in a directed graph subject to an intermixed sequence of edge insertions and edge deletions. The bounds reported in this entry were originally presented for the case of directed acyclic graphs, but can be extended to general directed graphs using the following theorem from [2]:

Theorem 1 *Given a general directed graph with n vertices, there is a data structure for the fully dynamic reachability problem that supports each insertion/deletion in* $O(n^{1.575})$ *time and each reachability query in* $O(n^{0.575})$ *time. The algorithm is randomized with one-sided error.*

The idea described in [1] is to maintain reachability information from the source vertex s to all other vertices explicitly by keeping a Boolean array R of size n such that $R[y] = 1$ if and only if there is a directed path from s to y. An instance D of the data structure for fully dynamic reachability of Theorem is also maintained. After each insertion or deletion, it is possible to update D in $O(n^{1.575})$ time and then rebuild R in $O(n \cdot n^{0.575}) = O(n^{1.575})$ time by letting $R[y] \leftarrow D.\texttt{reachable}(s,y)$ for each vertex y. This yields the following bounds for the single-source fully dynamic reachability problem:

Theorem 2 *Given a general directed graph with n vertices, there is a data structure for the single-source fully dynamic reachability problem that supports each insertion/deletion in* $O(n^{1.575})$ *time and each reachability query in* $O(1)$ *time.*

Applications

The graph reachability problem is particularly relevant to the field of databases for supporting transitivity queries on dynamic graphs of relations [3]. The problem also arises in many other areas such as compilers, interactive verification systems, garbage collection, and industrial robotics.

Open Problems

An important open problem is whether one can extend the result described in this entry to maintain fully dynamic single-source shortest paths in subquadratic time per operation.

Cross References

► Trade-Offs for Dynamic Graph Problems

Recommended Reading

1. Demetrescu, C., Italiano, G.: Trade-offs for fully dynamic reachability on dags: Breaking through the $O(n^2)$ barrier. J. Assoc. Comput. Machin. (JACM) **52**, 147–156 (2005)
2. Sankowski, P.: Dynamic transitive closure via dynamic matrix inverse. In: FOCS '04: Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS'04), pp. 509–517. IEEE Computer Society, Washington DC (2004)
3. Yannakakis, M.: Graph-theoretic methods in database theory. In: Proc. 9-th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, Nashville, 1990 pp. 230–242

Single-Source Shortest Paths

1999; Thorup

SETH PETTIE
Department of Computer Science,
University of Michigan, Ann Arbor, MI, USA

Keywords and Synonyms

Shortest route; Quickest route

Problem Definition

The *single source* shortest path problem (SSSP) is, given a graph $G = (V, E, \ell)$ and a *source* vertex $s \in V$, to find the shortest path from s to every $v \in V$. The difficulty of the problem depends on whether the graph is directed or undirected and the assumptions placed on the length function ℓ. In the most general situation $\ell \colon E \to \mathbb{R}$ assigns arbitrary (positive & negative) real lengths. The algorithms of Bellman-Ford and Edmonds [1,4] may be applied in this situation and have running times of roughly $O(mn)$,[1] where $m = |E|$ and $n = |V|$ are the number of edges and vertices. If ℓ assigns only *non-negative* real edge lengths then the algorithms of Dijkstra and Pettie-Ramachandran [4,14] may be applied on directed and undirected graphs, respectively. These algorithms include a *sorting bottleneck* and, in the worst case, take $\Omega(m + n \log n)$ time.[2]

A common assumption is that ℓ assigns *integer* edge lengths in the range $\{0, \dots, 2^w - 1\}$ or $\{-2^{w-1}, \dots, 2^{w-1} - 1\}$ and that the machine is a w-bit *word RAM*; that is, each edge length fits in one register. For general integer edge lengths the best SSSP algorithms improve on Bellman-Ford and Edmonds by a factor of roughly $\sqrt{n}$ [7]. For non-negative integer edge lengths the best SSSP algorithms are faster than Dijkstra and Pettie-Ramachandran

[1] Edmonds's algorithm works for undirected graphs and presumes that there are no negative length simple cycles.

[2] The [14] algorithm actually runs in $O(m + n \log\log n)$ time if the ratio of any two edge lengths is polynomial in n.

by up to a logarithmic factor. They are frequently based on integer priority queues [10].

Key Results

Thorup's primary result [17] is an optimal linear time SSSP algorithm for undirected graphs with integer edge lengths. This is the first and only linear time shortest path algorithm that does not make serious assumptions on the class of input graphs.

Theorem 1 *There is a SSSP algorithm for integer-weighted undirected graphs that runs in $O(m)$ time.*

Thorup avoids the sorting bottleneck inherent in Dijkstra's algorithm by precomputing (in linear time) a *component hierarchy*. The algorithm of [17] operates in a manner similar to Dijkstra's algorithm [4] but uses the component hierarchy to identify groups of vertices that can be visited in any order. In later work, Thorup [18] extended this approach to work when the edge lengths are floating-point numbers.[3]

Thorup's hierarchy-based approach has since been extended to directed and/or real-weighted graphs, and to solve the *all pairs* shortest path (APSP) problem [12,13,14]. The generalizations to related SSSP problems are summarized by below. See [12,13] for hierarchy-based APSP algorithms.

Theorem 2 *(Hagerup [9], 2000) A component hierarchy for a directed graph $G = (V, E, \ell)$, where $\ell: E \to \{0, \ldots, 2^w - 1\}$, can be constructed in $O(m \log w)$ time. Thereafter SSSP from any source can be computed in $O(m + n \log\log n)$ time.*

Theorem 3 *(Pettie and Ramachandran [14], 2005) A component hierarchy for an undirected graph $G = (V, E, \ell)$, where $\ell: E \to \mathbb{R}^+$, can be constructed in $O(m\alpha(m, n) + \min\{n \log\log r, n \log n\})$ time, where r is the ratio of the maximum-to-minimum edge length. Thereafter SSSP from any source can be computed in $O(m \log \alpha(m, n))$ time.*

The algorithms of Hagerup [9] and Pettie-Ramachandran [14] take the same basic approach as Thorup's algorithm: use some kind of component hierarchy to identify groups of vertices that can safely be visited in any order. However, the assumption of directed graphs [9] and real edge lengths [14] renders Thorup's hierarchy inapplicable or inefficient. Hagerup's component hierarchy is based on a directed analogue of the minimum spanning tree. The Pettie-Ramachandran algorithm enforces a certain degree of balance in its component hierarchy and, when computing SSSP, uses a specialized priority queue to take advantage of this balance.

[3]There is some flexibility in the definition of *shortest path* since floating-point addition is neither commutative nor associative.

Applications

Shortest path algorithms are frequently used as a subroutine in other optimization problems, such as flow and matching problems [1] and facility location [19]. A widely used commercial application of shortest path algorithms is finding efficient routes on road networks, e. g., as provided by Google Maps, MapQuest, or Yahoo Maps.

Open Problems

Thorup's SSSP algorithm [17] runs in linear time and is therefore optimal. The main open problem is to find a linear time SSSP algorithm that works on *real*-weighted *directed* graphs. For real-weighted undirected graphs the best running time is given in Theorem 3. For integer-weighted directed graphs the fastest algorithms are based on Dijkstra's algorithm (not Theorem 2) and run in $O(m\sqrt{\log\log n})$ time (randomized) and deterministically in $O(m + n \log\log n)$ time.

Problem 1 *Is there an $O(m)$ time SSSP algorithm for integer-weighted directed graphs?*

Problem 2 *Is there an $O(m) + o(n \log n)$ time SSSP algorithm for real-weighted graphs, either directed or undirected?*

The complexity of SSSP on graphs with positive & negative edge lengths is also open.

Experimental Results

Asano and Imai [2] and Pettie et al. [15] evaluated the performance of the hierarchy-based SSSP algorithms [14,17]. There have been a number of studies of SSSP algorithms on integer-weighted directed graphs; see [8] for the latest and references to many others. The trend in recent years is to find practical preprocessing schemes that allow for very quick point-to-point shortest path queries. See [3,11,16] for recent work in this area.

Data Sets

See [5] for a number of US and European road networks.

URL to Code

See [6] and [5].

Cross References

► All Pairs Shortest Paths via Matrix Multiplication

Recommended Reading

1. Ahuja, R.K., Magnati, T.L., Orlin, J.B.: Network Flows: Theory, Algorithms, and Applications. Prentice Hall, Englewood Cliffs (1993)
2. Asano, Y., Imai, H.: Practical efficiency of the linear-time algorithm for the single source shortest path problem. J. Oper. Res. Soc. Jpn. **43**(4), 431–447 (2000)
3. Bast, H., Funke, S., Matijevic, D., Sanders, P., Schultes, D.: In transit to constant shortest-path queries in road networks. In: Proceedings 9th Workshop on Algorithm Engineering and Experiments (ALENEX), 2007
4. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms. MIT Press, Cambridge (2001)
5. Demetrescu, C., Goldberg, A.V., Johnson, D.: 9th DIMACS Implementation Challege—Shortest Paths. http://www.dis.uniroma1.it/~challenge9/ (2006)
6. Goldberg, A.V.: AVG Lab. http://www.avglab.com/andrew/
7. Goldberg, A.V.: Scaling algorithms for the shortest paths problem. SIAM J. Comput. **24**(3), 494–504 (1995)
8. Goldberg, A.V.: Shortest path algorithms: Engineering aspects. In: Proc. 12th Int'l Symp. on Algorithms and Computation (ISAAC). LNCS, vol. 2223, pp. 502–513. Springer, Berlin (2001)
9. Hagerup, T.: Improved shortest paths on the word RAM. In: Proc. 27th Int'l Colloq. on Automata, Languages, and Programming (ICALP). LNCS vol. 1853, pp. 61–72. Springer, Berlin (2000)
10. Han, Y., Thorup, M.: Integer sorting in $O(n\sqrt{\log\log n})$ expected time and linear space. In: Proc. 43rd Symp. on Foundations of Computer Science (FOCS), 2002, pp. 135–144
11. Knopp, S., Sanders, P., Schultes, D., Schulz, F., Wagner, D.: Computing many-to-many shortest paths using highway hierarchies. In: Proceedings 9th Workshop on Algorithm Engineering and Experiments (ALENEX), 2007
12. Pettie, S.: On the comparison-addition complexity of all-pairs shortest paths. In: Proc. 13th Int'l Symp. on Algorithms and Computation (ISAAC), 2002, pp. 32–43
13. Pettie, S.: A new approach to all-pairs shortest paths on real-weighted graphs. Theor. Comput. Sci. **312**(1), 47–74 (2004)
14. Pettie, S., Ramachandran, V.: A shortest path algorithm for real-weighted undirected graphs. SIAM J. Comput. **34**(6), 1398–1431 (2005)
15. Pettie, S., Ramachandran, V., Sridhar, S.: Experimental evaluation of a new shortest path algorithm. In: Proc. 4th Workshop on Algorithm Engineering and Experiments (ALENEX), 2002, pp. 126–142
16. Sanders, P., Schultes, D.: Engineering Highway Hierarchies. In: Proc. 14th European Symposium on Algorithms (ESA), 2006, pp. 804–816
17. Thorup, M.: Undirected single-source shortest paths with positive integer weights in linear time. J. ACM **46**(3), 362–394 (1999)
18. Thorup, M.: Floats, integers, and single source shortest paths. J. Algorithms **35** (2000)
19. Thorup, M.: Quick and good facility location. In: Proceedings 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2003, pp. 178–185

Ski Rental Problem

1990; Karlin, Manasse, McGeogh, Owicki

MARK S. MANASSE
Microsoft Research, Mountain View, CA, USA

Index Terms

Ski-rental problem, Competitive algorithms, Deterministic and randomized algorithms, On-line algorithms

Keywords and Synonyms

Oblivious adversaries, Worst-case approximation, Metrical task systems

Problem Definition

The ski rental problem was developed as a pedagogical tool for understanding the basic concepts in some early results in on-line algorithms.[1] The ski rental problem considers the plight of one consumer who, in order to socialize with peers, is forced to engage in a variety of athletic activities, such as skiing, bicycling, windsurfing, rollerblading, sky diving, scuba-diving, tennis, soccer, and ultimate Frisbee, each of which has a set of associated apparatus, clothing, or protective gear.

In all of these, it is possible either to purchase the accoutrements needed, or to rent them. For the purpose of this problem, it is assumed that one-time rental is less expensive than purchasing. It is also assumed that purchased items are durable, and suitable for reuse for future activities of the same type without further expense, until the items wear out (which occurs at the same rate for all users), are outgrown, become unfashionable, or are disposed of

[1] In the interest of full disclosure, the earliest presentations of these results described the problem as the wedding-tuxedo-rental problem. Objections were presented that this was a gender-biased name for the problem, since while groomsmen can rent their wedding apparel, bridesmaids usually cannot. A further complication, owing to the difficulty of instantaneously producing fitted garments or ski equipment outlined below, suggests that some complications could have been avoided by focusing on the dilemma of choosing between daily lift passes or season passes, although this leads to the pricing complexities of purchasing season passes well in advance of the season, as opposed to the higher cost of purchasing them at the mountain during the ski season. A similar problem could be derived from the question as to whether to purchase the daily newspaper at a newsstand or to take a subscription, after adding the challenge that one's peers will treat one contemptuously if one has not read the news on days on which they have.