Dynamic Shortest Paths and Transitive Closure: an Annotated Bibliography
(Draft)

Camil Demetrescu [*]
Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza", Roma, Italy

Giuseppe F. Italiano [†]
Dipartimento di Informatica, Sistemi e Produzione
Università di Roma "Tor Vergata", Roma, Italy

June 18, 2005

Abstract

This is an annotated bibliography on fully dynamic algorithms for path problems on general directed graphs. In particular, we consider two fundamental problems: dynamic transitive closure and dynamic shortest paths. Although research on these problems spans over more than three decades, in the last couple of years many novel algorithmic techniques have been proposed.

1 Dynamic Path Problems

A dynamic graph algorithm maintains a given property $\mathcal{P}$ on a graph subject to dynamic changes, such as edge insertions, edge deletions and edge weight updates. A dynamic graph algorithm should process queries on property $\mathcal{P}$ quickly, and perform update operations faster than recomputing from scratch, as carried out by the fastest static algorithm. We say that an algorithm is *fully dynamic* if it can handle both edge insertions and edge deletions. A *partially dynamic* algorithm can handle either edge insertions or edge deletions, but not both: we say that it is *incremental* if it supports insertions only, and *decremental* if it supports deletions only. In this annotated bibliography, we focus on fully dynamic algorithms for maintaining path problems on general directed graphs. In particular, we consider two fundamental problems.

In the *fully dynamic transitive closure problem* we wish to maintain a directed graph $G = (V, E)$ under an intermixed sequence of the following operations:

Insert(x, y):	insert an edge from x to y;
Delete(x, y):	delete the edge from x to y;
Query(x, y):	return *yes* if y is reachable from x, and return *no* otherwise.

---

[*] Email: `demetres@dis.uniroma1.it`. URL: `http://www.dis.uniroma1.it/~demetres`.

[†] Email: `italiano@disp.uniroma2.it`. URL: `http://www.disp.uniroma2.it/users/italiano`.

In the *fully dynamic All Pairs Shortest Path (APSP) problem* we wish to maintain a directed graph $G = (V, E)$ with real-valued edge weights under an intermixed sequence of the following operations:

Update(x, y, w):	update the weight of edge (x, y) to the real value w; this includes as a special case both edge insertion (if the weight is set from $+\infty$ to $w < +\infty$) and edge deletion (if the weight is set to $w = +\infty$);
Distance(x, y):	output the shortest distance from x to y.
Path(x, y):	report a shortest path from x to y, if any.

Throughout, we denote by m and by n the number of edges and vertices in G, respectively.

Although research on dynamic transitive closure and dynamic shortest paths problems spans over more than three decades, in the last couple of years we have witnessed a surprising resurge of interests in those two problems.

2 History of the Problems

We first list the bounds obtainable for dynamic transitive closure with simple-minded methods. If we do nothing during each update, then we have to explore the whole graph in order to answer reachability queries: this gives $O(n^2)$ time per query and $O(1)$ time per update in the worst case. On the other extreme, we could recompute the transitive closure from scratch after each update; as this task can be accomplished via matrix multiplication [1, 32], this approach yields $O(1)$ time per query and $O(n^\omega)$ time per update in the worst case, where ω is the best known exponent for matrix multiplication (currently $\omega < 2.736$ [3]).

For the *incremental* version of transitive closure, the first algorithm was proposed by Ibaraki and Katoh [23] in 1983: its running time was $O(n^3)$ over any sequence of insertions. This bound was later improved to $O(n)$ amortized time per insertion by Italiano [24] and also by La Poutré and van Leeuwen [30]. Yellin [44] gave an $O(m^* \delta_{max})$ algorithm for m edge insertions, where m^* is the number of edges in the final transitive closure and δ_{max} is the maximum out-degree of the final graph. All these algorithms maintain explicitly the transitive closure, and so their query time is $O(1)$.

The first *decremental* algorithm was again given by Ibaraki and Katoh [23], with a running time of $O(n^2)$ per deletion. This was improved to $O(m)$ per deletion by La Poutré and van Leeuwen [30]. Italiano [25] presented an algorithm which achieves $O(n)$ amortized time per deletion on directed acyclic graphs. Yellin [44] gave an $O(m^* \delta_{max})$ algorithm for m edge deletions, where m^* is the initial number of edges in the transitive closure and δ_{max} is the maximum out-degree of the initial graph. Again, the query time of all these algorithms is $O(1)$. More recently, Henzinger and King [17] gave a randomized decremental transitive closure algorithm for general directed graphs with a query time of $O(n/\log n)$ and an amortized update time of $O(n \log^2 n)$.

Despite fully dynamic algorithms were already known for problems on undirected graphs since the earlier 80's [12], directed graphs seem to pose much bigger challenges. Indeed, the first *fully dynamic* transitive closure algorithm was devised by Henzinger and King [17] in 1995: they gave a randomized Monte Carlo algorithm with one-side error supporting a query time of $O(n/\log n)$ and an amortized update time of $O(n\widehat{m}^{0.58} \log^2 n)$, where $\widehat{m}$ is the average number of edges in the graph throughout the whole update sequence. Since $\widehat{m}$ can be as high as $O(n^2)$, their update time is $O(n^{2.16} \log^2 n)$. Khanna, Motwani and Wilson [26] proved that, when a lookahead of $\Theta(n^{0.18})$ in the updates is permitted, a deterministic update bound of $O(n^{2.18})$ can be achieved.

The situation for dynamic shortest paths has been even more dramatic. Indeed, the first papers on dynamic shortest paths date back to 1967 [31, 33, 36]. In 1985 Even and Gazit [9] and Rohnert [40] presented algorithms for maintaining shortest paths on directed graphs with arbitrary real weights. Their algorithms required $O(n^2)$ per edge insertion; however, the worst-case bounds for edge deletions were comparable to recomputing APSP from scratch. Also Ramalingam and Reps [34, 35] considered dynamic shortest path algorithms with arbitrary real weights, but in a different model. Namely, the running time of their algorithm is analyzed in terms of the output change rather than the input size (*output bounded complexity*). Frigioni *et al.* [13, 14] designed fast algorithms for graphs with bounded genus, bounded degree graphs, and bounded treewidth graphs in the same model. Again, in the worst case the running times of output-bounded dynamic algorithms are comparable to recomputing APSP from scratch.

Up to few years ago, there seemed to be few dynamic shortest path algorithms which were provably faster than recomputing APSP from scratch, and they only worked on special cases and with small integer weights. In particular, Ausiello *et al.* [2] proposed an incremental shortest path algorithm for directed graphs having positive integer weights less than C: the amortized running time of their algorithm is $O(Cn \log n)$ per edge insertion. Henzinger *et al.* [20] designed a fully dynamic algorithm for APSP on planar graphs with integer weights, with a running time of $O(n^{9/7} \log(nC))$ per operation. Fakcharoemphol and Rao in [10] designed a fully dynamic algorithm for single-source shortest paths in planar directed graphs that supports both queries and edge weight updates in $O(n^{4/5} \log^{13/5} n)$ amortized time per operation.

3 Novel Techniques for Dynamic Path Problems

Quite recently, many new algorithms for dynamic transitive closure and shortest path problems have been proposed.

3.1 Dynamic transitive closure

For dynamic transitive closure, King and Sagert [28] in 1999 showed how to support queries in $O(1)$ time and updates in $O(n^{2.26})$ time for general directed graphs and $O(n^2)$ time for directed acyclic graphs; their algorithm is randomized with one-side error. The bounds of King and Sagert were further improved by King [27], who exhibited a deterministic algorithm on general digraphs with $O(1)$ query time and $O(n^2 \log n)$ amortized time per update operations, where updates are insertions of a set of edges incident to the same vertex and deletions of an arbitrary subset of edges. All those algorithms are based on reductions to fast matrix multiplication and tree data structures for encoding information about dynamic paths.

Demetrescu and Italiano [7] proposed a deterministic algorithm for fully dynamic transitive closure on general digraphs that answers each query with one matrix look-up and supports updates in $O(n^2)$ amortized time. This bound can be made worst-case as shown by Sankowski in [41]. We observe that fully dynamic transitive closure algorithms with $O(1)$ query time maintain explicitly the transitive closure of the input graph, in order to answer each query with exactly one lookup (on its adjacency matrix). Since an update may change as many as $\Omega(n^2)$ entries of this matrix, $O(n^2)$ seems to be the best update bound that one could hope for this class of algorithms. This algorithm hinges upon the well known equivalence between transitive closure and matrix multiplication on a closed semiring [11, 16, 32].

In [6] the authors show how to trade off query times for updates on directed acyclic graphs: each query can be answered in time $O(n^{\epsilon})$ and each update can be performed in time $O(n^{\omega(1,\epsilon,1)-\epsilon}+n^{1+\epsilon})$, for any $\epsilon \in [0,1]$, where $\omega(1,\epsilon,1)$ is the exponent of the multiplication of an $n \times n^{\epsilon}$ matrix by an $n^{\epsilon} \times n$ matrix. Balancing the two terms in the update bound yields that ϵ must satisfy the equation $\omega(1,\epsilon,1) = 1 + 2\epsilon$. The current best bounds on $\omega(1,\epsilon,1)$ [3, 22] imply that $\epsilon < 0.575$. Thus, the smallest update time is $O(n^{1.575})$, which gives a query time of $O(n^{0.575})$. This subquadratic algorithm is randomized, and has one-side error. This result has been generalized to general graphs within the same bounds by Sankowski in [41], who has also shown how to achieve an even faster update time of $O(n^{1.495})$ at the expense of a much higher $O(n^{1.495})$ query time. Roditty and Zwick presented an algorithm [37] with $O(m\sqrt{n}\,)$ update time and $O(\sqrt{n}\,)$ query time and another algorithm [38] with $O(m + n \log n)$ update time and $O(n)$ query time.

Techniques for reducing the space usage of algorithms for dynamic path problems are presented in [29]. An extensive computational study on dynamic transitive closure problems appears in [15].

3.2 Dynamic shortest paths

For dynamic shortest paths, King [27] presented a fully dynamic algorithm for maintaining all pairs shortest paths in directed graphs with positive integer weights less than C: the running time of her algorithm is $O(n^{2.5}\sqrt{C \log n}\,)$ per update. As in the case of dynamic transitive closure, this algorithm is based on clever tree data structures. Demetrescu and Italiano [8] proposed a fully dynamic algorithm for maintaining APSP on directed graphs with arbitrary real weights. Given a directed graph G, subject to dynamic operations, and such that each edge weight can assume at most S different *real* values, their algorithm supports each update in $O(S \cdot n^{2.5} \log^3 n)$ amortized time and each query in optimal worst-case time. We remark that the sets of possible weights of two different edges need not be necessarily the same: namely, any edge can be associated with a different set of possible weights. The only constraint is that throughout the sequence of operations, each edge can assume at most S different real values, which seems to be the case in many applications. Differently from [27], this method uses dynamic reevaluation of products of real-valued matrices as the kernel for solving dynamic shortest paths. Finally, the same authors [4] have studied some combinatorial properties of graphs that make it possible to devise a different approach to dynamic all pairs shortest paths problems. This approach yields a fully dynamic algorithm for general directed graphs with non-negative real-valued edge weights that supports any sequence of operations in $O(n^2 \log^3 n)$ amortized time per update and unit worst-case time per distance query, where n is the number of vertices. Shortest paths can be reported in optimal worst-case time. The algorithm is deterministic, uses simple data structures, and appears to be very fast in practice. Using the same approach, Thorup [42] has shown how to achieve $O(n^2(\log n + \log^2((m+n)/n)))$ amortized time per update and $O(mn)$ space. His algorithm works with negative weights as well. In [43], Thorup has shown how to achieve worst-case bounds at the price of a higher complexity: in particular, the update bounds become $\widetilde{O}(n^{2.75})$, where $\widetilde{O}(f(n))$ denotes $O(f(n) \cdot \text{polylog } n)$.

Lower bounds for dynamic shortest path problems are studied in [39]. An extensive computational study on dynamic all pairs shortest path problems appears in [5].

4 Open Problems

This bulk of recent work has raised some new and perhaps intriguing questions. First, can we reduce the space usage for dynamic shortest paths to $O(n^2)$? Second, and perhaps more importantly, can we solve efficiently fully dynamic *single-source* reachability and shortest paths on general graphs? Finally, are there any general techniques for making increase-only algorithms fully dynamic? Similar techniques have been widely exploited in the case of fully dynamic algorithms on undirected graphs [18, 19, 21].

References

[1] A.V. Aho, J.E. Hopcroft, and J.D. Ullman. *The Design and Analysis of Computer Algorithms.* Addison Wesley, 1974.

[2] G. Ausiello, G.F. Italiano, A. Marchetti-Spaccamela, and U. Nanni. Incremental algorithms for minimal length paths. *Journal of Algorithms*, 12(4):615–38, 1991.

[3] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *Journal of Symbolic Computation*, 9:251–280, 1990.

[4] C. Demetrescu and G. F. Italiano. A new approach to dynamic all pairs shortest paths. *Journal of the Association for Computing Machinery (JACM)*, 51(6):968–992, 2004.

[5] C. Demetrescu and G. F. Italiano. Experimental analysis of dynamic all pairs shortest path algorithms. *To appear in ACM Transactions on Algorithms*, 2005. Special issue devoted to the best papers selected from the 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'04).

[6] C. Demetrescu and G. F. Italiano. Trade-offs for fully dynamic reachability on dags: Breaking through the $O(n^2)$ barrier. *Journal of the Association for Computing Machinery (JACM)*, 52(2):147–156, 2005.

[7] C. Demetrescu and G.F. Italiano. Fully dynamic transitive closure: Breaking through the $O(n^2)$ barrier. In *Proc. of the 41st IEEE Annual Symposium on Foundations of Computer Science (FOCS'00)*, pages 381–389, 2000. Full paper available at the URL: `http://arXiv.org/abs/cs.DS/0104001`.

[8] C. Demetrescu and G.F. Italiano. Fully dynamic all pairs shortest paths with real edge weights. *To appear in Journal of Computer and System Sciences*, 2005.

[9] S. Even and H. Gazit. Updating distances in dynamic graphs. *Methods of Operations Research*, 49:371–387, 1985.

[10] J. Fakcharoemphol and S. Rao. Planar graphs, negative weight edges, shortest paths, and near linear time. In *Proc. of the 42nd IEEE Annual Symposium on Foundations of Computer Science (FOCS'01), Las Vegas, Nevada*, pages 232–241, 2001.

[11] M. J. Fischer and A. R. Meyer. Boolean matrix multiplication and transitive closure. In *Conference Record 1971 Twelfth Annual Symposium on Switching and Automata Theory*, pages 129–131, East Lansing, Michigan, 13–15 October 1971. IEEE.

[12] G.N. Frederickson. Data structures for on-line updating of minimum spanning trees. *SIAM J. Comput.*, 14:781–798, 1985.

[13] D. Frigioni, A. Marchetti-Spaccamela, and U. Nanni. Semi-dynamic algorithms for maintaining single source shortest paths trees. *Algorithmica*, 22(3):250–274, 1998.

[14] D. Frigioni, A. Marchetti-Spaccamela, and U. Nanni. Fully dynamic algorithms for maintaining shortest paths trees. *Journal of Algorithms*, 34:351–381, 2000.

[15] D. Frigioni, T. Miller, U. Nanni, and C. D. Zaroliagis. An experimental study of dynamic algorithms for transitive closure. *ACM Journal of Experimental Algorithms*, 6(9), 2001.

[16] M.E. Furman. Application of a method of fast multiplication of matrices in the problem of finding the transitive closure of a graph. *Soviet Math. Dokl.*, 11(5), 1970. English translation.

[17] M. Henzinger and V. King. Fully dynamic biconnectivity and transitive closure. In *Proc. 36th IEEE Symposium on Foundations of Computer Science (FOCS'95)*, pages 664–672, 1995.

[18] M. Henzinger and V. King. Maintaining minimum spanning forests in dynamic graphs. *SIAM Journal on Computing*, 31(2):364–374, 2001.

[19] M.R. Henzinger and V. King. Randomized fully dynamic graph algorithms with polylogarithmic time per operation. *Journal of the ACM*, 46(4):502–516, 1999.

[20] M.R. Henzinger, P. Klein, S. Rao, and S. Subramanian. Faster shortest-path algorithms for planar graphs. *Journal of Computer and System Sciences*, 55(1):3–23, August 1997.

[21] J. Holm, K. de Lichtenberg, and M. Thorup. Poly-logarithmic deterministic fully-dynamic algorithms for connectivity, minimum spanning tree, 2-edge, and biconnectivity. *Journal of the ACM*, 48:723–760, 2001.

[22] X. Huang and V.Y. Pan. Fast rectangular matrix multiplication and applications. *Journal of Complexity*, 14(2):257–299, June 1998.

[23] T Ibaraki and N. Katoh. On-line computation of transitive closure for graphs. *Information Processing Letters*, 16:95–97, 1983.

[24] G.F. Italiano. Amortized efficiency of a path retrieval data structure. *Theoretical Computer Science*, 48(2–3):273–281, 1986.

[25] G.F. Italiano. Finding paths and deleting edges in directed acyclic graphs. *Information Processing Letters*, 28:5–11, 1988.

[26] S. Khanna, R. Motwani, and R. H. Wilson. On certificates and lookahead in dynamic graph problems. *Algorithmica*, 21(4):377–394, 1998.

[27] V. King. Fully dynamic algorithms for maintaining all-pairs shortest paths and transitive closure in digraphs. In *Proc. 40th IEEE Symposium on Foundations of Computer Science (FOCS'99)*, pages 81–99, 1999.

[28] V. King and G. Sagert. A fully dynamic algorithm for maintaining the transitive closure. *Journal of Computer and System Sciences*, 65(1):150–167, 2002.

[29] V. King and M. Thorup. A space saving trick for directed dynamic transitive closure and shortest path algorithms. In *Proceedings of the 7th Annual International Computing and Combinatorics Conference (COCOON), LNCS 2108*, pages 268–277, 2001.

[30] J. A. La Poutré and J. van Leeuwen. Maintenance of transitive closure and transitive reduction of graphs. In *Proc. Workshop on Graph-Theoretic Concepts in Computer Science*, pages 106–120. Lecture Notes in Computer Science 314, Springer-Verlag, Berlin, 1988.

[31] P. Loubal. A network evaluation procedure. *Highway Research Record 205*, pages 96–109, 1967.

[32] I. Munro. Efficient determination of the transitive closure of a directed graph. *Information Processing Letters*, 1(2):56–58, 1971.

[33] J. Murchland. The effect of increasing or decreasing the length of a single arc on all shortest distances in a graph. Technical report, LBS-TNT-26, London Business School, Transport Network Theory Unit, London, UK, 1967.

[34] G. Ramalingam and T. Reps. An incremental algorithm for a generalization of the shortest path problem. *Journal of Algorithms*, 21:267–305, 1996.

[35] G. Ramalingam and T. Reps. On the computational complexity of dynamic graph problems. *Theoretical Computer Science*, 158:233–277, 1996.

[36] V. Rodionov. The parametric problem of shortest distances. *U.S.S.R. Computational Math. and Math. Phys.*, 8(5):336–343, 1968.

[37] L. Roditty and U. Zwick. Improved dynamic reachability algorithms for directed graphs. In *Proceedings of 43th Annual IEEE Symposium on Foundations of Computer Science (FOCS'02)*, pages 679–688, 2002.

[38] L. Roditty and U. Zwick. A fully dynamic reachability algorithm for directed graphs with an almost linear update time. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC'04)*, pages 184–191, 2004.

[39] L. Roditty and U. Zwick. On dynamic shortest paths problems. In *Proceedings of the 12th European Symposium on Algorithms (ESA'04)*, pages 580–591, 2004.

[40] H. Rohnert. A dynamization of the all-pairs least cost problem. In *Proc. 2nd Annual Symposium on Theoretical Aspects of Computer Science, (STACS'85), LNCS 182*, pages 279–286, 1985.

[41] P. Sankowski. Dynamic transitive closure via dynamic matrix inverse. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS'04), Rome, Italy*, 2004.

[42] M. Thorup. Fully-dynamic all-pairs shortest paths: Faster and allowing negative cycles. In *Proceedings of the 9th Scandinavian Workshop on Algorithm Theory (SWAT'04)*, pages 384–396, 2004.

[43] M. Thorup. Worst-case update times for fully-dynamic all-pairs shortest paths. In *Proceedings of the 37th ACM Symposium on Theory of Computing (STOC 2005)*, 2005.

[44] D. M. Yellin. Speeding up dynamic transitive closure for bounded degree graphs. *Acta Informatica*, 30:369–384, 1993.