The word relax is described as follows. A vertex, say A, can relax another vertex, say B, means the distance from the source vertex to B can be shortened through A. It takes $O(1)$ time to insert a relaxed vertex and $O(n)$ time to delete a vertex from an array. To relax all vertices, Dijkstra algorithm runs in $O(m)$ plus the time of maintaining the array, overall, it takes $O(m + n^2)$.

Thorup [8] is an algorithm which theoretically proved that solving the SSSP problem in linear time with pre-processed index. The paper proposed a hierarchy- and buckets-based algorithm to pre-process indices for performing queries in undirected graphs with non-negative weights. Theoretically, this algorithm constructs the minimum spanning tree in $O(m)$, constructs the component tree in $O(n)$, constructs unvisited data structure in $O(n)$, and calculates distance of all vertices based on constructed structures in $O(m + n)$. But in practice, due to the difficulty of implementation, Thorup algorithm occasionally does not perform as expected according to the experimental result provided by Asano and Imai [9], and Pruehs[10].

One of the reasons is that the data structures given by Thorup are complicated and having complex operations. They are difficult to be efficient in practice.

This study proposes two improvements,

1. Make the component tree able to maintain the tentative distances of vertices.
2. Reduce the depth of the component tree by extending the weights' limitation when creating components.

About the first improvement, the basic idea of the new algorithm proposed in this study is that, the component tree could be an efficient structure for maintaining tentative distances, construct another structure to answer queries from the component tree may not necessary. The new algorithm maintains the tentative distance of each vertex with the component tree, so as to avoid creating and using any unvisited data structures. This change has two benefits as follows.

1. Save the time cost from constructing unvisited structures, which is in the first phase of Thorup algorithm.
2. Accelerate the process of calculating shortest paths, which is in the second phase of Thorup algorithm.

Two variables should be added to each component of a component tree,

1. distance, which is used to record the tentative distance of each component.
2. deleted, which is used to mark that whether the tentative distance of a component is not needed to be updated. It happens when we start to bucket the component' s children.

About the second improvement, the depth of a component tree is reduced by extending the limitation for edges' weights of components in different levels. This will increase the advantage provided by using buckets. To visit components frequently through such a structure will be more efficient.

The improved MX-CIF quadtree [11] has been used to prune the experiment datasets to delete single lines which both sides of this kind of line do not connect to other lines. Otherwise, structures used in the algorithm cannot be constructed.

References

[1] D. En, H. Wei, J. Yang, N. Wei, X. Chen, Y. Liu, "Analysis of the Shortest Path of GPS Vehicle Navigation System Based on Genetic Algorithm," Electrical, Information Engineering and Mechatronics 2011, Springer London, 2012, pp.413-418.

[2] G. Bhalotia, A. Hulgeri, C. Nakhe, S. Chakrabarti, S. Sudarshan, "Keyword Searching and Browsing in Databases using BANKS," ICDE Conf, 2002, pp.431-440.

[3] R. Sivakumar, P. Sinha, V. Bharghavan, "CEDAR: a core-extraction distributed ad hoc routing algorithm," IEEE Journal on Selected Areas in Communications, 17, 1999, pp.1454-1465.

[4] R. Sedgewick, "Algorithms in Java Part 5, Graph 3rd Edition," Addison-Wesley Professional, 2003, ch. 21.

[5] E. W. Dijkstra, "A note on two problems in connexion with graphs," Numerische Mathematik 1, 1959, pp.269-271.

[6] D. Erik, R. Rivest, S. Devadas, "Introduction to Algorithms," Massachusetts Institute of Technology: MIT OpenCourseWare, 2008. Avalible: http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-006-introduction-to-algorithms-spring-2008/lecture-notes/

[7] J. Y. Yen, "A Shortest Path Algorithm," Ph.D. dissertation, University of California, Berkeley, 1970.

[8] M. Thorup, "Undirected single source shortest paths in linear time," Proceedings of the 38th Symposium on Foundations of Computer Science, 1997, pp.12-21.

[9] Y. Asano, H. Imai, "Practical Efficiency of the Linear Time Algorithm for the Single Source Shortest Path problem," Journal of the Operations Research, Society of Japan, Vol. 43, No. 4, 2000, pp.431-447.

[10] N. Pruehs, "Implementation of Thorup's Linear Time Algorithm for Undirected Single-Source Shortest Paths with Positive Integer Weights," Bachelor thesis, University of Kiel, 2009.

[11] Y. Wei and S. Tanaka, 2013 Improvement of MX-CIF Quadtree With Downloadable Source Code and Benchmark Datasets, LAP LAMBERT Academic Publishing, 116 pp.

論文審査結果の要旨

本研究は，Thorup が提案したコンポーネント木の構造を改良することで，索引の作成と最短経路探索の時間計算量を大幅に削減することを可能にしたものである。

辺に正の重み（距離）を持ち単一の出発点からすべての頂点への最短経路を求める問題（Single Source Shortest Paths Problem: SSSP）は，ナビゲーションシステムはもとより，キーワード検索，コンピュータネットワークの OSPF プロトコルなどの多くの用途で頻繁に用いられるアルゴリズムである。

SSSP アルゴリズムは，ダイクストラのアルゴリズムとして大変良く知られる（1959)。その実装プログラムのうち，いまだに影響力があるのは，UC バークレイ校の J.Y. Yen (1970）が提案したものであり，配列データ構造を用いている。配列の要素に各頂点間の距離を一時的に格納し，順次頂点を回訪して最短の距離を更新していく。対象とするグラフの頂点数を n とすると，これには時間計算量 $O(n^2)$を必要としていた。

これに対して Thorup（1997）は，索引構造を工夫することで，検索時間を理論的に線形時間，$O(n)$とするアルゴリズムを開発した。しかしコンポーネント木を用いた実際のデータ構造実装の困難さから，その高速な実現は事実上困難であった。

申請者はこの点に着目し，次の 2 点について，データ構造を工夫した。

1. コンポーネント木のノードに一時的に最短距離を格納する変数を追加した
2. 辺の重みの制約をコンポーネント木の各深さに緩めて拡張することで，その高さを低く抑えた

このことで申請者は，各頂点に対する頻繁な回訪の回数と時間を大幅に削減し，最短経路を短時間に求める方法を考案した。

最短経路検索には，辺が各頂点で連結されている必要がある。従って検証用のデータには，連結の頂点が確実に同じ(x, y) 平面座標値を持たなくてはならない。申請者は国土地理院の道路網の実際の数値情報を，申請者自身が改良を加えて高速化した mx-cif 4 分木を用いて，日本の本州ほぼ全てをカバーする無矛盾の検証用データを整備した。検証用データは，頂点数が約 2,000 のものから，約 20,000 のものまで 5 種類を用意した。

辺に正の重み，つまり距離を持つこれらのデータに対して，配列構造，Fibonacci ヒープ構造，現在までに提案されている Thorup 実装の構造，そして今回申請者が提案した Thorup アルゴリズム改良の構造を比較した結果，全ての場合において，申請者が提案したデータ構造が最も高速な結果を示した。

この SSSP に関する申請者の研究は，依然として改良が求められている計算機科学の中核的な問題に対して，ひとつの実際的な解答を示し，またその結果を踏まえて，さらにデータ構造を改良する展望を拓いたといえる。

このため，申請者の研究成果は，博士の学位にふさわしいものであると判断される。