

ON A GREEDY HEURISTIC FOR COMPLETE MATCHING*

EDWARD M. REINGOLD[†] AND ROBERT E. TARJAN[‡]

Abstract. Finding a minimum weighted complete matching on a set of vertices in which the distances satisfy the triangle inequality is of general interest and of particular importance when drawing graphs on a mechanical plotter. The "greedy" heuristic of repeatedly matching the two closest unmatched points can be implemented in worst-case time $O(n^2 \log n)$, a reasonable savings compared to the general minimum weighted matching algorithm which requires time proportional to n^3 to find the minimum cost matching in a weighted graph. We show that, for an even number n of vertices whose distances satisfy the triangle inequality, the ratio of the cost of the matching produced by this greedy heuristic to the cost of the minimal matching is at most $\frac{4}{3}n^{1/2} - 1$, $\lg \frac{3}{2} \approx 0.58496$, and there are examples that achieve this bound. We conclude that this greedy heuristic, although desirable because of its simplicity, would be a poor choice for this problem.

Key words. graph algorithms, matching, greedy heuristic, analysis of algorithms

Introduction. We begin with some motivation, the connection of which to our central topic will become clear later.

The problem of drawing a graph $G = (V, E)$ on a mechanical plotter with pre-specified vertex locations arises in numerous applications [7]. For example, in the solution of shock wave propagation problems by the finite element method [2] it is necessary to plot meshes of thousands of nodes in order to check them visually. Other applications include the drawing of maps, PERT charts, electrical networks, etc. To draw the graph efficiently we must minimize wasted plotter-pen movement, i.e., movement with the pen off the paper so that no line is drawn. The wasted pen movement can be significant; in [5] the use of a naive algorithm resulted in excessively long plotting times.

If the graph G contains an Eulerian cycle or path, then it can be drawn with no wasted pen movement. Moreover, since a graph contains an Eulerian cycle if and only if it contains no vertices of odd degree and an Eulerian path if and only if it contains two vertices of odd degree, it is easy to determine whether either case applies and if so then to use a simple depth-first search algorithm to find the cycle or path [see [10, p. 399]]. If the graph does not contain an Eulerian cycle or path then it contains an even number $n > 2$ of vertices of odd degree. In this case, as a simple consequence of the triangle inequality, the minimum wasted pen movement is achieved by finding a minimum weighted complete matching of the n vertices of odd degree, and drawing the graph by traversing the Eulerian cycle that exists when the edges of the minimum matching are added to the original graph; these edges are traversed with the pen off the paper during the drawing.

The currently known best algorithm for finding a minimum complete matching in a weighted graph requires time proportional to n^3 [8]. For our application that is too inefficient, since n can be fairly large in practice, and we do not want to sacrifice much (relatively expensive) computer time to save (relatively inexpensive) plotter time.

* Received by the editors December 7, 1978, and in revised form October 2, 1980. This research was supported in part by the National Science Foundation under grants NSF MCS 77 22830 and NSF MCS 75-22870, by the Office of Naval Research under contract N00014-76-C-0688, and by a Guggenheim Fellowship.

[†] Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801.

[‡] Department of Computer Science, Stanford University, Stanford, California 94305. Current address: Bell Laboratories, 600 Mountain Avenue, Murray Hill, New Jersey 07974.

However, it would certainly be worthwhile to be able to economize somewhat on plotter time if it could be done without an excessive amount of computer time.

Thus we arrive at the following problem: Can we find near minimum complete matchings of n vertices in the Euclidean plane in, say, time $O(n \log n)$ or time $O(n^2)$ as was done for the traveling salesman problem in [11]?

A greedy heuristic. An obvious heuristic is the following “greedy” algorithm: Repeatedly match the two closest unmatched remaining vertices, resolving any ties in an arbitrary fashion. For n vertices this can be done in worst-case time $O(n^2 \log n)$ by sorting the n^2 distances. This time bound represents reasonable savings for the moderately large values of n encountered in practice, but we must consider how far from minimum the resulting matching will be. The average behavior of this heuristic has been analyzed in [1].

In examining this question we quickly arrive at the sequence of examples in Fig. 1. For 2^t vertices ($t \geq 1$), we have an example in which the minimum matching has cost

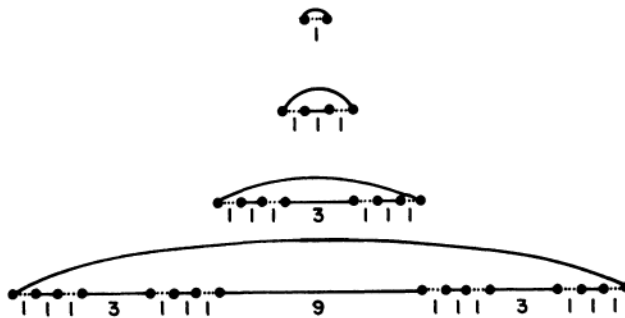


FIG. 1. Examples in which the greedy heuristic produces matchings (shown in solid lines) costing $\frac{4}{3}n^{\lg \frac{3}{2}} - 1$ times as much as the minimal matching (shown in dotted lines) for $n = 2^t$. Comparable examples are easy to construct for N even but not a power of 2.

2^{t-1} , while the cost of the solution produced by the greedy algorithm can be as bad as $2 \cdot 3^{t-1} - 2^{t-1}$ (the solution of the recurrence relation $g_1 = 1, g_{t+1} = 2g_t + 2 \cdot 3^{t-1}$). The ratio of the cost of the matching found by the greedy algorithm to the cost of the minimum matching is

$$\frac{2 \cdot 3^{t-1} - 2^{t-1}}{2^{t-1}} = 2 \cdot \left(\frac{3}{2}\right)^{t-1} - 1 = \frac{4}{3}(2^t)^{\lg \frac{3}{2}} - 1.$$

This tells us that the ratio can be as bad as $\frac{4}{3}n^{\lg \frac{3}{2}} - 1$ for n vertices ($\lg \frac{3}{2} \approx 0.58496$). We now prove that it can be no worse.

THEOREM. Given an even number n of vertices, the distances between which satisfy the triangle inequality, the ratio of the cost of the matching found by the greedy algorithm to the cost of the minimum matching is at most

$$\left(\frac{4}{3}n^{\lg \frac{3}{2}}\right) \frac{3^\theta}{2^{\theta+1} - 1} - 1,$$

where $\theta = \lceil \lg n \rceil - \lg n$.

Before proving this theorem we observe that the function $3^\theta / (2^{\theta+1} - 1)$ is close to 1 throughout the interval $0 \leq \theta < 1$. Its maxima occur at the endpoints $\theta = 0$ and $\theta = 1$ when it is exactly 1; its minimum of approximately 0.94650 occurs at $\theta = \lg$

$\log_2 3 - 1 \approx 0.43803$. The value of $\theta = 0$ corresponds precisely to the examples in Fig. 1, where the number of vertices is a power of 2. For numbers of vertices that are even but not a power of 2, the bound corresponds precisely to examples analogous to those in Fig. 1. The bound of this theorem is thus tight for all even n .

Proof. Observe that the union of any two matchings is a collection of disjoint cycles, the edges of which alternate between the two matchings. (An edge that is in both matchings forms a “double edge” in the union, that is, a cycle of length two.) Consider the collection of such cycles that results from taking the union of the minimum matching and the matching produced by the greedy algorithm for an arbitrary set of $n = 2k$ vertices. Let these cycles be C_1, C_2, \dots, C_m , and let M_i be the sum of the lengths of the edges from the minimum matching in C_i and G_i be the corresponding sum of the edges in C_i from the matching produced by the greedy algorithm. Clearly M_i is the cost of a minimum matching on vertices of C_i and G_i is the cost of the matching resulting from applying the greedy algorithm to the vertices of C_i . We want to bound

$$\begin{aligned} \frac{G_1 + G_2 + \dots + G_m}{M_1 + M_2 + \dots + M_m} &= \frac{M_1}{M_1 + M_2 + \dots + M_m} \frac{G_1}{M_1} + \frac{M_2}{M_1 + M_2 + \dots + M_m} \frac{G_2}{M_2} \\ &\quad + \dots + \frac{M_m}{M_1 + M_2 + \dots + M_m} \frac{G_m}{M_m} \\ &= \alpha_1 \frac{G_1}{M_1} + \alpha_2 \frac{G_2}{M_2} + \dots + \alpha_m \frac{G_m}{M_m}, \end{aligned}$$

where

$$\alpha_i = \frac{M_i}{M_1 + M_2 + \dots + M_m},$$

$\alpha_i > 0, \sum \alpha_i = 1$. Thus

$$\frac{G_1 + G_2 + \dots + G_m}{M_1 + M_2 + \dots + M_m}$$

is a weighted average of $G_1/M_1, G_2/M_2, \dots, G_m/M_m$ and hence less than the largest G_i/M_i . It therefore suffices to consider the ratio of the two costs when the union of the two matchings is a single cycle.

Consider the cycle shown in Fig. 2, in which the edge AD is the last edge added by the greedy algorithm and the edge BC is the penultimate. Linearly scale the edge

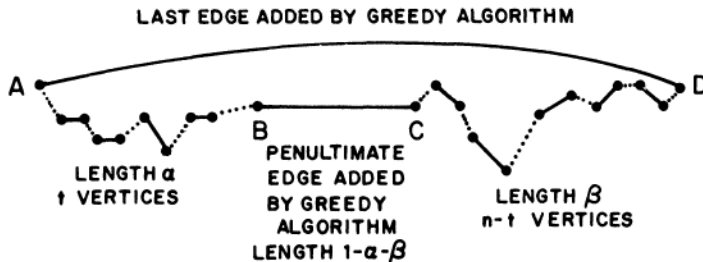


FIG. 2. Proof of the theorem.

lengths so that the sum of the lengths of the edges along the path A to B to C to D is 1. Now, define $f(n)$ to be the smallest fraction of this unit length that can consist of edges of the minimum matching of the n vertices, for any set of n vertices with the property that

the union of the optimal match and the greedy match is a single cycle. Before continuing with an analysis of $f(n)$, observe that the ratio of the matching produced by the greedy algorithm to the minimal matching is bounded above by

$$\frac{\text{length of } AD + (1 - f(n))}{f(n)}.$$

Since the length of AD is at most 1 by the triangle inequality, this ratio is at most

$$(1) \quad \frac{2}{f(n)} - 1.$$

We get a lower bound on $f(n)$ by developing and solving a recurrence relation. Let α be the (scaled) length of the sum of the edges from A to B and let β be the (scaled) length of the sum of the edges from C to D . Thus the edge BC has length $1 - \alpha - \beta$. For the greedy algorithm to choose BC in preference to AB or CD as the penultimate edge selected, we must have $\alpha \geq 1 - \alpha - \beta$ and $\beta \geq 1 - \alpha - \beta$, respectively. Since the vertices B and C are distinct, we also have that $1 - \alpha - \beta > 0$. Letting $t = 2l$ be the number of vertices along the path from A to B and $n - t$ the number along the path from C to D , we have

$$(2) \quad f(n) = \min_{\substack{2 \leq t \leq n-2 \\ \alpha \geq 1 - \alpha - \beta > 0 \\ \beta \geq 1 - \alpha - \beta > 0}} \{ \alpha f(t) + \beta f(n - t) \}$$

and, obviously,

$$f(2) = 1.$$

Since the extremum of a linear function on a polyhedron must occur at a vertex of the polyhedron [3, p. 154], (2) reduces to

$$(3) \quad f(n) = \min \{ f(2), f(4), \dots, f(n - 2), \\ \frac{1}{3}f(2) + \frac{1}{3}f(n - 2), \frac{1}{3}f(4) + \frac{1}{3}f(n - 4), \dots, \frac{1}{3}f(n - 2) + \frac{1}{3}f(2) \},$$

because the vertices of the polyhedron defined by $\alpha \geq 1 - \alpha - \beta \geq 0$ and $\beta \geq 1 - \alpha - \beta \geq 0$ are $(0, 1)$, $(\frac{1}{3}, \frac{1}{3})$, and $(1, 0)$. We now show by induction that the solution to (3) with $f(2) = 1$ is

$$(4) \quad f(2k) = 1 - \sum_{i=2}^k \frac{1}{3^{\lceil \lg i \rceil}}.$$

This is clearly true for $k = 1$. Suppose (4) holds for all $k < t$. Using (4) for $k < t$, we find the minimum occurs at

$$f(2t) = \begin{cases} \frac{2}{3}f(t), & t \text{ even,} \\ \frac{1}{3}f(t+1) + \frac{1}{3}f(t-1), & t \text{ odd.} \end{cases}$$

We consider only the case when t is even; the case of t odd is similar. Let $t = 2u$. From (4) we have

$$f(2t) = \frac{2}{3}f(t) = \frac{2}{3}f(2u) = \frac{2}{3} \left(1 - \sum_{i=2}^u \frac{1}{3^{\lceil \lg i \rceil}} \right) = 1 - \frac{1}{3} - \frac{2}{3} \sum_{i=2}^u \frac{1}{3^{\lceil \lg i \rceil}}.$$

Substituting the identity

$$\frac{1}{3} = \frac{1}{3} \sum_{i=2}^u \frac{1}{3^{\lceil \lg i \rceil}} + \sum_{i=u+1}^{2u} \frac{1}{3^{\lceil \lg i \rceil}},$$

we get

$$\begin{aligned} f(2t) &= 1 - \frac{1}{3} \sum_{i=2}^u \frac{1}{3^{\lceil \lg i \rceil}} - \sum_{i=u+1}^{2u} \frac{1}{3^{\lceil \lg i \rceil}} - \frac{2}{3} \sum_{i=2}^u \frac{1}{3^{\lceil \lg i \rceil}} \\ &= 1 - \sum_{i=2}^{2u} \frac{1}{3^{\lceil \lg i \rceil}} \\ &= 1 - \sum_{i=2}^t \frac{1}{3^{\lceil \lg i \rceil}}, \end{aligned}$$

as claimed.

For $n = 2^l - 2x$, $2^{l-2} > x \geq 0$, (4) becomes

$$\begin{aligned} f(n) &= 1 - \sum_{i=2}^{2^{l-1}} \frac{1}{3^{\lceil \lg i \rceil}} + \frac{x}{3^{l-1}} \\ &= \left(\frac{2}{3}\right)^{l-1} + \frac{x}{3^{l-1}}, \end{aligned}$$

and it follows that if $\theta = \lceil \lg n \rceil - \lg n$ then

$$f(n) = \frac{2^\theta - \frac{1}{2}}{3^{\theta-1}} n^{\lg 3}.$$

From (1) the ratio of interest is at most

$$\left(\frac{4}{3} n^{\lg 3}\right) \frac{3^\theta}{2^{\theta+1} - 1} - 1,$$

as desired.

Conclusions. Together with the examples of Fig. 1, the theorem tells us that the performance of this greedy heuristic is disappointing. Since a similar greedy heuristic for the traveling salesman problem results in tours costing at most twice the cost of the optimal tour when the triangle inequality holds [11], we might have hoped for comparable results for this matching problem. Considering results in [13], we conclude that this greedy heuristic would in general be a poor choice for the matching problem.

It would be interesting to investigate other heuristics. For example, it may be useful to construct the Voronoi diagram in $O(n \log n)$ time and restrict our attention to its straight line dual; Drysdale [4] has shown that this approach will not guarantee an optimal matching, but perhaps it results in a near optimal one. It may also be possible to develop partition algorithms in the style of [6] or [9] that have good average-case behavior. Related results can be found in [1] and [13].

Acknowledgments. We are grateful to Klaus Ecker for pointing out a flaw in an earlier version of the recurrence relation (2).

REFERENCES

- [1] D. AVIS, *Two greedy heuristics for the weighted matching problem*, Proc. Ninth Southeastern Conf. on Combinatorics, Graph Theory, and Computing, 1978, pp. 65–76.
- [2] C. J. CONSTANTINO, *Two dimensional wave propagation through nonlinear media*, J. Comp. Phys., **4** (1969), pp. 147–170.
- [3] G. B. DANTZIG, *Linear Programming and Extensions*, Princeton University Press, Princeton, NJ, 1963.
- [4] S. DRYSDALE, personal communication.
- [5] R. W. GOLLAND, E. M. REINGOLD, R. R. ROBINSON AND A. WACHOWSKI, *Stress waves in layered arbitrary media slam code free-field study*, vol. IV, SAMSO TR 68-181, Department of the Air Force, July 1968.
- [6] R. M. KARP, *Probabilistic analysis of partitioning algorithms for the traveling-salesman problem in the plane*, Math. Oper. Res., **2** (1977), pp. 209–224.
- [7] R. KOPPE, *Automatische Abbildung eines planaren Graphen in die Ebene mit beliebig vorgebbaren Örttern der Knotenbilder*, Computing, **20** (1978), pp. 61–73.
- [8] C. H. PAPADIMITRIOU AND K. STEIGLITZ, *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, Englewood Cliffs, NJ, in press.
- [9] M. O. RABIN, *Probabilistic Algorithms*, in Algorithms and Complexity, Academic Press, New York, 1976.
- [10] E. M. REINGOLD, J. NIEVERGELT AND N. DEO, *Combinatorial Algorithms: Theory and Practice*, Prentice-Hall, Englewood Cliffs, NJ, 1977.
- [11] D. J. ROSENKRANTZ, R. E. STEARNS AND P. M. LEWIS, *An analysis of several heuristics for the traveling salesman problem*, this Journal, **6** (1977), pp. 563–581.
- [12] M. I. SHAMOS AND D. HOEY, *Closest-point problems*, Proc. 16th IEEE Symposium on Foundations of Computer Science, 1975, pp. 151–162.
- [13] K. J. SUPOWIT, D. A. PLAISTED AND E. M. REINGOLD, *Heuristics for weighted perfect matching*, Proc. 12th ACM Symposium on Theory of Computing, 1980, pp. 398–419.