A Divide-and-Conquer Algorithm for Min-Cost Perfect Matching in the Plane*

Kasturi R. Varadarajan†

October 5, 1998

Given a set V of $2n$ points in the plane, we can associate a complete undirected graph $G(V)$ (or simply G) with V as follows. The vertex set of G is the set of points V, and its edge set E consists of all unordered pairs (u, v) such that $u, v \in V$ and $u \neq v$. The *cost* of an edge (u, v) is the Euclidean distance $d(u, v)$ between u and v. A *matching* of G (or of V) is a collection M of edges such that no vertex in V is incident on more than one edge in M. A *perfect* matching of V is a matching M in which every vertex in V is incident on *exactly* one edge M. Note that a perfect matching of V has cardinality n. We define the *cost of a matching* M to be $\sum_{(u,v)\in M} d(u, v)$, the sum of the costs of the edges in M. The *Euclidean min-cost perfect matching problem* (MCPM) is to find a perfect matching of V whose cost is the smallest.

The MCPM problem has applications in operations research, pattern recognition, statistics, and VLSI (see [6]). The fact that MCPM and related problems can be solved in polynomial time for general graphs is a classical and fundamental result due to Edmonds [3]. Lawler [5] gave an $O(|V|^3)$ implementation of Edmonds' algorithm; using this, the MCPM problem in the plane can be solved in $O(n^3)$ time. The question that motivates us is whether we can exploit geometry to do much better. (Note that the complete graph induced by the set of $2n$ points is entirely specified by the co-ordinates of the points.)

Vaidya [8] was the first to show that geometry can be exploited to get a sub-cubic algorithm; his $O(n^{5/2}\log^4 n)$-time algorithm is the best known for Euclidean MCPM. For the bipartite version of this problem, Agarwal et al. [1] have given a near-quadratic algorithm that improves over an earlier sub-cubic algorithm of Vaidya [8].

Our results. We present an $O(n^{3/2}\log^5 n)$-time algorithm for computing a min-cost perfect matching in the plane, which is an improvement over the previous best algorithm of Vaidya [8] by nearly a factor of n. Vaidya's algorithm is an implementation of the algorithm of Edmonds [3], which runs in n phases, and computes a matching with i edges at the end of the i-th phase. Vaidya shows that geometry can be exploited to implement a single phase in $\tilde{O}(n^{3/2})$ time (we use the $\tilde{O}()$ notation when ignoring polylog-factors), thus obtaining an $O(n^{5/2}\log^4 n)$-time algorithm. We improve upon this in two major ways. First, we develop a variant of Edmonds' algorithm that uses geometric divide-and-conquer, so that in the conquer step we need only $O(\sqrt{n})$ phases. (Divide-and-conquer has been used before for special cases of MCPM in the plane, for instance by Marcotte

* Work on this paper was done when the author was at Duke University, and was supported by National Science Foundation Grant CCR-93–01259, by an Army Research Office MURI grant DAAH04-96-1-0013, by a Sloan fellowship, by an NYI award, by matching funds from Xerox Corporation, and by a grant from the U.S.–Israeli Binational Science Foundation.

† DIMACS, Rutgers University krv@dimacs.rutgers.edu

and Suri [6], but these approaches rely heavily on the properties of the special cases.) The geometric tool that we use for divide-and-conquer is based on the technique of Miller et al. [7] for finding geometric separators for overlap graphs. Second, we show a single phase for an n-point set can be implemented in $O(n \log^5 n)$ time. To do this, we interpret the dual variables geometrically and establish certain nice properties that they exhibit. We then exploit these properties to show that to implement a single phase, it suffices to look at a subset of $\tilde{O}(n)$ *candidate edges*, and not all the $n(n-1)/2$ edges. The candidate edges are not known at the beginning of the phase itself, but are generated as the phase unfolds, using a total of $\tilde{O}(n)$ time. Combining this with the data structures of Galil et al. [4] for implementing a phase of the matching algorithm for sparse graphs in $\tilde{O}(|E|)$ time, we obtain an $O(n \log^5 n)$-time implementation of a phase. For generating the candidate edges, we introduce a notion called the *semi-separated decomposition*, which is a relaxation of the *well-separated decomposition* of Callahan and Kosaraju [2].

References

[1] P. K. Agarwal, A. Efrat, and M. Sharir. Vertical decomposition of shallow levels in 3-dimensional arrangements and its applications. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 39–50, 1995.

[2] P. B. Callahan and S. R. Kosaraju. A decomposition of multi-dimensional point-sets with applications to k-nearest-neighbors and n-body potential fields. In *Proc. 24th Annu. ACM Sympos. Theory Comput.*, pages 546–556, 1992.

[3] J. Edmonds. Maximum matching and a polyhedron with (0,1) vertices. *J. Res. National Bureau of Standards*, 69B:125–130, 1965.

[4] Z. Galil, S. Micali, and H. N. Gabow. Priority queues with variable priority and an $o(ev \log v)$ algorithm for finding a maximal weighted matching in general graphs. In *Proc. 22nd Annual IEEE Symposium on Foundations of Computer Science*, pages 255–261, 1982.

[5] E. Lawler. *Combinatorial Optimization: Networks and Matroids.* Holt, Rinehart & Winston, New York, 1976.

[6] O. Marcotte and S. Suri. Fast matching algorithms for points on a polygon. In *Proc. 30th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 60–65, 1989.

[7] G. L. Miller, S.-H. Teng, W. Thurston, and S. A. Vavasis. Geometric separators for finite element meshes. *SIAM J. Sci. Comput.*, 19:364–386, 1998.

[8] P. M. Vaidya. Geometry helps in matching. *SIAM J. Comput.*, 18:1201–1225, 1989.