PLISKA

STUDIA MATHEMATICA BULGARICA

ПЛИСКА

БЪЛГАРСКИ МАТЕМАТИЧЕСКИ СТУДИИ

Pliska Stud. Math. Bulgar. **12** (1998), 57-70

PLISKA
STUDIA MATHEMATICA
BULGARICA

A FAST ALGORITHM FOR WEBER PROBLEM ON A GRID

H. Lee*

The Weber problem is to find a supply point in a plane such that total weighted distance between supply point and a set of demand points is minimized. No known algorithm finds the exact solution. In this paper, we consider a restricted case of Weber problem. The solution domain considered here is confined to grid points (that is, points with integer coordinates). Given m demand points in a plane and a convex polygon P with n vertices, we propose an algorithm to find a supply point with integer coordinates in convex polygon P such that total weighted distance from the point to these m points is minimized. If P is contained in $U \times U$ lattice, the worst case running time of the algorithm is $O((n+m)\log U + \log^2 U)$.

Our method works as follows. Fist, the searching domain polygon P is transformed so that either a number of horizontal grid lines can cover P or a central grid point in P can be found. If the first case holds, the supply point with integer coordinates can be found easily by scanning the grid points on those covering grid lines. Otherwise, we can prune a way a fixed fraction of the polygon by drawing a tangent line through the central grid point so that a smaller searching domain P can be obtained and do the searching recursively.

Keywords: continued fraction expansions, integer programming, location problem, number theory.

AMS subject classification: 90B80

1 Introduction

Weber problem is a well known location problem. This problem was stated in the context of location theory in the celebrated book of Alfred Weber in 1909 [8]. The problem is to find a supply point in a plane such that total weighted distance between supply point and a set of demand points is minimized. This is also known as the 1-median problem.

*This research work was partially supported by the National Science Council of the Republic of China under grant No. NSC86-2416-h-019-001-

No known algorithm finds the exact coordinates of the median (See [3] for a discussion of this difficulty). However, Euclidean p-median problem was shown to be NP-hard [4]. The problem considered here is a restricted version. The solution domain is confined to grid points in a convex polygon. By a grid point, we mean a point with integer coordinates.

The restricted Weber problem can be formulated as follows: Given points $p_1, \dots, p_m \in \mathbb{R}^2$, find a grid point p in a convex polygon P minimizing the total weighted distance function

$$d(p) = \sum_{i=1}^{m} w_i \|p - p_i\|,$$

where w_i are weights and $\| \cdot \|$ is the Euclidean vector norm of $\mathbb{R}^2$. An immediate result for this problem would be $O(mU^2)$ if P is contained in a $U \times U$ lattice. However, it can be easily shown that $d(p)$ is a convex function. This motivates us to propose a general approach for arbitrary objective function which is convex and differentiable. In the rest of the paper, we shall propose a general approach for a more general problem: Given a differentiable convex objective function f defined on $\mathbb{R}^2$ and a convex polygon P formed by n linear constraints, find a integer point in P such that f is minimized. Assume the time to evaluate f is F and P is contained in a $U \times U$ lattice. Time complexity of our algorithm is $O((n+F)\log U + \log^2 U)$. Hence the complexity of our algorithm for the restricted Weber problem is $O((n+m)\log U + \log^2 U)$.

Given a convex polygon P with n vertices, we first try to narrow the search area down by repeatedly pruning some constant fraction of P. This kind of pruning step consists of two actions. One is to find a "central" grid point in P such that any line through this grid point can evenly dissect P; the other is to determine a line through this grid point such that function values of grid points on one side of this line are greater than function value of this grid point. So grid points on this side of the line can be discarded. The pruning step is repeated until the polygon becomes so narrow that the pruning step cannot go further.

If P is narrow, we have difficulties in finding a "central" grid point in P; therefore, the pruning step cannot go further. This is the most difficult situation. An exhaustive search for the optimal solution would be inefficient. Since the objective function f is also convex in a line, grid points which are collinear can be searched by binary search for the grid point with minimum function value: if the function value of a grid point is less than or equal to those of its neighbors, then it is the optimal solution among these collinear grid points; if the function value of a grid point is less than that of its right (left) neighbor and greater than that of its left (right) neighbor, then the optimal solution is on the left (right) side. A line-covering approach is to find a minimal set of parallel lines to cover grid points in P; binary search is then performed on these lines. By continued fraction expansions [2, 5], a set of parallel lines with equal spacing to cover grid points in P can be found. The number of these parallel lines is $O(\sqrt{U})$ in worst case. The general optimization problem considered can be solved in $O(n\log U + F\sqrt{U}\log U)$ time by line-covering approach.

Based on the line-covering approach, a sophisticated algorithm can be obtained. A transformation is devised to transform P to P' when P is narrow. After P is transformed into P', either P' would become wider so that the pruning step can be continued, or grid points in P' can be covered by constant number of parallel lines. If grid points in P' can be covered by a constant number of parallel lines, optimal solution in P' can be found by binary search. With the aid of the transformation, the optimization problem can be solved in $O((n + F) \log U + \log^2 U)$ time.

Rest of the paper is organized as follows: In Section 2, the pruning step is discussed. In Section 3, being a gateway to a more elaborate algorithm, a line- covering algorithm based on the idea of covering grid points in P with parallel lines is presented. In Section 4, the transformation based on the line-covering algorithm is introduced. Algorithm based on the transformation and its complexity are summarized in Section 5. Concluding remarks are given in Section 6.

2 The pruning step

The goal of pruning step is to discard a constant fractional area of P. It consists of two phases: the first is to find a "central" grid point in P so that any line through this grid point dissects P into two even parts; then a line through this grid point is determined so that grid points on one side of this line take values greater than that of the grid point. So grid points on one side can be discarded.

By a "central" grid point in P, we mean a grid point in P so that any line through it dissects P evenly. A more precise definition is given as follows.

Definition 2.1 *A point is said to be an α-pruning point of P, if lines of any orientation through this point divide P into two subpolygons such that ratio of area of the smaller one to that of P is at least α. An α-pruning point is said to be an α-pruning grid point if the point is a grid point.*

In the following, we will show how to find a 1/32-pruning grid point of P. Before describing how to find a 1/32-pruning grid point of P, some definitions are needed.

Definition 2.2 ([7]) *The diameter of a convex polygon is the maximum distance between any two of its vertices. A diametral pair of a convex polygon is a pair of its vertices whose distance is exactly the diameter of the polygon.*

Without loss of generality, we assume the diametral pair of P is unique. If there are more than one diametral pair, we can choose any of them. Let $\{a, b\}$ be a diametral pair of P. Draw two supporting lines parallel to the line through a and b such that P is bounded by these two lines. Distance between these two lines is then said to be width of P. Let $q = (x, y)$ be a grid point. We associate with q the square $N(q) = \left[x - \frac{1}{2}, x + \frac{1}{2}\right) \times \left[y - \frac{1}{2}, y + \frac{1}{2}\right)$ and we say q is the nearest grid point to p if $p \in N(q)$.

A 1/32-pruning grid point of P can be found as follows. First, find a diametral pair of P; let it be $\{a, b\}$. This can be done in $O(n)$ time [7]. Let c be the furthest vertex of P to the line passing through a and b. Then determine the nearest grid point to the centroid of triangle abc. If width of P is greater than $16\sqrt{2}$, then it is a 1/32-pruning grid point. To show this, following lemma is needed.

Lemma 2.1 *The centroid of triangle abc is a* $1/4$*-pruning point of triangle abc.*

PROOF. Let q be the centroid of triangle abc. Let m_1, m_2, and m_3 be mediums of line segments cb, ac and ba respectively. Therefore, $|\overline{am_2}|=|\overline{cm_2}|$, $|\overline{cm_1}|=|\overline{bm_1}|$ and $|\overline{bm_3}|=|\overline{am_3}|$. Let L be a line of any orientation through q. L divides triangle abc into two parts. Without loss of generality, assume the left part is smaller. Its area Q is

$$\text{area of } \triangle abm_1 + \text{area of } \triangle aqt - \text{area of } \triangle sm_1q.$$

Since

$$Q \geq \text{area of } \triangle abm_1 - \text{area of } \triangle sm_1q$$

and

$$\begin{aligned}&\text{area of } \triangle sm_1q\\ &\quad\leq \text{area of } \triangle bm_1q = \frac{1}{2}\text{area of} \triangle bcq\\ &\quad< \frac{1}{2}\text{area of } \triangle bcm_3 = \frac{1}{4}\text{area of } \triangle abc,\end{aligned}$$

we have

$$\begin{aligned}Q &\geq \frac{1}{2}\text{area of} \triangle abc - \frac{1}{4}\text{area of } \triangle abc\\ &= \frac{1}{4}\text{area of } \triangle abc.\end{aligned}$$

Hence, q is a $1/4$-pruning point of triangle abc. □

With Lemma 2.1, we can derive the following lemma.

Lemma 2.2 *If the width of P is greater than or equal to* $16\sqrt{2}$*, then the nearest grid point to the centroid of triangle abc is a* $1/32$*-pruning grid point.*

PROOF. Let $\{a, b\}$ be a diametral pair of P and let c be the farthest vertex of P to line ab. Let q be the centroid of triangle abc and let g be the nearest grid point to q. Let L be a line of any orientation through g and L' be a line parallel to L through q. Let f and e be the intersections of L' with triangle abc. Let t and s be the intersections of L with triangle abc. Assume L is to the left of L'. Since q is a $1/4$-pruning point triangle abc, the area of polygon $astc$ is at least $1/4$ that of triangle abc. If we can show that the area of triangle bts is at least $1/8$ that of triangle abc, then g is a $1/8$-pruning point of triangle abc. Let h denote the distance form c to line ab and l denote the length of line segment ab. Then

$$\begin{aligned}&\text{area of } \triangle bts\\ &\quad= \text{area of } \triangle bfe - \text{area of polygon stfe}\\ &\quad\geq \frac{1}{4}\text{area of } \triangle abc - l \times \frac{\sqrt{2}}{2} = \frac{1}{8}lh - \frac{\sqrt{2}}{2}l.\end{aligned}$$

Since the width of P is greater than or equal to $16\sqrt{2}$, $h \geq 8\sqrt{2}$. Therefore,

$$\frac{\text{area of } \triangle bts}{\text{area of } \triangle abc} \geq \frac{1}{4} - \frac{\sqrt{2}}{h} \geq \frac{1}{8}.$$

Since the area of triangle abc is at least 1/4 that of P, g is 1/32-pruning grid point of P. □

After a 1/32-pruning grid point is found, we have to determine a line through this grid point such that grid points on one side of this line take values greater than that of the grid point. Let p denote a 1/32-pruning grid point of P. Let S_p denote the set $\{q : f(q) \leq f(p), q \in \mathbb{R}^2\}$. S_p is then a convex set [6]. If p is not globally optimal, tangent line to S_p through p, denoted as L_p, is the line we want. If S_p is on the right side of L_p, grid points on left side of L_p take values greater than that of p; they hence can be discarded.

The area of P is $O(U^2)$. Since each pruning step prunes at least 1/32 area of P, the pruning step can be repeated at most $O(\log U)$ times. However, if the width of P is less than $16\sqrt{2}$, the pruning step cannot go further. By intuition, if the width of P is less than $16\sqrt{2}$, grid points in P can be covered by a constant number of parallel lines; binary search can then be performed on these lines. However, the intuition is incorrect. Though there are $O(U)$ grid points in P of constant width, to cover these grid points needs $O(\sqrt{U})$ parallel lines. In next section, a line-covering algorithm based on the idea of covering grid points with minimum parallel lines is presented.

3 A line covering algorithm

Main issue here is to circumvent the desperate situation occurs when P becomes narrow, with the idea of covering grid points with fewer parallel lines such that binary search can then be performed on these lines. Algorithm proposed in this section consists of two phases. In phase 1, the pruning step is repeated until the width of P is less than $16\sqrt{2}$. There are then $O(U)$ grid points in P in the worst case. As mentioned in previous section, it may need $O(\sqrt{U})$ parallel lines to cover grid points in P in the worst case. In phase 2, we try to cover these $O(U)$ grid points with $O(\sqrt{U})$ parallel lines. In the following, we shall show the direction of these parallel lines can be found by continued fraction expansions in time $O(\log U)$.

A vector consisting of two relative prime integers is called a rational vector. Since each line covering more than one gird point has a slope of rational number, direction of a set of parallel straight lines covering grid points in a convex polygon can be represented by a rational vector. A rational vector shall be used to represent the direction of a set of parallel lines; such a vector is called a direction vector. The length of a vector is defined as the Euclidean norm of a vector as usual. In addition, we shall use l to denote the diameter of P; w denotes the width of P; β denotes the slope of the line passing through the diametral pair.

Let octants of plane be defined as eight wedges bounded by lines $x = y$, $x = -y$, $x = 0$, and $y = 0$. Without loss of generality, assume $0 \leq \beta \leq 1$. We only have to pay attention to direction vectors in Octant 1, since each direction vector in other octants has one correspondent in Octant 1 and it is no better than its correspondent in Octant 1. Each rational vector in Octant 1 has one corresponding irreducible fraction in $[0, 1]$. Since there are $O\Big(\frac{3U^2}{\pi^2}\Big)$ irreducible fractions in $[0,1]$ with denominators less than or equal to U [1], there are $O\Big(\frac{3U^2}{\pi^2}\Big)$ possible directions of parallel lines to cover grid points in P. An exhaustive search for the direction, in which the number of parallel lines to cover grid points in P is $O(\sqrt{U})$, needs $O(U^2)$ time. Based on continued fraction expansions, such a direction can be found by a search described in the following, which is as effective as binary search.

By the method called continued fraction expansions, β can be uniquely written in the form

$$\beta = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \cdots}}$$

where $a_0 = \lfloor \beta \rfloor$ and $a_1, a_2, \ldots$ are positive integers. In fact, $a_1, a_2, \ldots$ can be defined by the recurrence

$$\beta_0 = \beta, a_0 = \lfloor \beta \rfloor, \beta_{k+1} = \frac{1}{\beta_k - a_k}, a_{k+1} = \lfloor \beta_{k+1} \rfloor.$$

Let the rational number

$$\frac{g_k}{h_k} = a_0 + \cfrac{1}{a_1 + \cfrac{1}{\ddots_{a_{k-1} + \frac{1}{a_k}}}}.$$

Then $\frac{g_k}{h_k}$ is called k-th convergent of β [2]. For example, if $\beta = \frac{123}{638}$, by continued fraction expansions,

$$\frac{123}{638} = 0 + \cfrac{1}{5 + \cfrac{1}{5 + \cfrac{1}{2 + \cfrac{1}{1 + \frac{1}{7}}}}}.$$

Its convergents are then $\frac{g_0}{h_0} = \frac{0}{1}$, $\frac{g_1}{h_1} = \frac{1}{5}$, $\frac{g_2}{h_2} = \frac{5}{26}$, $\frac{g_3}{h_3} = \frac{11}{57}$, $\frac{g_4}{h_4} = \frac{16}{83}$, and $\frac{g_5}{h_5} = \frac{123}{638}$.

Following properties can be derived:

Lemma 3.1 ([2])

1. $g_{k+1} = a_{k+1}g_k + g_{k-1}, h_{k+1} = a_{k+1}h_k + h_{k-1}$,

2. $g_{k+1}h_k - g_k h_{k+1} = (-1)^k$,

3. $\left|\beta - \frac{g_k}{h_k}\right| < \frac{1}{h_k h_{k+1}}$.

Since each convergent of β is a irreducible fraction in $[0,1]$, it corresponds to a direction vector. Let $v_k = (h_k, g_k)$ for $k \geq 0$. Let i be the largest integer such that $h_i \leq \min\Big\{\sqrt{\frac{l}{w}}, l\Big\}$. $\frac{g_i}{h_i}$ is the closest convergent of β with denominator less than or equal to $\min\Big\{\sqrt{\frac{l}{w}}, l\Big\}$. Vector v_i is then the direction we want. For example, if $\beta = \frac{123}{638}$, $l = 640$

and $w = 10$, the direction we want is $(5, 1)$. By Lemma 3.1, $i \le \log_2 \min\left\{\sqrt{\frac{l}{w}}, l\right\}$. Since w is a constant and $l \le \sqrt{2}U$, direction v_i can be found in time $O(\log U)$. It remains to be shown that v_i is a direction in which the number of parallel lines covering grid points in P is $O(\sqrt{U})$.

To estimate the number of parallel lines in direction v needed to cover grid points in P, we need to know the range to be covered and the minimum distance between two adjacent lines with respect to direction v. Let θ_v denote the angle between line passing through diametral pair and vector v. The distance of the range to cover grid points in P is $l\sin\theta_v + w\cos\theta_v$. The minimum distance between two adjacent lines of parallel lines in direction v to cover grid points in a plane is shown in following lemma:

Lemma 3.2 *Minimum distance between two adjacent lines of parallel lines in direction v to cover grid points in a plane is* $\frac{1}{\|v\|}$.

Proof. Let L_1 and L_2 be two adjacent lines with direction v such that there are no grid points between these two lines. Let u be a vector from one grid point on L_1 to one grid point on L_2. $\{u, v\}$ is then a basis of lattice $\mathbb{Z}^2$; the area of the fundamental parallelogram formed by u and v is 1 [2]. The distance between L_1 and L_2 is therefore $\frac{1}{\|v\|}$.

Let $L(v) = (l\sin\theta_v + w\cos\theta_v)/\left(\frac{1}{\|v\|}\right)$. The number of parallel lines in direction v to cover gird points in P is less than $L(v)$. □

The following lemma can be derived.

Lemma 3.3 *If* $\sqrt{\frac{l}{w}} < l$ *then* $L(v_i) < 2\sqrt{2}\sqrt{lw}$*; otherwise,* $L(v_i) < 2\sqrt{2}$.

Proof. Consider the following two cases:

1. v_i is the last convergent of β:
   That is, $\frac{g_i}{h_i} = \beta$. Then
   $$\begin{aligned} L(v_i) &\le (l\sin\theta_{v_i} + w)\|v_i\| = w\|v_i\| \\ &< \sqrt{2}wh_i \le \sqrt{2}\min\left\{\sqrt{\frac{l}{w}}, l\right\}w \end{aligned}$$
   If $\sqrt{\frac{l}{w}} \le l$ then $L(v_i) < \sqrt{2}\sqrt{lw}$; otherwise, $L(v_i) < \sqrt{2}lw < \sqrt{2}$.

2. v_i is not the last convergent of β:
   Let $\theta_{v_i,v_{i+1}}$ denote angle between v_i and v_{i+1}. Since β is between $\frac{g_i}{h_i}$ and $\frac{g_{i+1}}{h_{i+1}}$, $\theta_{v_i} \le \theta_{v_i,v_{i+1}}$. Also, $\sin\theta_{v_i,v_{i+1}} = \frac{\|v_i \times v_{i+1}\|}{\|v_i\|.\|v_{i+1}\|}$ if these vectors are treated as

three dimensional vectors, where $\times$ denote the cross product. Since $\|v_i \times v_{i+1}\| = |g_{i+1}h_i - g_i h_{i+1}| = 1$, $\sin\theta_{v_i,v_{i+1}} = \dfrac{1}{\|v_i\|.\|v_{i+1}\|}$. Then,

$$\begin{aligned} L(v_i) &\leq (l\sin\theta_{v_i} + w)\|v_i\| \\ &\leq \Big(\frac{l}{\|v_i\|.\|v_{i+1}\|} + w\Big)\|v_i\| \\ &= \frac{l}{\|v_{i+1}\|} + w\|v_i\| \\ &< \frac{l}{h_{i+1}} + w\sqrt{2}h_i \\ &< \frac{l}{\min\Big\{\sqrt{\frac{l}{w}}, l\Big\}} + w\sqrt{2}\min\Big\{\sqrt{\frac{l}{w}}, l\Big\}. \end{aligned}$$

If $\sqrt{\dfrac{l}{w}} < l$ then $L(v_i) < \sqrt{lw} + \sqrt{2}\sqrt{lw} < 2\sqrt{2}\sqrt{lw}$; otherwise, $L(v_i) < 1 + \sqrt{2}lw < 2\sqrt{2}$. □

Since w is a constant and $l \leq \sqrt{2}U$, $L(v_i)$ is $O(\sqrt{U})$ in worst case. That is, the number of parallel lines in direction v_i to cover grid points in P of constant width is $O(\sqrt{U})$ in the worst case.

The line-covering algorithm works as follows. At first, the pruning step is performed repeatedly until the width of the polygon is less than $16\sqrt{2}$. At this time, the diameter of the polygon is $O(U)$ in the worst case; therefore there are $O(U)$ grid points in the polygon in the worst case. Each pruning step takes time linear in the number of polygon's vertices. In addition, an extra time, $O(F)$, must be paid to evaluate the function value of the "central" grid point in each pruning step, such that the optimal solution obtained up to now can be kept track of. The pruning step can be repeated at most $O(\log U)$ times. Hence, the total time in this stage is $O(n\log U + \log^2 U + F\log U)$. Secondly, convergents of the slope of the line through diametral pair are computed. One of these convergents corresponding to v_i as stated above represents the direction we need. With this direction, only $O(\sqrt{U})$ parallel lines are needed to cover grid points in the polygon. The interval of each parallel line, in which grid points on the line are inside the polygon, can be determined in time linear in the number of vertices – $O(n + \log U)$. There are at most $O(U)$ grid points in the interval; optimal solution among these grid points can therefore be found in $O(F\log U)$ time by binary search. Since there are $O(\sqrt{U})$ parallel lines, time spent in this stage is $O(n + \log U + F\sqrt{U}\log U)$. Hence total time of line-covering algorithm is $O(n\log U + F\sqrt{U}\log U)$. Details of the algorithm is stated in the following.

Algorithm: Line-covering algorithm.

Step 1: Repeat step 1.1 through step 1.3 until the width of the polygon is less than $16\sqrt{2}$.

Step 1.1: Find the "central" grid point of the polygon.

Step 1.2: Compute the function value of the "central" grid point. If the value is less than the value computed up to now, replace the optimal solution with this point.

Step 1.3: Determine the tangent line through the "central" grid point. Set new polygon to be searched to the subpolygon on the right side of the tangent line if the gradient is toward the left and vice-versa.

Step 2: Let β be the slope of the line through the diametral pair of the polygon. Let l and w denote the diameter and width of the polygon. Do step 2.1 through step 2.2.

Step 2.1: Compute the convergent $\frac{g_i}{h_i}$ of β where i is the largest integer such that $h_i \leq \min\left\{\sqrt{\frac{l}{w}}, l\right\}$. Then $S = \{h_i y - g_i x = c, c \in Z\}$ is the set of parallel lines we want.

Step 2.2: Without loss of generality, assume the polygon is above the line $h_i y - g_i x = 0$. Let d denote the minimum distance from vertices of the polygon to line $h_i y - g_i x = 0$. Then $h_i y - g_i x = \lceil d \rceil$ is the lowest line in S intersecting with the polygon.

For $c = \lceil d \rceil, \lceil d \rceil + 1, \cdots$ **do** :

If the line $h_i y - g_i x = c$ intersects with the polygon then compute the intersection points; otherwise stop. Let the q_l and q_r denote the intersection points. Let $\{p_1, p_2, \ldots, p_j\}$ be the set of grid points between q_l and q_r on line $h_i y - g_i x = c$. Perform binary search on the set $\{p_1, p_2, \ldots, p_j\}$. If the optimal solution on the line is better than the solution found up to now, update the solution. Note that the set $\{p_1, p_2, \ldots, p_j\}$ can be easily obtained once the coordinates of one grid point on line $h_i y - g_i x = c$ is known. For $c \neq 0$, one grid point on line $h_i y - g_i x = c$ is (ch_{i-1}, cg_{i-1}) if $\frac{g_i}{h_i} < \frac{g_{i-1}}{h_{i-1}}$, or $(-ch_{i-1}, -cg_{i-1})$ if $\frac{g_i}{h_i} > \frac{g_{i-1}}{h_{i-1}}$.

4 The transformation

In this section a transformation based on the idea of line-covering algorithm is introduced to improve the line-covering algorithm so that the optimization problem can be solved in $O((n+F)\log U + log^2 U)$ time. Recall that difficulties will occur when the polygon is narrow; we, therefore, try to widen the polygon by the transformation proposed. After the polygon is transformed, the polygon would become wider. Two possibilities will occur: either the polygon becomes wide enough so that the pruning step can be continued, or the polygon is still narrow but it can be covered by a constant number of parallel lines. Hence, the problem can be solved in a more sophisticated way: the pruning step and transformation are applied alternately until the transformed polygon is narrow; the line-covering approach is then applied. The major difference between the line-covering algorithm in previous section and the method here is that only constant number of parallel lines are needed in line-covering phase of the method here.

To be a possible candidate, a transformation must meet following requirements: it is an area-preserved transformation; image of a grid point is also a grid point and vice versa; images of collinear grid points are also collinear. It means that each components of the transformation matrix must be integers and the determinant of the matrix is ± 1. In other words, the basis of the new coordinate system is formed by two integer vectors and the determinant of vectors is ± 1. By Lemma 3.1,

$$\begin{vmatrix} h_k & g_k \\ h_{k+1} & g_{k+1} \end{vmatrix} = (-1)^k.$$

That is, each pair of two consecutive convergents corresponds to a possible basis of new coordinate system. The problem is which one satisfies our requirements – after the polygon is transformed, either the polygon would become wider so that the pruning step can be continued, or grid points in the polygon can be covered by a constant number of parallel lines.

Let P' be the image of P under a transformation whose matrix is composed of integers and determinant is ± 1. Let rectangle *abcd* be the smallest rectangle containing P' with sides parallel to axes. Suppose we can show that the area of rectangle *abcd* is roughly equal to that of P'. As a consequence, the transformation is what we want. The argument is as follows. Let l' and w' denote the diameter and width of P'. Then $l' \times w'$ is roughly equal to the area of P'. Let s_l and s_w denote the lengths of longer sides and shorter sides of rectangle *abcd*. Since area of P' is roughly equal to that of rectangle *abcd*, we have

$$l' \times w' \approx s_l \times s_w.$$

Since $l' \approx s_l$, $s_w \approx w'$. That is, the length of shorter sides of rectangle *abcd* is roughly equal to the width of P'. Assume the shorter sides of rectangle *abcd* are parallel with Y axis. If the width of P' is a constant less than $16\sqrt{2}$, then grid points in P' can be covered by a constant number of lines parallel with X axis; otherwise, P' can be processed by the pruning step.

Recall that in line-covering algorithm we use v_i as the direction of the set of parallel lines to cover grid points in P. Here v_i is used as the direction of X axis of the new coordinate system. That is, v_i is one of the basis vectors of the new coordinate system. The other vector of the basis is therefore either v_{i-1} or v_{i+1} determined by the following criterion: if $\frac{g_i}{h_i} = \beta$ or $h_i h_{i+1} > \frac{l}{w}$ then it is v_{i-1}; otherwise it is v_{i+1}. In other words, if $\dfrac{g_i}{h_i} = \beta$ or $h_i h_{i+1} > \dfrac{l}{w}$ the basis is $\{v_i, v_{i-1}\}$; otherwise the basis is $\{v_i, v_{i+1}\}$. Let A denote the matrix of the transformation using $\{v_i, v_{i-1}\}$ or $\{v_i, v_{i+1}\}$ as basis depending on whether the condition, $\dfrac{g_i}{h_i} = \beta$ or $h_i h_{i+1} > \dfrac{l}{w}$, is true or not. Then

$$A = \begin{pmatrix} (-1)^i g_{i-1} & (-1)^{i+1} h_{i-1} \\ (-1)^{i+1} g_i & (-1)^i h_i \end{pmatrix} \text{ or } \begin{pmatrix} (-1)^i g_{i+1} & (-1)^{i+1} h_{i+1} \\ (-1)^{i+1} g_i & (-1)^i h_i \end{pmatrix}.$$

The image of p is given by $A \cdot p$. Since A is a linear transformation, it satisfies the third precondition of candidacy of a transformation. By Lemma 3.1, determinant of A is equal to ± 1. Moreover, components of A are all integers. Matrix A therefore satisfies the other two preconditions. If we can show that A satisfies the requirement in the following theorem, transformation A is what we want.

Theorem 4.1 *Let P' be the image of P under transformation A. Let Q denote the area of the smallest rectangle containing P' with sides parallel to axes. Then $Q \approx$* (area of P').

PROOF. The lengths of the sides of the smallest rectangle containing P' with sides parallel to axes are $L(v_i)$ and $L(v_{i+1})$ if $\{v_i, v_{i+1}\}$ is used as the basis of new coordinate system; otherwise, they are $L(v_i)$ and $L(v_{i-1})$. We want to prove that $L(v_i)L(v_{i-1}) < 6lw$ if $\frac{g_i}{h_i} = \beta$ or $h_i h_{i+1} > \frac{l}{w}$; otherwise, $L(v_i)L(v_{i+1}) < 6lw$. Note that $lw \approx$ area of P (= area of P').

By a similar argument in the proof of Lemma 3.3, we have

$$\begin{aligned} L(v_{i-1}) &< (l\sin\theta_{v_{i-1},v_i} + w)\|v_{i-1}\| \\ &< \frac{l}{h_i} + \sqrt{2}wh_{i-1}. \end{aligned}$$

If $\frac{g_i}{h_i} = \beta$ then

$$L(v_i) < \sqrt{2}wh_i;$$

otherwise,

$$\begin{aligned} L(v_i) &< (l\sin\theta_{v_i,v_{i+1}} + w)\|v_i\| \\ &< \frac{l}{h_{i+1}} + \sqrt{2}wh_i. \end{aligned}$$

Similarly, if v_{i+1} is the last convergent, then

$$L(v_{i+1}) = w\|v_{i+1}\| < \sqrt{2}wh_{i+1};$$

otherwise,

$$\begin{aligned} L(v_{i+1}) &< (l\sin\theta_{v_{i+1},v_{i+2}} + w)\|v_{i+1}\| \\ &< \frac{l}{h_{i+2}} + \sqrt{2}wh_{i+1} \end{aligned}$$

Hence, if v_{i+1} is the last convergent then

$$L(v_i)L(v_{i+1}) < \sqrt{2}lw + 2w^2 h_i h_{i+1};$$

otherwise,

$$\begin{aligned} L(v_i)L(v_{i+1}) &< \frac{l^2}{h_{i+1}h_{i+2}} + \sqrt{2}lw + \sqrt{2}lw\frac{h_i}{h_{i+2}} + 2w^2 h_i h_{i+1} \\ &< 4lw + 2w^2 h_i h_{i+1} \end{aligned}$$

In other words,

$$L(v_i)L(v_{i+1}) < 4lw + 2w^2 h_i h_{i+1}.$$

If $\frac{g_i}{h_i} = \beta$,

$$L(v_i)L(v_{i-1}) < \sqrt{2}lw + 2w^2 h_i h_{i-1} < 4lw.$$

If $\frac{g_i}{h_i} \neq \beta$ and $h_i h_{i+1} > \frac{l}{w}$,

$$\begin{aligned} & L(v_i)L(v_{i-1}) \\ & \quad < \frac{l^2}{h_i h_{i+1}} + \sqrt{2}\frac{h_{i-1}}{h_{i+1}}lw + \sqrt{2}lw + 2w^2 h_i h_{i-1} \\ & \quad < \frac{l^2}{h_i h_{i+1}} + 5lw < 6lw. \end{aligned}$$

If $\frac{g_i}{h_i} \neq \beta$ and $h_i h_{i+1} \leq \frac{l}{w}$,

$$L(v_i)L(v_{i+1}) < 4lw + 2w^2 h_i h_{i+1} \leq 6lw. \qquad \square$$

The overheads of the transformation is little. The transformation matrix can be found in $O(\log \min\{\sqrt{\frac{l}{w}}, l\})$ time. Since $l \leq \sqrt{2}U$, the matrix can be found in $O(\log U)$ time. Each polygon can be transformed in time linear in its number of vertices. A complete algorithm and time complexity analysis are given in the following section.

5 The algorithm and complexity

Following the discussions in previous section, we have an improved algorithm for the optimization problem. The algorithm works as follows. At first stage, the pruning step is repeatedly applied until width of the polygon is less than $16\sqrt{2}$. At second stage, the transformation is used to widen the polygon. If the width of the transformed polygon is greater than or equal to $16\sqrt{2}$, then proceed to the first stage; otherwise, go to final stage. In the final stage, grid points in the polygon can be covered by a constant number of horizontal or vertical parallel lines. Binary search can then be used on these parallel lines to find optimal solution in the polygon. Details of the algorithm is given in the following. Here we assume that current minimum is kept in $CURMIN$ and the mapping between original coordinate system and current coordinate system is T. Initially, $CURMIN$ is set to any grid point in the polygon and T is set to an identity mapping. Needs of T arise from that each computation of function value is done in the original coordinate system. For each point p in current coordinate system, its function value is $f(T^{-1}(p))$.

Algorithm: An improved line-covering algorithm.

Step 1: Repeat step 1.1 through step 1.3 until the width of the polygon is less than $16\sqrt{2}$.

Step 1.1: Find the "central" grid point of the polygon. Let it be p_c.

Step 1.2: Compute the function value of the "central" grid point $f(T^{-1}(p_c))$. If the value is less than $f(CURMIN)$, replace $CURMIN$ with $T^{-1}(p_c)$.

Step 1.3: Determine the tangent line through the "central" grid point. Set new polygon to be searched to the subpolygon on the right side of the tangent line if the gradient is toward the left and vice-versa.

Step 2: Compute the transformation matrix A. Let T_A denote the corresponding transformation. Transform the polygon by T_A. Update T with the composite of T and $T_A - T \circ T_A$. If the width of the transformed polygon is greater than $16\sqrt{2}$ then go to step 1; otherwise, go to step 3.

Step 3: Assume the shorter sides of the smallest rectangle containing the polygon with sides parallel to axes are parallel to Y axis. Let d denote the Y coordinate of the lowest vertex of the polygon.

For $c = \lceil d \rceil, \lceil d \rceil + 1, \cdots$ **do** :
If the line $y = c$ intersects with the polygon then compute the intersection points; otherwise stop. Let the q_l and q_r denote the intersection points. Let $\{p_1, p_2, \ldots, p_j\}$ be the set of grid points between q_l and q_r on line $y = c$. Perform binary search on the set $\{p_1, p_2, \ldots, p_j\}$. If the optimal solution on the line is better than $CURMIN$, update $CURMIN$.

In each iteration, step 1 takes the time linear in the number of the polygon's vertices plus the time to evaluate objective function, and step 2 takes $O(\log U)$ time to find the transformation matrix and the time (linear in the number of the polygon's vertices) to transform the polygon. Since step 1 and 2 can be repeated at most $O(\log U)$ times, total time spent in step 1 and 2 is $O((n + F)\log U + \log^2 U)$. In step 3, the time to compute the intersection points is $O(n + \log U)$ and the time to perform binary search is $O(F \log U)$. Since the loop in step 3 can be repeated at most constant times, the time spent in step 3 is $O(n + \log U + F \log U)$. Total time of the algorithm is therefore $O((n + F)\log U + \log^2 U)$.

We can conclude above discussions by the following theorem.

Theorem 5.1 *A two dimensional convex integer programming problem with n linear constraints can be solved in $O((n + F)\log U + \log^2 U)$ time, if the polygon formed by the constraints is contained in a $U \times U$ lattice and the time to evaluate the objective function is F. The restricted Weber problem is therefore can be solved in time $O((n + m)\log U + \log^2 U)$.*

6 Concluding remarks

In this paper, we have solved the Weber problem on a grid efficiently. We also have proposed a general approach for a more general optimization problem: Given a differentiable convex objective function f defined on $\mathbb{R}^2$ and a convex polygon P formed by n

linear constraints, find a integer point in P such that f is minimized. A transformation is proposed to widen the polygon so that either a "central" grid point can be found or the optimal solution can be easily found when the polygon is narrow. Such a transformation can be easily computed by the method called continued fraction expansions. By repeating the pruning step and the transformation, the general optimization problem can be solved efficiently.

Our algorithm provides a good integer approximation for the two dimensional convex integer programming problem in a sense that the integer solution found is the best among all integer solutions. The most time consuming step of our algorithm is to find the diameter of the polygon and determine the transformation. If the diameter of a changing polygon can be found more efficiently (in stead of linear time) and the transformation can be determined faster, complexity of the algorithm can be reduced. An extension of our algorithm to problems of higher dimensions is also a challenging topic.

REFERENCES

[1] D. E. Knuth. The art of computer programming. Addison-Wesley, second edition, 1981.

[2] L. Lovasz. An algorithmic theory of numbers, graphs and complexity. Society for Industrial and Applied Mathematics, Pennsylvania, 1986.

[3] Z. A. Melzak. A companion to concrete mathematics. Wiley Interscience, New York, 1973.

[4] N. Megiddo, K. J. Supowit. On the complexity of some common geometric location problems. *SIAM J. Comput.* **13** (1984), 182-196.

[5] S. Mehta, M. Mukherjee, G. Nagy. Constrained integer approximation to 2-D line intersections. Second Canadian Conf. on Computational Geometry, Ottawa, Canada, August 1990, 302-305.

[6] C. H. Papadimitriou, K. Steiglitz. Combinatorial optimization: algorithms and complexity. Prentice-Hall, New Jersey, 1982.

[7] F. P. Preparata, M. I. Shamos. Computational geometry: An introduction. Springer-Verlag, New York, 1985.

[8] A. Weber. Uber den Standort der Industrien. Erster Teil: Reine Theorie dis Standorts. Mit einem mathematischen Anhang von Georg Pick. J.C.B. Mohr (Paul Siebeck), Tubingen, 1909.

Department of Shipping and Transportation Management
National Taiwan Ocean University
Keelung, 200, Taiwan
Republic of China