Lecture Notes on Geometric Approximation Algorithms

Binhai ZHU
Department of Computer Science
Montana State University
Bozeman, MT 59717

Phone: (406)994-4836
Fax : (406)994-4376
email: bhz@cs.montana.edu

I. Definitions

We briefly define some necessary concepts in approximation algorithms. An approximation algorithm for a (maximization) optimization problem Π provides a **performance guarantee** of ρ if for every instance I of Π, the solution value returned by the approximation algorithm is at least $1/\rho$ of the optimal value for I. For the simplicity of description, we simply say that this is a factor ρ approximation algorithm for Π.

Similarly, an approximation algorithm for a (minimization) optimization problem Π provides a **performance guarantee** of ρ if for every instance I of Π, the solution value returned by the approximation algorithm is at most ρ of the optimal value for I.

Notice that following the above definitions, ρ is at least 1.

II. Examples on minimization problems

(2.1) Given a simple polygon P with n vertices $v_1, ..., v_n$, decompose P into minimum number of convex pieces without using Steiner points (i.e., using diagonals only). (Clearly, such a diagonal d must be incident to v and v must be reflex.)

Given a triangulation of P, a diagonal d is said to be *essential for vertex v* if cutting P by removing d creates a piece that is non-convex at v. (v must be incident to d and v must be reflex.) A diagonal which is not essential for any vertex is said to be *inessential.*

Algorithm: Start with a triangulation of P, repeat removing inessential diagonals. Clearly this runs in $O(n)$ time. We will prove that this algorithm (by Hertel and Mehlhorn) has an approximation factor of 4.

Lemma 2.1.1. Let O^* be the minimum number of convex pieces one can obtain and r be the number of reflex vertices in P. Then $\lceil r/2 \rceil + 1 \le O^* \le r + 1$.

Lemma 2.1.2. There can be at most 2 diagonals essential for any reflex vertex.

Theorem 2.1.3. Hertel-Mehlhorn algorithm achieves an approximation factor of 4.

Proof. When the algorithm stops, there can be at most $2r$ essential diagonals left. The number of convex pieces N produced by the algorithm satisfies $2r + 1 \ge N$. By Lemma 2.1.1, $O^* \ge \lceil r/2 \rceil + 1$. Therefore, $4O^* \ge 2r + 4 > 2r + 1 \ge N$. $\square$

(2.2) Geometric TSP (Traveling Salesman Problem): Given n points S in the plane, find the shortest closed path passing through all of them.

The problem is NP-complete (Papadimitriou, 1977). We show below two very simple approximation algorithms. Let the length of the Minimum Spanning Tree of S (MST(S)) be $l(MST(S))$.

Algorithm 1: Compute the Minimum Spanning Tree of S. Double each edge of MST(S) and return the closed path as an approximation.

Theorem 2.2.1. Algorithm 1 achieves an approximation factor of 2 in $O(n \log n)$ time.

Proof. Let O^* be the optimal TSP tour and let its length be $l(O^*)$. Let O_{path} be a path obtained from O^* by deleting an edge. Then, obviously we have $l(MST(S)) \leq l(O_{path})$. Consequently, $l(MST(S)) \leq l(O_{path}) < l(O^*)$. Therefore the approximation solution value $2l(MST(S))$ satisfies, $2l(MST(S)) \leq 2l(O_{path}) < 2l(O^*)$. The time complexity of Algorithm 1 is dominated by the computation of $MST(S)$, which is $O(n \log n)$. $\square$

Algorithm 2: Compute the MST(S). Compute a minimum Euclidean matching M on all vertices with odd degree in MST(S). Traverse MST(S)$\cup M$, which is Eulerian.

Theorem 2.2.2. Algorithm 2 achieves an approximation factor of 1.5 in $O(n^3)$ time.

Proof. Let O^* be the optimal TSP tour and let its length be $l(O^*)$. Clearly, $l(MST(S)) < l(O^*)$ and we need to show that $l(M) \leq \frac{1}{2}l(O^*)$. First, let S' be the set of vertices with odd degrees. So we have $|S'|$ is even. We consider the tour O' obtained by taking shortcuts for odd vertices in O^*. Apparently, we have $l(O') \leq l(O^*)$. Now if we pick up every other edge in O' we obtain two matchings of S', by the optimality of M, $l(M)$ is no larger than the shorter of the two matchings. Therefore, $l(M) \leq \frac{1}{2}l(O') \leq \frac{1}{2}l(O^*)$. Consequently, the approximation solution value $l(MST(S)) + l(M)$ satisfies that $l(MST(S)) + l(M) \leq 1.5l(O^*)$. The running time is dominated by the cost for computing M in $O(n^3)$ time (Gabow, 1972). $\square$

(2.3) Optimal bridge problem: Given two (convex) polygons P, Q, compute two points $p \in P, q \in Q$ such that $d_{u \in P}(u, p) + d(p, q) + d_{v \in Q}(q, v)$ is minimized.

The problem was first studied by Cai, Xu and Zhu [CXZ99]. Recently, it was found that the problem can be solved in linear time (though the solution is still hard). We present below a simple approximation solution.

Algorithm: Compute the minimum distance between P, Q, $d(p^", q^")$. Let u be the furthest vertex of P from $p^"$ and let v be the furthest vertex of Q from $q^"$. Return $(u, p^", q^", v)$ as the approximation solution.

Theorem 2.3.1. The above algorithm presents a factor 2 approximation to the optimal bridge problem.

Proof. Let (u^*, p^*, q^*, v^*) be the the optimal solution. The optimal solution value is $d(u^*, p^*) + d(p^*, q^*) + d(q^*, v^*)$. Let $D(P), D(Q)$ be the diameters of P, Q respectively. Clearly, $d(u^*, p^*) \geq \frac{1}{2}D(P)$ and $d(q^*, v^*) \geq \frac{1}{2}D(Q)$.

On the other hand, $d(u, p^") \leq D(P)$ and $d(q^", v) \leq D(Q)$ since u is the furthest vertex in P from $p^"$ and v is the furthest vertex in Q from $q^"$. Therefore, $\{d(u, p^") + d(p^", q^") + d(q^", v)\} \leq D(P) + d(p^", q^") + D(Q) \leq 2d(u^*, p^*) + d(p^*, q^*) + 2d(q^*, v^*) < 2\{d(u^*, p^*) + d(p^*, q^*) + d(q^*, v^*)\}$.
□

(2.4) Minimum diameter spanning tree problem: Given a set P of n points, compute a spanning tree whose diameter is minimized.

The problem was first studied by Ho, et al. It was found that the problem can be solved in $O(n^3)$ time when the distance is in L_2 [HLCW91]. We present below a simple approximation solution.

Algorithm: Arbitrarily pick up a point p_0 and compute a star S centered at p_0. Return S as an approximation solution.

Theorem 2.4.1. The above algorithm presents a factor 2 approximation to the minimum diameter spanning tree problem.

Proof. Let $D(P)$ be the diameter of P. Let V^* be the value of the optimal solution. Clearly, $V^* \geq D(P)$. Any edge in S is at most $D(P)$, so the diameter of S is at most $2D(P) \leq 2V^*$. $\square$

References (available upon request)

[CXZ99] L. Cai, Y. Xu and *Binhai Zhu*, Computing the optimal bridge between two convex polygons. *Information Processing Letters, 69(3)*, Pages 127-130, Feb, 1999.

[HM83] S. Hertel and K. Mehlhorn. Fast triangulation of simple polygons. *Proc. 4th International Conf. Found. Comput. Theory, Pages 207-218, 1983 (LNCS series, 158).*

[HLCW91] J-M. Ho, D.T. Lee, C-H. Chang and C.K. Wong, Minimum diameter spanning trees and related problems. *SIAM J. Comput., 20*, Pages 987-997, 1991.

[Pa77] C. Papadimitriou. The Euclidean traveling salesman problem is NP-complete. *Theoretical Computer Science, 4*, Pages 237-244, 1977.

III. Examples on maximization problems

(3.1) Diameter problem: Given a set of n points, compute a pair whose inter-distance is the maximum. (The points can be in any fixed dimension.)

The problem can be solved easily in $O(n^2)$ time. We present below two approximation algorithms. (Notice the time-quality tradeoff.)

Algorithm 1: Pick any point w. Find the furthest point x from w. Return (w, x) as an approximation solution.

Theorem 3.1.1: Algorithm 1 achieves an approximation factor of 2 and it runs in $n+O(1)$ time.

Proof. Let (u, v) be the farthest pair. By triangle inequality, for any w, $d(u, w) + d(w, v) \geq d(u, v)$. So the larger of $d(u, w), d(w, v)$ is at least a half of $d(u, v)$. As x is the farthest from w, $d(w, x) \geq max\{d(u, w), d(w, v)\} \geq \frac{1}{2}d(u, v)$. $\square$

Algorithm 2: Find the left-most and right-most point a, b. Find the top-most and bottom-most point c, d. Return the longer of (a, b) and (c, d).

Theorem 3.1.2: Algorithm 2 achieves an approximation factor of $\sqrt{2}$ and it runs in $2n + O(1)$ time.

Proof. Let (u, v) be the farthest pair. Clearly, by definition $d(u, v) \geq$ the longer of (a, b) and (c, d). We have, $d^2(u, v) \leq (x(a) - x(b))^2 + (y(c) - y(d))^2 \leq$ twice the larger of $(x(a) - x(b))^2, (y(c) - y(d))^2$. So $d(u, v) \leq \sqrt{2}\times$ the larger of $|x(a) - x(b)|, |y(c) - y(d)|$, which is $\leq \sqrt{2}\times$ the longer of (a, b) and (c, d). $\square$

(3.2) Labeling points with circle pairs: Given a set S of n points in the plane, label each point with a pair of circles of identical size. The objective is to maximize the radii of the circles such that no two circles intersect and no point is contained in any circle. (A point must be exactly at the tangent point of the two circles.)

The problem is known to be NP-hard [QWXZ00]. The first approximation algorithm was given in [ZP99] and [WTX00]. We will only discuss the algorithm in [ZP99].

Algorithm 1: Find the closest pair $D_2(S)$ of S. Label each point with a pair of circles of radius of $D_2(S)/4$ arbitrarily.

Theorem 3.2.1: Algorithm 1 achieves an approximation factor of 2 and it runs in $O(n \log n)$ time.

Proof. First of all, it is easy to see that no two circles generated by Algorithm 1 intersect each other. As the closest pair has length $D_2(S)$, clearly, the optimal solution value d^* is at most $\frac{1}{2}D_2(S)$, i.e., $\frac{1}{2}D_2(S) \geq d^*$. As the approximation solution value is $\frac{1}{4}D_2(S) \geq d^*$, we have $\frac{1}{4}D_2(S) \geq \frac{1}{2}d^*$. $\square$

(3.3) Labeling points with uniform circles: Given a set S of n points in the plane, label each point with a circle of identical size. The objective is to maximize the radii of the circles such that a point is on the boundary of its labeling circle, no two circles intersect and no point is contained in any circle.

For this problem, I want to illustrate two algorithms. The first one is a simple factor-29.86 approximation which uses a direct method [DMMMZ97]. The second one is a factor-3 approximation which is more involved.

Define $D_3(S)$ as the minimum diameter of all subsets of S with size 3. Let C_i be the open circle centered at $p_i \in S$ with radius $D_3(S)/2$. Clearly, C_i can contain at most two points in S (including its center p_i).

Algorithm 1: Identify the points contained in C_i for all i. If C_i only contains p_i then label p_i with a circle of radius $D_3(S)/8$ arbitrarily; otherwise, if C_i contains another point p_j then label p_i, p_j with circles of radii $D_3(S)/8$ along $< p_j, p_i >$ and $< p_i, p_j >$ respectively.

Theorem 3.3.1: Algorithm 1 achieves an approximation factor of 29.86 and it runs in $O(n \log n)$ time.

Proof. First of all, it is easy to see that no two circles generated by Algorithm 1 intersect each other. Let R^* be the radius of those circles in the optimal solution for the problem. We have $R^* \leq (2+\sqrt{3})D_3(S)$. The approximation solution value is $\frac{1}{8}D_3(S) \geq \frac{1}{8(2+\sqrt{3})}R^* \approx \frac{1}{29.86}R^*$. $\square$

The details of Algorithm 2 are in a separate handout.

References (available upon request)

[DMMMZ97] S. Doddi, M. Marathe, A. Mirzaian, B. Moret and Binhai Zhu, Map labeling and its generalizations. *Proc. 8th ACM-SIAM Symp on Discrete Algorithms (SODA'97), New Orleans, LA, Pages 148-157, Jan, 1997.*

[QWXZ00] Z.P. Qin, Alex Wolff, Yinfeng Xu and Binhai Zhu, New algorithms for two-label point labeling. *Proc. 8th Annu. European Symp. on Algorithms (ESA'00), Pages 368-379, Sep 5-8, 2000 (LNCS series, 1879).*

[WTX00] A. Wolff, M. Thon and Y. Xu. A better lower bound for two-circle point labeling. *Proc. 11th International Symp on Algorithms and Computation (ISAAC'00), Pages 422-431, Dec 18-20, 2000 (LNCS series, 1969).*

[ZP99] Binhai Zhu and C.K. Poon, Efficient approximation algorithms for multi-label map labeling. *Proc. 10th International Symp on Algorithms and Computation (ISAAC'99), Pages 143-152, Dec 16-18, 1999 (LNCS series, 1741).*

IV. Inapproximability results

In geometric optimization, inapproximability results are mainly restricted to inapproximability within some (usually constant) factor. The technique used is usually a NP-hardness reduction (similar to the usual NP-hardness proof). We will illustrate an example here on labeling points with circle pairs (or, alternatively, labeling points with circles if you view each point as a pair of identical points).

The details are in a separate handout.

References (available upon request)

[JBQZ04] M. Jiang, S. Bereg, Z. Qin and B. Zhu. New bounds on labeling points with circular labels. *Proc. 15th Intl. Symp. on Algorithms and Computations (ISAAC'04)*, Pages 606-617, Dec 20-22, 2004 (LNCS series 3341).

V. Decision procedures for designing approximation algorithms

In geometric approximation, usually a direct algorithm based on the properties of the optimal solution will only give a very rough approximation. In reality, to obtain better approximation algorithms, a very typical way is to design a decision procedure. We use MLUC as an example. Suppose that given fixed R, we can decide whether the set of input points S can be labelled with circles of radii R/c, where c is a constant. Then we can run binary search in the domain $[D_2(S)/2, (2+\sqrt{3})D_3(S)]$. By Theorem 3.3.1, R^* must fall within this interval. Then, counting out errors introduced in the binary search process, a factor $c+\epsilon$ can be obtained.

This is the main idea in [JQQZC03]. This idea can also be combined with the griding technique and geometric separators to obtain efficient approximation algorithms (sometimes, PTAS).

It is worth mentioning that this idea can be used in geometric optimization to obtain exact optimal solution. The technique is called parametric search *and is based on using both sequential and parallel decision procedures to speed up the search of optimal solution. The method was initially invented by Megiddo (1983, JACM, vol 30(4):852-865) and Cole (1987, JACM, vol 34(1):200-208). In computational geometry, it has been widely used by the group of Micha Sharir.*

The following is a simple example to apply parametric search to solve the planar 2-watchtower problem (COCOON'2001, SoCG'2005).

——Finally, we show that with parametric search [Me83] the discrete 2-watchtower problem can be solved in $O(n^3 \log^2 n)$ time. The height h of two towers is a parameter of the decision problem which asks whether there exist a pair of vertices v_i and v_j of the terrain T such that every point of the terrain is visible from towers of height h located at v_i and v_j. We need two algorithms, sequential and parallel, for the decision problem. Let T_s be the running time of the sequential decision algorithm. (From the previous discussion, it is easy to see $T_s = O(n^3)$.) Let T_p and P denote the running time and the number of processors of the parallel algorithm respectively. The parametric searching scheme allows us to solve the optimization problem, i.e., the planar discrete 2-watchtower problem, in $O(PT_p + T_P T_s \log P)$ time. We use a parallel algorithm of Atallah et al. [ACW91, JACM] for computing visibility in a simple polygon. Their algorithm is based on a divide-and-conquer technique and can be implemented in the weak parallel model of computation of Valiant [Va75]. The running time is $O(\log n)$ using $O(n/\log n)$ processors. We just need a simplified version of the algorithm with $O(\log n)$ running time and $O(n)$ processors. We apply the algorithm in parallel for all pairs of v_i, v_j using $O(n^3)$ processors. Hence $T_p = O(\log n)$ and $P = O(n^3)$.——

References (available upon request)

[JQQZC03] Minghui Jiang, Jianbo Qian, Zhongping Qin, Binhai Zhu and Robert Cimikowski. A simple factor-3 approximation for labeling points with circles. *Information Processing Letters*, 87:101-105, 2003.

VI. Open problems

(6.1) For decomposing a polygon into convex pieces without using Steiner points, Hertel-Mehlhorn algorithm obtains an approximation factor of 4 and it runs in $O(n)$ time. Can we design an $o(n^2)$ time algorithm to achieve a better approximation factor?

`Status of the problem.` The problem can be solved in $O(n^3 \log n)$ time using dynamic programming (Keil, 1985).

(6.2) For decomposing a polygon into convex pieces with Steiner points, no fast constant-factor approximation algorithm is known. Can we design an $o(n^3)$ time algorithm to achieve a constant-factor approximation?

`Status of the problem.` The problem can be solved in $O(n + r^3) = O(n^3)$ time using dynamic programming (Chazelle, 1980).

(6.3) For computing the minimum diameter spanning tree of a set of n points, can we design an $o(n^3)$ time algorithm to achieve an approximation with a factor better than 2?

`Status of the problem.` The problem can be solved in $O(n^3)$ time using dynamic programming (Ho, Lee, Chang and Wong, 1991). In L_1 metric, the problem can be solved in $O(n^2 \log n)$ time.

(6.4) For problem of labeling points with maximum sliding squares, can we design an algorithm to achieve an approximation with a factor better than 2?

`Status of the problem.` The problem was proved to be NP-hard (in fact, NP-hard to approximate with a factor better than 1.33) (van Kreveld, Strijk and Wolff, 1999). Zhu and Qin obtained a factor-4 approximation (Zhu and Qin, 2002) and a factor-2 approximation (Qin and Zhu, 2002)—the proof was very complex and not really complete in the conference proceeding. Recently, an alternative, very simple factor-3 approximation is obtained (Zhu and Jiang, 2006).

(6.5) For problem of labeling points with maximum uniform circles, can we design an algorithm to achieve an approximation with a factor better than 3?

`Status of the problem.` The problem was proved to be NP-hard (Strijk and Wolff, 2001); in fact, NP-hard to approximate with a factor better than 1.0349 (Jiang, Bereg, Qin and Zhu, 2004). Doddi et al. first obtained a factor-29.86 approximation (SODA'97), and Strijk and Wolff obtained a factor-19.35 approximation in 1999 (journal version: Strijk and Wolff, 2001). Doddi, Marathe and Moret improved the factor significantly to 3.6 in 2000. Recently Jiang et al. improved the factor further to 3 (Jiang at al., 2003) and 2.98 (Jiang et al., 2004).

(6.6) For problem of labeling points with maximum sliding square pairs, can we design an algorithm to achieve an approximation with a factor better than 2?

`Status of the problem.` The problem was proved to be NP-hard (in fact, NP-hard to approximate with a factor better than 1.33) (Spriggs, 2000. manuscript). Zhu and Poon obtained a factor-4 approximation (Zhu and Poon, 1999). Zhu and Qin then obtained a factor-3 approximation (Zhu and Qin, 2002). The best factor of 2 is due to Qin et al. [QWXZ00].

(6.7) For problem of labeling points with maximum circle pairs, can we design an algorithm to achieve an approximation with a factor better than 1.5?

`Status of the problem.` The problem was proved to be NP-hard [QWXZ00]; in fact, NP-hard to approximate with a factor better than 1.0349 (Jiang et al. 2004). (The 1.37 inapproximability result in [QWXZ00] was wrong.) Zhu and Poon obtained a factor-2 approximation (Zhu and Poon, 1999). Qin et al. improved the factor to 1.95 [QWXZ00]. Then, Spriggs and Keil obtained a factor-1.687 approximation (Spriggs and Keil, 2002). The best approximation of 1.5 is due to Wolff, Thon and Xu [WTX00]. Recently the bound is improved to 1.495 (Jiang, et al., 2004).

References (available upon request)

[DMM00] S. Doddi, M. Marathe and B. Moret, Point set labeling with specified positions. *Proc. 16th ACM Symp on Computational Geometry (SOCG'00), Hong Kong, Pages 182-190, June, 2000.* (Also, *International Journal of computational geometry and applications*, 12(1-2):29-66, 2002.)

[DMMMZ97] S. Doddi, M. Marathe, A. Mirzaian, B. Moret and Binhai Zhu, Map labeling and its generalizations. *Proc. 8th ACM-SIAM Symp on Discrete Algorithms (SODA'97), New Orleans, LA, Pages 148-157, Jan, 1997.*

[JQQZC03] Minghui Jiang, Jianbo Qian, Zhongping Qin, Binhai Zhu and Robert Cimikowski. A simple factor-3 approximation for labeling points with circles. *Information Processing Letters*, 87:101-105, 2003.

[JBQZ04] M. Jiang, S. Bereg, Z. Qin and B. Zhu. New bounds on labeling points with circular labels. *Proc. 15th Intl. Symp. on Algorithms and Computations (ISAAC'04)*, Pages 606-617, Dec 20-22, 2004 (LNCS series 3341).

[KSW99] M. van Kreveld, T. Strijk and A. Wolff. Point labeling with sliding labels. *Computational Geometry: Theory and Applications*, 13:21-47, 1999.

[QWXZ00] Z.P. Qin, Alex Wolff, Yinfeng Xu and Binhai Zhu, New algorithms for two-label point labeling. *Proc. 8th Annu. European Symp. on Algorithms (ESA'00), Pages 368-379, Sep 5-8, 2000 (LNCS series, 1879).*

[QZ02] Z.P. Qin and B. Zhu, A factor-2 approximation for labeling points with maximum sliding labels. *Proc. 8th Scandinavian Workshop on Algorithm Theory (SWAT'02), Pages 100-109, July 3-5, 2002 (LNCS series, 2368).*

[SK02] M. Spriggs and M. Keil. A new bound for map labeling with uniform circle pairs. *Information Processing Letters*, 81:47-53, 2002.

[SW99] T. Strijk and A. Wolff. Labeling points with circles. *Technical Report B*, 99-08, Institut für Informatik, Freie Universität Berlin, 1999. (Also, *International Journal of computational geometry and applications*, 11(2):181-195, 2001.)

[WTX00] A. Wolff, M. Thon and Y. Xu. A better lower bound for two-circle point labeling. *Proc. 11th International Symp on Algorithms and Computation (ISAAC'00), Pages 422-431,* *Dec 18-20, 2000 (LNCS series, 1969).* (Also, *International Journal of computational geometry and applications*, 12(4):269-281, 2002.)

[ZJ06] B. Zhu and M. Jiang, A combinatorial theorem on labeling squares with points and its application. *Journal of Combinatorial Optimization, 11(4)*, Pages 411-420, June, 2006.

[ZP99] *Binhai Zhu* and C.K. Poon, Efficient approximation algorithms for multi-label map labeling. *Proc. 10th International Symp on Algorithms and Computation (ISAAC'99), Pages 143-152, Dec 16-18, 1999 (LNCS series, 1741).* (Also, *International Journal of computational geometry and applications*, 11(4):455-464, 2001.)

[ZQ02] B. Zhu and Z.P. Qin, New approximation algorithms for map labeling with sliding labels. *Journal of Combinatorial Optimization, 6(1)*, Pages 99-110, March, 2002.