Journal of King Saud University – Computer and Information Sciences (2014) **26**, 111–119

King Saud University

Journal of King Saud University – Computer and Information Sciences

www.ksu.edu.sa
www.sciencedirect.com

ORIGINAL ARTICLE

A speedup technique for dynamic graphs using partitioning strategy and multithreaded approach

R. Kalpana [a,*], P. Thambidurai [b]

[a] *Dept. of CSE, Pondicherry Engineering College, Puducherry, India*
[b] *Dept. of CSE, Perunthalaivar Kamarajar Institute of Engg. & Tech, Karaikal, Puducherry, India*

Received 30 October 2012; revised 18 May 2013; accepted 22 June 2013
Available online 1 July 2013

KEYWORDS

Dijkstra's algorithm;
Speedup techniques;
Dynamic graphs;
Parallel programming

Abstract There are many pre-processing-based speedup techniques for shortest path problems that are available in the literature. These techniques have an increased demand because of large datasets in such applications such as roadmaps, web search engines and mobile data sets. Pre-processing for the Time-Dependent Shortest Path Problem is still a demanding process that involves graph or network partitioning strategy. Efficient pre-processing of graphs or networks reduces the shortest path computation time while parallelizing the pre-processing phase improves the speedup of the system. In this paper, a speedup technique called Recursive Spectral Bisection (RSB) combined with the Elliptic Convolution of the shortest path method is proposed for dynamic Time-Dependent networks. The same method has been parallelized, and the results are tested on three types of graphs. It is observed that the Time-Dependent RSB combined with the Elliptic Convolution of the shortest path method has no update time, and the Query Performance Loss (QPL) is reduced in planar and road networks compared to random networks. In road networks, the proposed method achieves an average speedup in a QPL of 140. The use of the Parallel speedup technique results in an average speedup in a QPL of more than 1 in the planar and road networks.

1. Introduction

The computation of shortest paths from a given source to a specific destination has the largest scope in the present scenario of real-world applications. The most famous applications are route planning systems for cars, bikes and hikers and timetable information systems for scheduled vehicles, such as trains and buses. If such a system is realized as a central server, it must answer a large number of customer queries in which the customers ask for their best itineraries. Users of such a system continuously enter requests for finding their "best" connections. In addition, similar queries appear as sub-problems in line planning, timetable generation, tour planning, logistics, and traffic simulations. The algorithmic core problem that underlies the above scenario is a special case of the single-source shortest path problem on a given directed graph with non-negative edge lengths. The shortest path queries for such applications were originally solved by Dijkstra, Bellman-Ford, and Johnson. Dijkstra's algorithm implemented with Fibonacci heaps is still the fastest known algorithm for the general

* Corresponding author. Tel.: +91 413 2655282.
E-mail address: rkalpana@pec.edu (R. Kalpana).

Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

1319-1578 © 2013 Production and hosting by Elsevier B.V. on behalf of King Saud University.
http://dx.doi.org/10.1016/j.jksuci.2013.06.002

case of arbitrary nonnegative edge lengths, taking $O(m + n \log n)$ worst-case time. In real-world applications, a layout of the graph is given as an input, and the specific graphs to be considered are notably large. Moreover, the number of queries to be processed within a short time is also notably large. This problem motivates the use of speedup techniques for shortest-path computations (Wagner and Willhalm, 2007).

The main focus of the speedup techniques is to reduce the runtime of the on-line queries (Wagner and Willhalm, 2007). The speedup techniques for the Dijkstra's algorithm can reduce the running time and often result in a sub-linear running time. They crucially depend on the fact that the Dijkstra's algorithm is label setting and that it can be terminated when the destination node is determined. Therefore, the algorithm does not necessarily search the whole graph. Finding the shortest path in a dynamic network is the hottest topic of interest for real-time applications such as vehicle routing, route planning, web searching, and applications in which the edge weight of the graph changes randomly based on the parameters that are being considered. In dynamic road networks, there is always a need for such speedup techniques. In addition, dynamic road networks are of interest from an industrial point of view, and they are usually more accurate models than static networks. Many of these speedup techniques are built on Dijkstra's algorithm and heuristically improve their performance while maintaining their correctness, both in static and dynamic environments.

The focus is now moving to provide such speedup techniques for the Time-Dependent Shortest Path Problem (TDSPP), which constitutes the SPP applied on a Time-Dependent network. The Time-Dependent Shortest Path Problem (TDSPP) is a dynamic graph problem that is NP-hard and non-linear (Li et al., 2005).

In this paper, a new speedup technique is proposed that uses a new partitioning strategy. In addition, this new speedup technique will eliminate the update time and improve the Query Performance Loss (QPL). A Recursive Spectral Bisection partitioning strategy is incorporated in a pre-processing phase, and the Elliptic Convolution of the shortest path method is incorporated in the shortest path computation phase. This new technique is parallelized using parallel programming constructs, and a new speedup technique is proposed. The results of these new techniques are tested in random, planar and road networks.

The remainder of this paper is organized as follows. Section 2 briefly explains the studies that are related to dynamic graph speedup techniques. Section 3 describes the Time-Dependent Shortest Path Problem and the techniques that are used for the shortest path computation. Section 4 provides a detailed experimental evaluation of the proposed new techniques and analyzes the results. Section 5 discusses the concluding remarks.

2. Related work

The arc-flag method is the most popular speedup technique in dynamic environments (Berrettini et al., 2009; D'Angelo et al., 2011, 2012). There are a number of improvements of the basic variant of the arc-flag acceleration for point-to-point shortest path computations on large graphs (Berrettini et al., 2009; D'Angelo et al., 2011). In the case of a dynamic scenario, the changes in arc weight and also the recomputation of the pre-processing phase will be conducted.

There are voluminous contributions on Time-dependent networks. Both FIFO and Non-FIFO Time-dependent shortest problems (Ding et al., 2008) consider both directed and undirected graphs. The computation effort is reduced using an A^* search in bidirectional core-based routing (Delling and Nannicini, 2008) in Time-dependent dynamic environments. The idea behind bidirectional core-based routing (Delling and Nannicini, 2008) is to shrink the original graph to obtain a new core graph with a smaller number of vertices. This action reduces the search process because most of the time the search is conducted on the core. Landmark-based routing (Delling and Wagner, 2007) uses the minimum weight of each edge to compute the distance labels, which calculates the estimated departure time by altering the priority of a node. In Core-ALT (Delling and Nannicini, 2008) routing, an initial contraction step prior to ALT pre-processing must be performed. Landmarks are then chosen from the core and stored with a distance value. In geometric containers (Wagner and Willhalm, 2003), the search space of the algorithm can be reduced by extracting geometric information from a given layout of the graph and by encapsulating precomputed shortest-path information in the geometric containers. In a Goal-directed search, also called A^* (Delling and Nannicini, 2008), the algorithm pushes the search toward a target by adding a potential to the priority of each node. The usage of Euclidean potentials requires no pre-processing. For the use of an arc-flag approach in a dynamic scenario, a new algorithm for update operations that is subject to edge weight decrease operations (D'Angelo et al., 2012) decreases the pre-processing time and the low consumption of the space. A new data structure Road-sign (D'Angelo et al., 2011), which is used in the pre-processing phase of the arc-flag approach, helps to update the edges effectively and reduces the space consumption. The benefits of dynamic time-dependent planning (Ehmke and Mattfeld, 2010), in contrast to common static planning methods, are to achieve better quality results in real-time applications. Traffic data of City logistics show data allocation models that can be applied to any real data sets. The recursive spectral bisection (Zhang et al., 2010) technique is used as an iterative labeling algorithm for network decomposition because this approach is usually used to solve the problem in transportation applications. Here, the network partitioning is made based on the domain, and thus, there is no need to update the data structures.

Currently, every real-world application runs on a multi-core processor that has multiple cores that communicate via shared memory. The multi-core processor model (Sibai, 2013) presents the architecture of core processors and how data partitions are made on core processors. The efficiency of the applications that run on multi-core processors can be improved when **the** parallel Gaussian elimination method is used.

3. Time-Dependent Shortest Path Problem

This section addresses the speedup techniques for shortest path computations in a dynamic environment. The partitioning techniques' recursive spectral bisection method is considered for better improvement in a time-dependent dynamic environment. This new RSB-based partitioning technique dominates the existing arc-flag-based partitioning technique (Mohring