

A Randomized Fully Polynomial-time Approximation Scheme for Weighted Perfect Matching in the Plane

Yasser M. Abd El-Latif
Faculty of Science,
Ain Shams University,
Cairo, Egypt

Salwa M. Ali
Faculty of Science,
Ain Shams University,
Cairo, Egypt

Hanaa A.E. Essa
Faculty of Science,
Tanta University,
Tanta, Egypt

Soheir M. Khamis
Faculty of Science,
Ain Shams University,
Cairo, Egypt

Abstract— In the approximate Euclidean min-weighted perfect matching problem, a set V of $2n$ points in the plane and a real number $\varepsilon > 0$ are given. Usually, a solution of this problem is a partition of points of V into n pairs such that the sum of the distances between the paired points is at most $(1 + \varepsilon)$ times the optimal solution.

In this paper, the authors give a randomized algorithm which follows a Monte-Carlo method. This algorithm is a randomized fully polynomial-time approximation scheme for the given problem. Fortunately, the suggested algorithm is a one tackled the matching problem in both Euclidean nonbipartite and bipartite cases.

The presented algorithm outlines as follows: With repeating $\lceil 1/\varepsilon \rceil$ times, we choose a point from V to build the suitable pair satisfying the suggested condition on the distance. If this condition is achieved, then remove the points of the constructed pair from V and put this pair in M (the output set of the solution). Then, choose a point and the nearest point of it from the remaining points in V to construct a pair and put it in M . Remove the two points of the constructed pair from V and repeat this process until V becomes an empty set. Obviously, this method is very simple. Furthermore, our algorithm can be applied without any modification on complete weighted graphs K_m and complete weighted bipartite graphs $K_{n,n}$, where $n, m \geq 1$ and m is an even.

Keywords- Perfect matching; approximation algorithm; Monte-Carlo technique; a randomized fully polynomial-time approximation scheme; and randomized algorithm.

I. INTRODUCTION

In this paper, the authors deal with Euclidean min-weighted perfect matching problem. This problem and its special cases are very important since they have several applications in many fields such as operations research, pattern recognition, shape matching, statistics, and VLSI, see [1], [2], and [3].

The previous studies treated the underlining problem in several versions, e.g. (un-)weighted general graphs, bipartite graphs, and a case of a set of points in Euclidean plane. For un-weighted bipartite graphs, Hopcroft and Karp showed that maximum-cardinality matchings can be computed in $O(m\sqrt{n})$

time [1]. In [4], Micali and Vazirani introduced an $O(m\sqrt{n})$ algorithm for computing maximum-cardinality matchings on un-weighted graphs. Goel et al. [5] presented an $O(n \log n)$ algorithm for computing the perfect matchings on un-weighted regular bipartite graphs.

In 1955, Kuhn used the Hungarian method for solving the assignment problem. He introduced the first polynomial time algorithm on weighted bipartite graphs having n vertices for computing min-weighted perfect matching in $O(n^3)$ time [6]. The matching problem on complete weighted graphs with $2n$ vertices was solved in $O(n^4)$ by Edmonds' algorithm [7]. In [8], Gabow improved Edmonds' algorithm to achieve $O(n(m + n \log n))$ time, where m is the number of edges in a graph. For the Euclidean versions of the matching problem, Vaidya [9] showed that geometry can be exploited to get algorithms running in $O(n^{5/2} \log^{O(1)} n)$ time for both the bipartite and nonbipartite versions. Agarwal et al. [10] improved this running time for the bipartite case to $O(n^{2+\delta})$, where $\delta > 0$ is an arbitrarily small constant. For nonbipartite case, the running time was improved to $O(n^{3/2} \text{poly} \log n)$, see details in [11].

Several researchers did more efforts to find good approximation algorithms for optimal matching which are faster and simpler than the algorithms obtained the optimal solutions (e.g. [12], [13]).

In [13], Mirjam and Roger gave an approximation algorithm for constructing a minimum-cost perfect matching on complete graphs whose cost functions satisfy the triangle inequality. The running time of that algorithm is $O(n^2 \log n)$ and its approximations ratio is $\log n$. For bipartite version on two disjoint n -point sets in the plane, Agarwal and Varadarajan [14] showed that an ε -approximate matching can be computed in $O((n/\varepsilon)^{3/2} \log^5 n)$ time. In [15], Agarwal and Varadarajan proposed a Monte Carlo algorithm for computing an $O(\log(1/\varepsilon))$ -approximate matching in $O(n^{1+\varepsilon})$ time. Based on Agarwal's ideas, Indyk [16] presented an $O(n \log^{O(1)} n)$ algorithm with probability at least 1/2 and a cost at most $O(1)$ times the cost of the optimal matching. Sharathkumar and Agarwal [17] introduced a Monte-Carlo

algorithm that computes an ε -approximate matching in $O((n/\varepsilon^{O(d)}))poly \log n$ time with high probability.

In this paper, a randomized approximation algorithm for Euclidean min-weighted perfect matching problem is demonstrated. This algorithm follows Monte-Carlo technique. It computes an ε -approximate Euclidean matching of a set of $2n$ points with probability at least $1/2$. The probability is improved when running the algorithm more and more. We show that the algorithm is a *RFPTAS* (Randomized Fully Polynomial Time Approximation Scheme) for underline problem. We apply this algorithm for the Euclidean cases having $2n$ points and for general complete weighted graphs having $2n$ vertices.

The paper is organized as follows. In the next section, some basic definitions and concepts needed for the description of the topic are introduced. In section III, the description of our randomized algorithm for the min-weighted perfect matching problem in the plane is given. It is also proven that the given algorithm is a *RFPTAS*. Moreover, the same section contains some results of computational experiments on some classes of graphs. Finally, a conclusion of the paper is introduced in section IV.

II. BASIC DEFINITIONS AND CONCEPTS

Henceforth, let V be a set of $2n$ points in the plane. A matching of V is a collection M of pairs of V such that no point in V belongs to more than one pair in M [14]. A perfect matching of V is a matching of V in which every point in V belongs to exactly one pair of $z = d(u_1, u_7)$ and so, a perfect matching of V has n pairs. Let $d(u, v)$ be referred to the specified distance between u and v . The weight of a matching M is defined by summing the Euclidean distances between the paired points. For $\varepsilon > 0$, an ε -approximate perfect matching is a perfect matching M such that the weight of M is at most $(1 + \varepsilon)$ times the weight of a minimum weighted perfect matching [17]. In this paper, we consider the approximate Euclidean min-weighted perfect matching problem that finds an ε -approximate perfect matching of V . Let U be an optimization problem. An algorithm A is called a Randomized Fully Polynomial-Time Approximation Scheme (RFPTAS) [18] for U if there exists a function $p: N \times R^+ \rightarrow N$, such that for every input (x, ε) ,

$\Pr(A(x, \varepsilon) \in M(x)) = 1$ {for every random choice A computes a feasible solution of U },

$\Pr(R(x, \varepsilon) \leq 1 + \varepsilon) \geq 1/2$ {a feasible solution, whose approximation ratio is at most $(1 + \varepsilon)$, is produced with the probability at least $1/2$ }, and

$Time(x, \varepsilon^{-1}) \leq p(|x|, \varepsilon^{-1})$ and P is a polynomial in both its arguments $|x|$ and ε^{-1} .

In the approximate Euclidean bipartite min-weighted perfect matching problem, a real number $\varepsilon > 0$ and two disjoint sets V_1 and V_2 such that each of which has n points are given. In this version, if a pair (u, v) belongs to the matching, then $u \in V_1$ and $v \in V_2$. The solution of this problem is to find an ε -approximate perfect bipartite matching of $V_1 \cup V_2$.

In the following, the definition of an admissible pair is given. This definition is very important to design the main condition of our algorithm. We introduce first the meaning of a near point. Suppose V is a set of points. For any $v \in V$, $u \in V$ is called a near point of v if for all $u' \in V$; $d(v, u) \leq d(v, u')$.

Definition 2.1 Let V be a set of points. For any $u, v \in V$, a pair (u, v) is called an admissible pair if $d(u, v) \leq (1 + \varepsilon)d(u, u')$ or $d(u, v) \leq (1 + \varepsilon)d(v, v')$, otherwise it is called an inadmissible pair, where u', v' are the near points of u, v , respectively.

Let V be a set of points and $|V| = 2n$. For each pair $e = (u, v)$ and $u, v \in V$, we consider a random variable X_e defined by $X_e = 1$ if e is an inadmissible pair and $X_e = 0$ otherwise. Since each point $u \in V$ has at least a point $u' \in V$ such that the pair $e = (u, u')$ is an admissible pair (e.g. u' is the near point of u). Thus, $\Pr(X_e = 1) \leq (2n - 2)^2 / (2n)^2 = (n - 1)^2 / n^2$ and $\Pr(X_e = 0) \geq 1 - (n - 1)^2 / n^2$. Let X be a random variable defined by $X = \sum_{e \in M} X_e$, where M is any matching of V .

This means that X counts the number of inadmissible pairs in any matching M .

Let $v \in V$ and $V' \subset V$. Assume that v is a near point of every point in V' . The next lemma gives the upper bound of the cardinality of V' .

Lemma 2.1 Let V be a set of $2n$ points in the plane. If $v \in V$ and $V' \subset V$ satisfying v is a near point to every point in V' , then $|V'| \leq 6$.

Proof. Assume that $|V'| \geq 7$ and v is a near point to every point in V' . First, consider $|V'| = 7$, i.e., $V' = \{u_1, u_2, \dots, u_7\}$. In the plane, connect the point v to all points in V' as shown in Fig.1. Clearly from computational geometry that there exist at least one angle, α less than 60° . Consider w.l.o.g.

$\alpha = \angle u_1 v u_7$. To complete the proof, connect u_1 to u_7 . From Fig.1, $x = d(v, u_1)$, $y = d(v, u_7)$, and $z = d(u_1, u_7)$.

Note that $x \leq y$ So,

$$z^2 = (x \sin \alpha)^2 + (y - x \cos \alpha)^2 = x^2 + y^2 - 2xy \cos \alpha \quad \square$$

Since $\alpha < 60^\circ$ and $x \leq y$. So, $\cos \alpha > 1/2$ and we obtain $z < y$. This means that u_1 is a near one to u_7 rather than v and this is contradiction. Evidentially, when $|V'| > 7$, there exist more angles less than 60° and so we reach to the same contradiction. Therefore, cardinality of V' should be less than or equal 6. ■

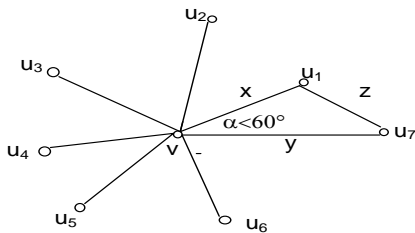


Figure 1. An illustrative example of the proof of Lemma 2.1.

III. A RANDOMIZED ALGORITHM FOR WEIGHTED PERFECT MATCHING

In this section, we describe a Randomized Perfect Matching, *RPM*, algorithm to construct an ε -approximate matching of a set V of $2n$ points in the plane. The analysis of running time and the derivation of an approximation ratio \square for this algorithm is introduced. We deduce that this algorithm is a Randomized Fully Polynomial Time Approximation Scheme.

This section is organized as follows. Section A is devoted to describe the *RPM*-algorithm. In addition, we derive that the *RPM*-algorithm is a RFPTAS for Euclidean min-weighted perfect matching problem in section B. Some results of computational experiments in cases of the set of $2n$ points in the plane, the Euclidean bipartite version, complete weighted graphs, and complete weighted bipartite graphs are contained in section C.

A. The description of *RPM*-algorithm

This part contains the details of the *RPM*-algorithm for Euclidean min-weighted perfect matching problem. The idea of the algorithm summarizes as follows. First, we choose uniformly at random a point v from V and get a near point to it from V , see u . Then, check the suggested condition of Definition 2.1 on $d(v, u)$: If the condition is satisfied, then the pair (v, u) is taking into the output solution M and eliminate v, u from V . Second, after repeating the process $\lceil 1/\varepsilon \rceil$ times, choose uniformly at random a point u' , get a near point to it from V , see v' , and put a pair (u', v') in M .

Delete u', v' from V and repeat this work until V becomes empty. Note that deciding that a pair is admissible based on computing near points w.r.t. all $2n$ points in V ; i.e. before removing any point during the execution of the *RPM*-algorithm.

The steps of the *RPM*-algorithm are introduced in the following:

Algorithm *RPM*

Input: a set V of $2n$ points in the plane and a real number $\varepsilon > 0$.

Output: M // a perfect matching of V .

N // the number of admissible pairs which is used insure that the probability of success $\geq 1/2$.

Begin

- 1: $M \leftarrow \varnothing$;
- 2: $k \leftarrow \lceil 1/\varepsilon \rceil$;
- 3: $N \leftarrow 0$;
- 4: $W \leftarrow V$;
- 5: for $i = 1$ to k do
- 6: choose uniformly at random a point v from W ;
- 7: find u which is a near point to v in W ;
- 8: find v' which is a near point to v in V ;
- 9: find u' which is a near point to u in V ;
- 10: if a pair (v, u) is an admissible pair then
- 11: put a pair (v, u) in M ;
- 12: $W \leftarrow W - \{v, u\}$;
- 13: $N \leftarrow N + 1$;
- 14: while $W \neq \varnothing$ do
- 15: choose uniformly at random a point u from W ;
- 16: find v which is a near point to u in W ;
- 17: $W \leftarrow W - \{u, v\}$;
- 18: put a pair (u, v) in M ;
- 19: find v' which is a near point to v in V ;
- 20: find u' which is a near point to u in V ;

- 21: if a pair (u, v) is an admissible pair then
 22: $N \leftarrow N + 1$;
 23: return M and N ;
 End

As shown in the RPM-algorithm, the steps of this algorithm are simple and can be easily implemented. The main advantage of this method is that it can be applied on any weighted complete graph. This is because V can be considered as the set of vertices of a graph and the distance between any two vertices is measurable by the weight of an edge. Without any modification, this algorithm is also applied on any complete weighted bipartite graph. The results of the RPM-algorithm for solving the underlining problem in cases of the set of $2n$ points in the plane, the Euclidean bipartite version, complete weighted graphs, and complete weighted bipartite graphs are given in section C.

B. The analysis of RPM-algorithm

In this part, we prove that the RPM-algorithm is a RFPTAS for Euclidean min-weighted perfect matching problem. At first, we prove that the probability of success is greater than $1/2$, via using the following two lemmas.

Assume that M' is the set of inadmissible pairs obtained from the output matching of the RPM-algorithm. The following lemma gives an upper bound of the cardinality of M' .

Lemma 3.1 Let V be a set of $2n, n > 4$, points in the plane. If M' is the set of inadmissible pairs obtained from the produced matching M of the RPM-algorithm, then $|M'| \leq 5n/7$.

Proof. We partition the set V into disjoint $k, k > 1$, subsets such that each subset contains a point v and all points that v is a near point to each of them. From Lemma 2.1, we conclude that the number of subsets in this partition is at least $2n/7$. We can obtain at least an admissible pair in each subset in this partition. Thus, the number of admissible pairs in the matching M is at least $2n/7$. Since the cardinality of the matching M is n . Therefore, the number of inadmissible pairs in M is at most $5n/7$. ■

Lemma 3.2 Let V be a set of $2n$ points in the plane. Suppose M is a matching of V which is constructed by the RPM-algorithm and X is a random variable whose value is the number of inadmissible pairs in M . Then, $\Pr(X < n/2) > 1/2$.

Proof. Let M' be the set of inadmissible pairs in M . As mentioned in Lemma 3.1, $|M'| \leq 5n/7$. So, the number of points in inadmissible pairs is at most $10n/7$ and the probability of choosing one of them is at least $7/10n$. In the RPM-algorithm, the steps of for-loop repeat k times,

$k = \lceil 1/\varepsilon \rceil$, so the number of choosing points that do not give admissible pairs is at most $(7/10n)k$. Thus, the number of the admissible pairs which can be obtained in this loop is at least $((10n - 7)/10n)k$. Therefore, the number of the pairs that can be constructed in while-loop does not exceed $A = n - ((10n - 7)/10n)k$.

From linearity of expectation, we obtain

$$E(X) = E\left(\sum_{e \in M} X_e\right) = \sum_{e \in M} E(X_e) = \sum_{e \in M} \Pr(X_e = 1) \leq \frac{(n-1)^2}{n^2} A,$$

where X and X_e are given in section II.

Let $k > 30n^2 / (40n - 28)$, (as a result of executing of the RPM-algorithm more and more). From Morkov's inequality [19], $\Pr(X \geq n/2) \leq E(X)/(n/2) = (2(n-1)^2 A)/n^3$. By using $A = n - ((10n - 7)/10n)k$,

$$\Pr(X \geq n/2) \leq 2(n-1)^2 / n^3 (n - ((10n - 7)/10n)k).$$

Substituting $k = 30n^2 / (40n - 28)$, we get $\Pr(X \geq n/2) < (n-1)^2 / 2n^2 < 1/2$.

Therefore, $\Pr(X < n/2) = 1 - \Pr(X \geq n/2) > 1/2$. This completes the proof. ■

Now, we show that the RPM-algorithm is a RFPTAS.

Theorem 3.3 The RPM-algorithm is a RFPTAS for Euclidean min-weighted perfect matching problem.

Proof. First, we determine the running time of the RPM-algorithm. The time of steps 7- 9 is proportional to $O(n)$. So, the for-loop is executed in time $O(nk)$, where $k = \lceil 1/\varepsilon \rceil$.

Since the number of the pairs that can be constructed in while-loop does not exceed $A = n - ((10n - 7)/10n)k$ and the time of steps 16 and 19- 20 are proportional to $O(n)$, so the time of while-loop is $O(nA)$. Thus, the running time of the RPM-algorithm is $O(nk + nA)$.

Now, we show that the RPM-algorithm finds an ε -approximate perfect matching of V with probability at least $1/2$. It is clear that the output solution M of the RPM-algorithm is a perfect matching of V with cardinality n . From Lemma 3.2, we obtain that the number of inadmissible pairs in M is less than $n/2$ with probability at least $1/2$, hence the number of admissible pairs in M is greater than $n/2$ with the same probability. Let m be the number of admissible pairs in M . According to Definition 2.1, any admissible pair (u, v) satisfies $d(u, v) \leq (1 + \varepsilon)d(u, x)$ or $d(u, v) \leq (1 + \varepsilon)d(v, y)$, where x is a near point of u and y is a near point of v in V . Without loss of generality, we consider the pair (u, x) satisfies the above inequality. So,

$\sum_m d(u,v) \sum_m d(u,x)$. Since u is one coordinate of a pair in an optimal solution, OPT, see (u,z) and $m > n/2$ with probability at least $1/2$. Thus,

$$\sum_m d(u,v) \leq (1 + \varepsilon) \sum_m d(u,z) < (1 + \varepsilon) OPT$$

Obviously, one can deduce from the last equation that the approximation ratio of the RPM-algorithm is at most $(1 + \varepsilon)$ with probability at least $1/2$. This means that the RPM-algorithm is a RFPTAS. ■

C. The results of RPM-algorithm

In this section, we provide some results of the experimental evaluation of the proposed algorithm. We will show that the RPM-algorithm can be applied on four classes without any modification in its steps. These classes are:

- Class 1: Euclidean plane.
- Class 2: Euclidean bipartite plane.
- Class 3: complete weighted graphs.
- Class 4: complete weighted bipartite graphs.

In the first two classes, the inputs of the RPM-algorithm are a real number $\varepsilon > 0$ and a set V . V is the set of $2n$ points which are generated at random. The inputs of the RPM-algorithm in the last two classes are a real number $\varepsilon > 0$ and a complete weighted (bipartite) graph. The weights of edges are positive numbers which are generated at random (in Classes 3, 4) and are the Euclidean distances between any two points (in Classes 1, 2). In table 1, we introduce the numbers of admissible pairs which are produced from the RPM-algorithm for some examples of each class.

Table 1. The performance of the RPM-algorithm.

The values of n	Class 1	Class 2	Class 3	Class 4
20	13	18	16	19
50	33	48	36	49
100	72	90	66	94
300	215	273	190	283
500	364	450	295	454
600	436	538	364	544
1000	714	910	577	912
1200	848	1068	865	1088
1500	1044	1351	873	1379
2000	1376	1776	1122	1819
2030	1395	1807	1137	1850

Since the number of admissible pairs is greater than $n/2$ for each example, so the probability of success of the RPM-algorithm is greater than $1/2$. From these results, the algorithm is suitable for execution on all examples and for any value of n .

Fig. 2 shows the relation between the values of n and the number of admissible pairs. The algorithm is implemented by the Java language. The program works on a 3.00GHz Pentium IV personal computer.

IV. CONCLUSION

In this paper, we introduce a randomized approximation algorithm which follows a Monte-Carlo method for treating the weighted perfect matching problem in the plane. The idea of this algorithm is very simple. Since this algorithm succeeds in computing an ε -approximation matching with probability at least $1/2$ in $O(nk + nA)$ time, where $k = \lceil 1/\varepsilon \rceil$ and $A = n - ((10n - 7)/10n)k$, so this algorithm is a RFPTAS. Our algorithm can be applied without any modification on complete weighted graphs K_m and complete weighted bipartite graphs $K_{n,n}$, where $n, m \geq 1$ and $m = 2n$. The inputs of our algorithm are a set V of $2n$ points in the plane or a complete weighted graph and a real number $\varepsilon > 0$ and the outputs are a perfect matching of an input instance and the number of admissible pairs in this matching. The numbers of admissible pairs which are produced from the suggested algorithm for some examples of each state are summarized in Table 1.

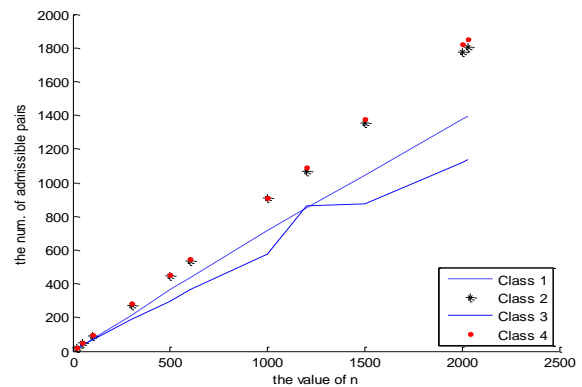


Figure 2. The relation between the values of n and the number of admissible pairs.

REFERENCES

- [1] J. E. Hopcroft and R. M. Karp, An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs, SIAM J. Comput., (1973) 2(4):225–231.
- [2] L. Lovász, On determinants, matchings and random algorithms, In L. Budach, editor, Fundamentals of Computation Theory, FCT '79, Akademie-Verlag Berlin (1979), pages 565–574..
- [3] O. Marcotte and S. Suri, Fast matching algorithms for points on a polygon, In Proc. 30th Annu. IEEE Sympos. Found. Comput. Sci., 1989, pages 60–65.
- [4] S. Micali and V. V. Vazirani, An $o(\sqrt{v})e$ algorithm for finding maximum matching in general graphs, In FOCS, 1980, pages 17–27.
- [5] A. Goel, M. Kapralov, and S. Khanna, Perfect matchings in $O(n \log n)$ time in regular bipartite graphs, In STOC, 2010, pages 39–46.
- [6] H. W. Kuhn, The Hungarian method for the assignment problem, Naval Research Logistics Quarterly, 1955, 2:83–97.
- [7] J. Edmonds, Paths, trees and flowers, Canadian Journal of Mathematics, 1965, 17:449–467.

- [8] H.N. Gabow, Data structures for weighted matching and nearest common ancestors with linking, In Proceedings of the 1st Annual ACM-SIAM Symposium on Discrete Algorithms, 1990, pages 434–443.
- [9] P. M. Vaidya, Geometry helps in matching. SIAM J. Comput., 1989, 18:1201–1225, December.
- [10] P. K. Agarwal, A. Efrat, and M. Sharir, Vertical decomposition of shallow levels in 3-dimensional arrangements and its applications, SIAM J. Comput., 1999, 29:39–50.
- [11] K. R. Varadarajan, A divide-and-conquer algorithm for min-cost perfect matching in the plane, In FOCS, 1998, pages 320–331.
- [12] M.X. Goemans and D.P. Williamson, A general approximation technique for constrained forest problems, In SODA: ACM-SIAM Symposium on Discrete Algorithms (A Conference on Theoretical and Experimental Analysis of Discrete Algorithms), 1992.
- [13] M. Wattenhofer and R. Wattenhofer, Fast and Simple Algorithms for Weighted Perfect Matching, Electronic Notes in Discrete Mathematics 17 (2004) 285–291.
- [14] K.R. Varadarajan and P.K. Agarwal, Approximation algorithms for bipartite and non-bipartite matching in the plane, In SODA: ACM-SIAM Symposium on Discrete Algorithms (A Conference on Theoretical and Experimental Analysis of Discrete Algorithms), 1999.
- [15] P. K. Agarwal and K. R. Varadarajan, A near-linear constant-factor approximation for Euclidean bipartite matching, In Proc. of 12th Annual Sympos. on Comput. Geom., 2004, pages 247–252.
- [16] P. Indyk, A near linear time constant factor approximation for Euclidean bichromatic matching (cost), In Proc. Annual Sympos. on Discrete Algorithms, 2007, pages 39–42.
- [17] R. Sharathkumar and P. K. Agarwal, A Near-Linear Time ϵ -Approximation Algorithm for Bipartite Geometric Matching, Accepted, to appear in STOC 2012.
- [18] J. Hromkovi, Algorithmics for hard problems, Springer, 2001.
- [19] S. M. Ross, Introduction to Probability Models, Academic press an imprint of Elsevier, Vol. 9, 2007, pp. 77-78.
- [20] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. (*references*)
- [21] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
- [22] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
- [23] K. Elissa, “Title of paper if known,” unpublished.
- [24] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
- [25] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetism Japan, p. 301, 1982].
- [26] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.