An Improved Implementation and Analysis of the Diaz and O'Rourke Algorithm for Finding the Simpson Point of a Convex Polygon

By: J. Bhadury, C.A. Tovey

Bhadury, J., & Tovey, C.A. (2010). An improved implementation and analysis of the Diaz and O'Rourke algorithm for finding the Simpson point of a convex polygon. *International Journal of Computer Mathematics, 87(2)*, 244-259. doi: 10.1080/00207160801965180

Abstract:

This paper focuses on the well-known Diaz and O'Rourke [M. Diaz and J. O'Rourke, *Algorithms for computing the center of area of a convex polygon*, Visual Comput. 10 (1994), 432–442.] iterative search algorithm to find the Simpson Point of a market, described by a convex polygon. In their paper, they observed that their algorithm did not appear to converge pointwise, and therefore, modified it to do so. We first present an enhancement of their algorithm that improves its time complexity from $O(\log^2 \epsilon)$ to $O(n \log 1/\epsilon)$. This is then followed by a proof of pointwise convergence and derivation of explicit bounds on convergence rates of our algorithm. It is also shown that with an appropriate interpretation, our convergence results extend to all similar iterative search algorithms to find the Simpson Point – a class that includes the original unmodified Diaz–O'Rourke algorithm. Finally, we explore how our algorithm and its convergence guarantees might be modified to find the Simpson Point when the demand distribution is non-uniform.

Keywords: competitive location | geometric optimisation | Simpson Point | centre of area | iterative algorithms

Article:

1. Introduction

The problem of finding the *Simpson Point*, also referred to as the (1|1)-*Centroid*, belongs to the area of competitive location theory, whose models have been studied extensively since the days of Hotelling 16 with applications in fields as diverse as economics, marketing, voting theory, operations research and regional science. Recent comprehensive surveys of the models in this area can be found in 6 9 19. A seminal model in this area is that of Hakimi 12 who assumes the following scenario: two competing firms vie for a common market by locating their own facilities to sell an identical product. Of the two competing firms, one is designated as the *leader* that decides to locate *p* of its own facilities first. The leader is aware of the fact that after it has entered the market with *p* facilities, the rival firm, denoted as the *follower*, will locate *r* of its own facilities in such a manner as to take away as much market share from the leader as possible. Given this, the decision problem facing the leader is to find optimal locations for its own *p*facilities such that the maximum market share that is lost to the follower is as small as possible – Hakimi refers to this as the ($r|p$)-*centroid* problem. In general, when $r, p \geq 2$, these problems are complex, for example, Hakimi 12 has shown that when the market is given by a network, with the customers located at the nodes, the ($1|p$)-*centroid*problem, or even computing an approximate solution for it, is NP-hard. In the context of a voting theory, the (1|1)-*centroid* problem is studied under a different name, namely, finding the Simpson solution 21 to a voting game – it is this name that we will use throughout the paper. The Simpson solution is the point in policy space against which the fewest voters can be mustered. This special case has been extensively studied in the literature. When the market is given by a network, with the customers located at the nodes, Hansen and Labbé 14 have given a polynomial algorithm to determine the Simpson point. The case where the market is given by a set of discrete points, which represent the customers, has been studied by Carrizosa *et al*. 2 3, Durier 8, Drezner 5, Drezner and Zemel 7, Tovey23 and Michelot 18. The case where the market is given by a polygon with a continuous distribution of customers has also received attention. When the demand distribution is uniform, the Simpson Point of a convex polygon is identical to the Center of Area of that polygon. This problem has the flavour of computational geometry. The literature of geometry has several classical results on this case - see for example, 1 10 11 15 17 22 24. Our point of departure in this paper is the work done by Diaz and O'Rourke 4, where they present simple, iterative search algorithms to compute the Center of Area of a convex polygon.

We consider a market area described by a convex polygon and explain the problem equivalences when the demand is uniformly distributed over the market area. The basic iterative algorithm enunciated by Diaz and O'Rourke 4 for this problem computes smaller and smaller convex sets inside the polygon that are guaranteed to contain the centre of the area. Diaz and O'Rourke 4 observed that the convex sets could fail to converge to a point and proposed a modification that converges pointwise. We present an enhancement of their algorithm that significantly reduces its time complexity from $O(n \log 1/\varepsilon + \log^2 1/\varepsilon)$ to $O(n \log 1/\varepsilon)$. We also derive explicit bounds on pointwise convergence of our or any similar algorithm. In doing so, we also show that Diaz and O'Rourke's 4 original unmodified algorithm actually does produce a series of points that converges to the Simpson point at a guaranteed convergence rate. In addition, we show that its convex sets fail to converge to a point only in an easily identifiable and exactly solvable special case. Finally, we explore how our modification might be extended to find the Simpson Point of the convex polygon when the demand distribution is non-uniform and derive a convergence guarantee, albeit weaker than pointwise – to the best of our knowledge, there is no discussion in the literature of this version of the (1|1)-*centroid*problem.

The remainder of the paper is divided as follows. The next section presents the notation used in the paper and the basic results that are to be used later. This is followed by Section 3, which discusses the enhancement to the basic iterative algorithm. The last portion of that section then discusses how our iterative algorithm can be modified when the demand distribution is non-uniform. Section 4 gives the pointwise convergence analysis and the convergence analysis for the non-uniform case.

2. Preliminaries – notation and basic results

In this section, we present definitions, notation, and some basic results. There are two firms competing for a common market area, by locating one facility each that sells an identical product (with respect to price, quality, etc.) to the customers. The market area itself is given by a convex polygon *P* with an inelastic demand uniformly distributed over it. Relocation of the facilities is assumed to be prohibitively expensive in the short run, thus presupposing that they are similar to heavy industries such as factories, stores, etc. One of the two firms, the *leader*, has decided to locate its facility first. The leader is aware that after it has entered the market, its competitor, which we refer to as the *follower*firm, will then locate its own facility with the aim of capturing as much market share from the leader as possible. The decision problem facing the leader is to locate its facility so as to minimize the market area that would be lost to the follower, assuming that the follower maximizes its market share – the point in the market that achieves this is referred to as the Simpson point or (1|1)-*centroid* of *P* and we denote it by x^*. The cost of transportation is assumed to be strictly increasing with distance (given as per the Euclidean norm), customers therefore patronize the closest facility, with all ties being broken in favour of the original entrant into the market, i.e. the leader.

The bounded *P* has *n* vertices $v_1, v_2, \ldots, v_n$, where the numbering is given in the clockwise order. The edges of *P*are thus given by the line segments $\overline{v_1v_2}, \overline{v_2v_3}, \ldots\ldots, \overline{v_{n-1}v_n}$, whose union constitutes the boundary of *P*, which is denoted by *bd*(*P*). The interior of *P*, which we assume is non-empty, is denoted by *In*(*P*). The line defined by two distinct points *x* and *y* will be denoted $\overleftrightarrow{xy}$ and the halfspaces defined by that line will be denoted $\overleftrightarrow{xy}^+$ and $\overleftrightarrow{xy}^-$. Which halfspace is which will be defined as needed. In general, we employ the convention that '-' faces inwards towards the centre of *P*. Consider any two distinct points $y, z \in bd(P)$. If the line segment $\overline{yz}$ intersects the interior of *P*, it is referred to as a cut of *P*, since it divides *P* into two smaller convex polygons. In order to define these two smaller polygons, *left-arc*[*y*, *z*] (respectively, *right-arc*[*y*, *z*]) is defined as the section of *bd*(*P*) that consists of all points on*bd*(*P*) that are encountered in a clockwise (respectively, counterclockwise) traversal of *bd*(*P*) from *y* to *z*, including these two points themselves. Given this, the convex polygon whose boundary is the union of $\overline{yz}$ and left-arc [*y*, *z*] (respectively, right-arc [*y*, *z*]) is denoted as Left[*y*,

z] (respectively, Right [y, z]). For example, in Figure 1, Left[v_7, v_2] (respectively, Right[v_7, v_2]) has vertices v_7, v_1 and v_2 (respectively, v_2, v_3, v_4, v_5, v_6 and v_7).

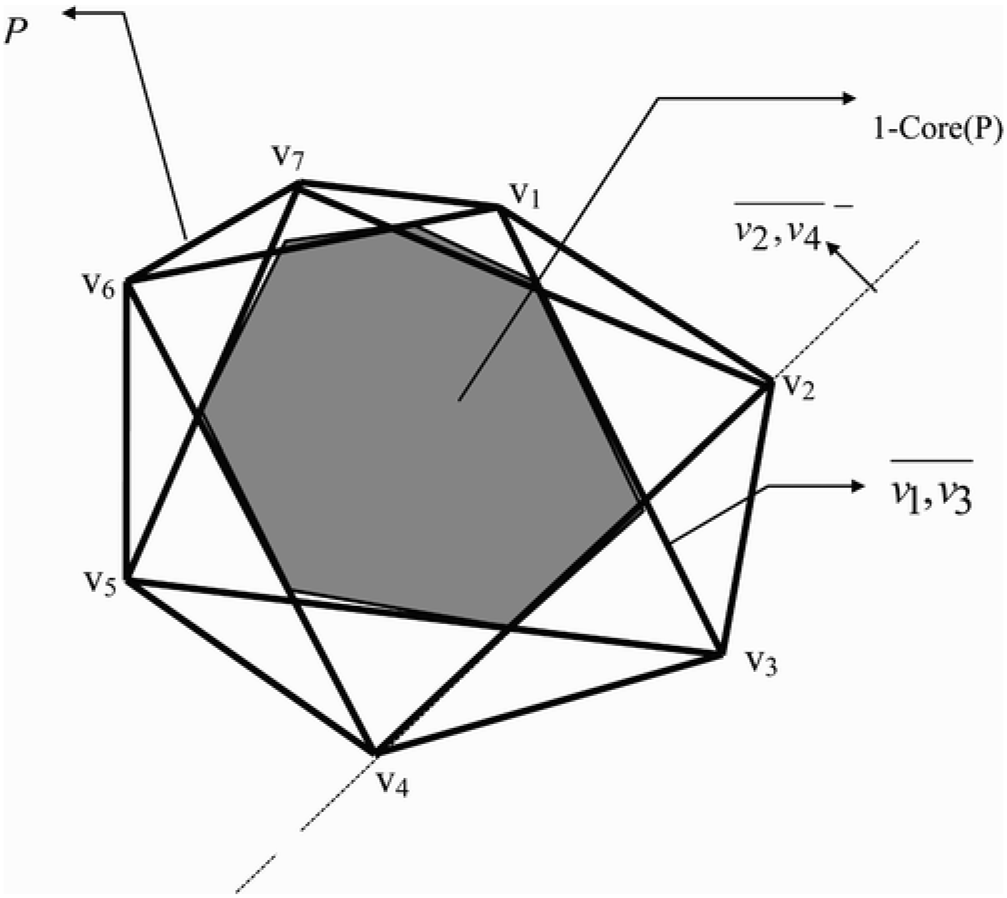

Figure 1. Convex polygon *P*.

The area of any polygon *S* will be denoted by *A*(*S*). Since the demand distribution is uniform, the demand generated in *S* is assumed to be proportional to *A*(*S*). The term 'larger of two given polygons' will refer to the one with the greater area. Consider now a point $x \in P$ and any cut $\overline{yz}$ of *P* that passes through *x*. The point *z* is determined by *P*, *x*and *y* and is denoted *z*(*x*, *y*, *P*). This set of all such cuts $\overline{yz(x, y, P)}$ is parameterized by *y* as *y* ranges over the set*bd*(*P*). Let *y*' denote the value of *y* in *bd*(*P*) that maximizes $A(\mathrm{Left}[y, z(x, y, P)])$ (note that the maximum is attained because *bd*(*P*) is compact and the area function is continuous). Define $H(x) = A(\mathrm{Left}[y', z(x, y', P)])$. Thus, *H*(*x*) denotes the larger of the two most unequal (in terms of area) pair of smaller polygons that any cut through *x* can partition *P* into. (Along similar lines, when the demand distribution is non-uniform, *H*(*x*) will be defined by considering all the sets that any cut through *x* can partition *P* into, and choosing the one with the maximum demand.)

With $H(x)$ defined as above, then following Hotelling's 16 ‘Principle of Minimum Differentiation’, if the leader locates at the point $x \in P$, the follower can be expected to locate at a distinct but arbitrarily close point q in $H(x)$ such that $\overleftrightarrow{qx}$ is normal to $\overleftrightarrow{y'z(x, y', P)}$. Hence, the leader must be prepared to lose the entire set $In(H(x))$ to the follower and be left with the market region $P \setminus In(H(x))$. Since $A(bd(H(x))) = 0$, an alternate definition of the Simpson pointx^* could be

$$x^* = \mathbf{argmin}_{x \in P}\{A(H(x))\}.$$

When defined as above, the Simpson point x^* is referred to as the *Centre of Area* of P (as distinct from the centre of mass) in the geometry literature. Diaz and O'Rourke 4 study the centre of area problem for a convex polygon and begin by giving an $O(n)$ time algorithm to compute $H(x)$ for any given point $x \in P$. They also give an $O(n^6)$ combinatorial algorithm to compute x^* – however, given its procedural and time complexity, they also present an iterative search algorithm, it is this latter algorithm that is the basis of our paper. We need three facts to describe their algorithm. The first two lemmas are immediate; the third is a classical result.

Lemma 2.1

4 *For any convex polygon P with* $A(P)>0$, $x^* \in In(P)$.

Lemma 2.2

4 *For any convex polygon P*, $x^* \in H(x) \forall x \in P$.

Lemma 2.3

10 13 *Given a polytope P ’ in d-dimensions with a volume of unity, let x denote its centroid* (*centre of mass*). *Then any hyperplane through x divides P ’ into two polytopes, the larger of which has a volume of no more than* $[1 - (d/(d+1))^d]$.

In terms of our function $H()$, Lemma 2.3 states that if $y \in P$ is the centroid of polygon P, then $A(H(y)) \leq 5A(P)/9$. Based on these facts, the basic iterative algorithm proposed by Diaz and O'Rourke 4 proceeds as follows: at the first step, choose an $x^{(1)}$ as the centroid of P and calculate $H(x^{(1)})$. By Lemma 2.2, $x^* \in H(x^{(1)})$; hence, denote the*Simpson_Polygon* at step one, $S^{(1)}$, by $H(x^{(1)})$. In the second step, choose a point $x^{(2)}$ as the centroid of $S^{(1)}$ and find $H(x^{(2)})$. Now we know that $x^* \in \{H((x^{(1)}) \cap H((x^{(2)})\} = \{S^{(1)} \cap H(x^{(2)})\}$. Thus, at the end of step two, update $S^{(2)}$ to $\{S^{(1)} \cap H(x^{(2)})\}$. By repeating this procedure, we get successively smaller *Simpson_Polygons*that are guaranteed to contain x^*. Assuming that the termination criterion is to produce a *Simpson_Polygon* that has an area no more than ϵA, where ϵ is the required error bound, this idea is summarized as the following algorithm.

Algorithm *Find_Simpson* _I(from 4) **Begin Step 1:** *i* = =0. Set $S^{(i)}=P$. **Step 2:** While $(A(S^{(i)}) > \varepsilon A(P))$ do { $i = i+1$. Choose $x^{(i)}$ *as the centroid of* $S^{(i-1)}$. Compute $H(x^{(i)})$. Set $S^{(i)} = \{S^{(i-1)} \cap H(x^{(i)})\}$ } **End**

As for the performance of algorithm *Find_Simpson* _I of Diaz and O'Rourke, note that Lemma 2.3 guarantees that at least 4/9ths of the area of the *Simpson_Polygon* is removed at any iteration *i*. Hence, the area of $S^{(i)}$ is given as

$$A(S^{(i+1)}) \leq 5A(S^{(i)})/9 \quad \text{for} \quad i \geq 1, \quad \text{with} \quad A(S^{(1)}) \leq 5A(P)/9.$$

Therefore, in order to produce a *Simpson_Polygon* with area $\varepsilon A(P)$ or less, at most $O(\log 1/\varepsilon)$ iterations of the algorithm are needed. However, the number of vertices of the *Simpson_ Polygon* can increase by one on every iteration; hence the total time taken by algorithm *Find_Simpson* _I to produce a *Simpson_Polygon* of area no more than $\varepsilon A(P)$ is $O(\log^2 1/\varepsilon + n \log 1/\varepsilon)$. A second problem with algorithm *Find_Simpson* _I is that, as noted by Diaz and O'Rourke, it can fail to exhibit pointwise convergence, i.e. converge in diameter as well as area. As mentioned previously, Diaz and O'Rourke have demonstrated that pointwise convergence can be guaranteed asymptotically by modifying the implementation of the algorithm. We will show in Section 4 that the unmodified algorithm, appropriately interpreted, does converge pointwise even though the diameter may not converge to zero.

3. An improved version of algorithm *Find_Simpson _I*

In this section, we discuss an enhancement of algorithm *Find_Simpson* _I that reduces its time complexity. This improvement is brought about by controlling the number of vertices of *Simpson_Polygon*. To facilitate this enhancement, we first need the concept of a *k*-Core of a convex polygon.

3.1. k-Core of P

For every given value of *k*, where *k* is any positive integer less than $n-2$, we define a convex polygonal subset of *P*called the *k-Core of P*, and designated as *k*-Core(*P*), as follows. For each $i=1\ldots n$ let $\{i+k+1\}$ denote $(i+k) \bmod (n)+1$. Then define

$$k-Core(P) = \bigcap_{i=1}^{n} \text{Right}[v_i, v_{\{i+k+1\}}].$$

See Figure 1 for an example of the 1-Core of a given convex polygon. Note that *k*-Core(*P*) may be empty. Regardless of the time required to compute, it can be obtained as follows. Let $\overleftrightarrow{v_i v_{\{i+k+1\}}}^-$ denote the closed halfspace defined by the line through v_i and $v_{\{i+k+1\}}$ that contains the polygon $\text{Right}[v_i, v_{\{i+k+1\}}]$. Then the *k*-Core(*P*) could also have been defined as

$$k - \mathrm{Core}(P) = \bigcap_{i=1}^{n} \overleftrightarrow{v_i v_{\{i+k+1\}}}^{-}$$

Thus finding the *k*-Core(*P*) reduces is finding the intersection of *n* linear inequalities, which can be accomplished in $O(n\log n)$ time, a standard result from computational geometry (20). Therefore,

Lemma 3.1

Given a convex polygon P with n vertices, and an integer $k\geq 1$, *the k -Core*(*P*) *can be computed in* $O(n \log n)$ *time*.

We will want the *k*-Core to have a non-empty interior. The next lemma gives a sufficient condition.

Lemma 3.2

Given any convex polygon P with n vertices and an integer $k\geq 1$, *if* $n\geq 3k+3$, *then the k -Core of P has non-empty interior*.

Proof

For convenience let R(*i*) denote $\mathrm{Right}[v_i, v_{\{i+k+1\}}]$. For each *i* the polygon *R*(*i*) contains $n-k$ vertices of *P*. Therefore, for any three polygons R(*a*), R(*b*), R(*c*), the multiset of their vertices from *P* has cardinality $3(n-k)$. It follows from $n\geq 3k+3$ that $3(n-k) \geq 2n+3$. Since no vertex can have multiplicity more than three in the multiset (as there are only three polygons), there must exist at least three vertices with multiplicity 3, i.e. that are common to all three polygons. Since no three vertices of *P* can be collinear, $\mathrm{R}(a) \cap \mathrm{R}(b) \cap \mathrm{R}(c)$ must moreover have non-empty interior. Therefore, $In(\mathrm{R}(a)) \cap In(\mathrm{R}(b)) \cap In(\mathrm{R}(c))$ has non-empty interior. Since this is true for every triple *a, b, c*, by Helly's Theorem for finite sets of open convex sets,

$$\bigcap_{i=1}^{n} In(R(i)) \neq \Phi.$$

This intersection is non-empty and open and hence it has a non-empty interior.

We will now prove a geometric property of the *k*-Core that will be useful in containing the number of vertices in the*Simpson_Polygon* generated by algorithm *Find_Simpson* _I. Consider any point $x \in In(k\text{-Core}(P))$ and any cut $\overline{yz}$that passes through *x*.
Since $x \in In(k\text{-Core}(P)) \subseteq In(P)$, *y* and *z* must occur on different edges of *P*. Now consider

the open set obtained by omitting the two endpoints *y* and *z* from left-arc[*y*, *z*] – we claim there are at least*k*+1 vertices of *P* in this open set.

If, on the contrary, there were *k* or fewer vertices of *P* in the open halfspace $In(\overleftrightarrow{yz}^+)$, then for convenience of labeling those vertices $v_2, \ldots, v_m$ where $m \le k+1$, consider the cut $\overline{v_1, v_{k+1}}$ and the polygon R(1) (using the notation from the proof of Lemma 3.2) it defines. On the one hand, *y* is in the half-open interval $[v_1, v_2)$ and *z* is in the half-open interval $(v_m, v_{m+1}]$. Therefore *x*, which is on the segment $\overline{yz}$, is in $\overleftrightarrow{yz}^+$ and hence not in *In*(R(1)). (It could be on the boundary of *R* (1) if $y=v_1$ and $z=v_{k+1}$). On the other hand, by definition $k\text{-Core}(P) \subseteq \mathrm{R}(1)$. Hence $\mathrm{x} \notin In(k\text{-Core}(P))$, a contradiction.

Since there are at least *k*+1 vertices in this open segment that is obtained from left-arc[*y*, *z*], the polygon Right[*y*, *z*] will have at most $(n-(k+1)+2) = n-k+1$ vertices.
Since $\mathrm{Left}[y,z] = \mathrm{Right}[z,y]$, we have bounded the number of vertices of both polygons formed by any cut through any interior point.

Lemma 3.3

Given any convex polygon P with n vertices, for any k≥1, *any cut through any point in* $In(k\text{-}Core(P))$ *partitions P into two polygons, each of which has no more than* (*n*−*k*+1) *vertices*.

3.2. An enhancement of algorithm Find_Simpson_I

Using the concept of *k*-Core given previously, we can now embellish algorithm Find_Simpson_I so that its time complexity reduces to $O(n|\log\varepsilon|)$. In order to do so, we use Lemma 2.3 to state that.

Lemma 3.4

Consider any cut $\overline{yz}$ *of a convex polygon P with Simpson solution x**. *If* $A(\mathrm{Left}[y,z]) \le 4A(P)/9$ (*respectively*, $A(\mathrm{Right}[y,z]) \le 4A(P)/9$)
, *then* $x^* \in \mathrm{Right}[y,z]$ (*respectively*, $x^* \in \mathrm{Left}[y,z]$).

Proof

Suppose $A(\mathrm{Left}[y,z]) \le 4A(P)/9$ (refer to Figure 2).
Let $x \in \mathrm{Left}[y,z] \backslash \overline{yz} = P \backslash \mathrm{Right}[y,z]$ be arbitrary. Let $\overline{fg}$ denote the cut through *x* that is parallel to the cut $\overline{yz}$. Obviously, $\mathrm{Left}[f,g] \subset \mathrm{Left}[y,z]$, implying that $A(\mathrm{Left}[f,g]) < A(\mathrm{Left}[y,z]) \le 4A(P)/9$ whence $A\mathrm{Right}[f,g] > 5A(P)/9$. By definition of *H*(*x*), we have $A(H(x)) \ge A(\mathrm{Right}[f,g]) > 5A(P)/9$. But by Lemma 2.3, the centroid *w* of *P* is a better location than *x*, as $H(w) \le 5A(P)/9$, thereby proving the non-optimality of *x*. The complementary case is immediate because $\mathrm{Right}[y,z] = \mathrm{Left}[z,y]$.

Given Lemma 3.4, we now define *Initial Triangle(P)* as a triangle inside *P* that is constructed as follows (refer to Figure 2). Starting from vertex v_1 of *P*, proceed clockwise on *bd*(*P*) from v_1 until the point *p* is reached with the property that $A(\text{Left}[v_1, p]) = 4A(P)/9$. Proceeding further clockwise from *p*, denote by *q* and *r* two more points on the boundary of *P*, such that $A(\text{Left}[q, v_1] = A(\text{Left}[p, r]) = 4A(P)/9$. In a clockwise traversal of the boundary of *P* that begins at *p* and ends at v_1, *r* must be encountered strictly after *q*, otherwise *P* would contain three disjoint regions, $In(\text{Left}[v_1, p])$, $In(\text{Left}[p, r])$, and $In(\text{Left}[q, v_1])$, with total area 12*A*(*P*)/9. This implies, in turn, that the two line segments $\overline{v_1 q}$ and $\overline{pr}$ will intersect in the interior of *P* – denote their point of intersection as *s*. The *Initial Triangle* is defined to be the triangle (v_1, *p*, *s*). Note that the *Initial Triangle* (i) has an area no more than *A*(*P*)/9, (ii) can be computed in *O*(*n*) time, and (iii) is guaranteed, by Lemma 3.4, to contain *x**, since $A(\text{Left}[v_1, p] = A(\text{Left}[p, r]) = A(\text{Left}[q, v_1]) = 4A(P)/9$.

The second idea that we will use to modify algorithm *Find_Simpson* _I is the *k*-Core(*P*) with *k*=2 (any larger *k* would also work). Recall that at iteration *i* of the algorithm, the *Simpson_* Polygon is given by $S^{(i)}$. Let the number of vertices in $S^{(i)}$ be denoted by $n^{(i)}$. Then by Lemmas 3.2 and 3.3 for the case of the 2-Core($S^{(i)}$), we have

Corollary 3.5

If $n^{(i)} \geq 9$, *i.e*., *the Simpson_Polygon at iteration i has at least nine vertices, the* 2-*Core*($S^{(i)}$) *has an non-empty interior*. *Further, every point in the interior of this* 2-*Core has the property that any cut of P through this point divides* $S^{(i)}$ *into two smaller polygons, each of which has at most* ($n^{(i)}-1$) *vertices*.

Based on the ideas of the *Initial Triangle* and Corollary 3.5, the specialized version of algorithm *Find_Simpson* _I, is presented below.

***Find_Simpson_* II Begin 1** *i* = 1. Set $S^{(i)}$ = *Initial Triangle(P)* **2 while** $(A(S^{(i)}) > \varepsilon A(P))$ **do** { **2.1 (Vertex reduction loop)** If ($n^{(i)}>8$, i.e., $S^{(i)}$ has more than 8 vertices and therefore $n^{(i)}=9$) { **2.1.1** Find 2-*Core*($S^{(i)}$). Choose $x^{(i+1)}$ as any point in the interior of this 2-*Core*. **2.1.2** Find $H(x^{(i+1)})$. **2.1.3** Set $S^{(i+1)} = \{S^{(i)} \cap H(x^{(i+1)})\}$. } **2.2 (Area reduction loop)** If ($n^{(i)} \leq 8$) { **2.2.1** Choose $x^{(i+1)}$ as the centroid of $S^{(i)}$. **2.2.2** Find $H(x^{(i+1)})$. **2.2.3** Set $S^{(i+1)} = \{S^{(i)} \cap H(x^{(i+1)})\}$ } **2.3** *i* = *i* + 1 **End.**

To calculate the time complexity, we have already noted that the *Initial Triangle* in step 1 can be found in *O*(*n*) time. Consider now the main while-do loop. At the first iteration we have $n^{(1)}=3$. Step 2.1 if executed decreases the number of vertices by at least one (Corollary 3.5); if executed can increase the number of vertices by at most one. Therefore $n^{(i)} \leq 9$ in every iteration. Moreover, Step 2.2 will be executed at least once every two consecutive iterations of the while-do loop, hence $A(S^{(i+2)}) \leq 5A(S^{(i)})/9$. The *O* (1) bound on *n* $^{(i)}$ will easily guarantee that each iteration of the while-do loop can be completed in *O*(*n*) time. Steps 2.1.1 and 2.2.1 take $O(n^{(i)}) = O(1)$ time. As previously stated, the polygon *H*(*x*) in steps 2.1.2 and 2.2.2 can be found in *O*(*n*) time by the algorithm of Diaz and O'Rourke (1994). Further, given *S* $^{(i)}$ with *n* $^{(i)}$ vertices, steps and will take $O(n + n^{(i)}) = O(n)$ time, as they require us to find the intersection of two convex polygons 20. (We can actually reduce this to *O* (1) since all we need of *H*(*x*) is the line through *x*, but the preceding step time dominates.)

Since each iteration takes *O*(*n*) time and $A(S^{(i+2)}) \leq (5/9)^i A(P)/9$, then in order to produce a *Simpson_Polygon*with area $\varepsilon A(P)$ or less, at most $(3.4 \log 1/\varepsilon - 6.5) = O(\log 1/\varepsilon)$ iterations are needed. In Section 4, we show that convergence in area of order ε^3 yields convergence pointwise of order ϵ. Our main theorem then follows.

Theorem 3.6

Given a convex polygon P with a uniform demand distribution, and an error bound ϵ, *algorithm Find_Simpson_II produces a polygon inside P of area no more than* ϵ *A, that is guaranteed to contain x*, in* $O(n \log 1/\varepsilon)$ *time. Moreover, the algorithm produces a point x that is guaranteed to be within distance* ϵ *of x*, in* $O(3n \log 1/\varepsilon) = O(n \log 1/\varepsilon)$ *time.*

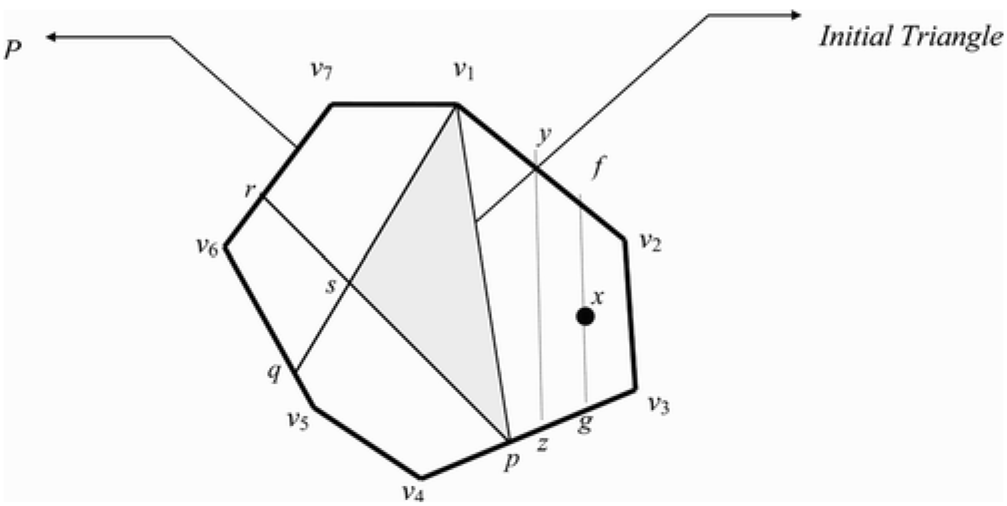

Figure 2. Initial triangle of *P*.

3.3. Non-uniform demand distribution

Here we will explore the case where the demand distribution over *P* is non-uniform. Our discussion is restricted to the high level. We assume the existence of subroutines to perform computations such as *H*(*x*) in time $\tau(n) = \Omega(n)$ on convex polygons with *n* vertices. We will assume that the demand distribution is continuous and positive over the domain of *P*. It can be verified that Lemmas 2.1, 2.2, 3.1, 3.2, 3.3, and Corollary 3.5 hold true even in this case. However, Lemmas 2.3 and 3.4 do not work in general when area is replaced by non-uniform demand. (Likewise, the results on the pointwise convergence may not hold). Thus, we cannot begin with an *Initial Triangle* that is guaranteed to contain the Simpson point. We will begin instead with a *Simpson_Polygon* that is given by P itself, and use the results of Lemmas 3.2 and 3.3 to produce one with no more than nine vertices. Then onwards, the algorithm will be executed similar to algorithm *Find_Simpson*_II.

Algorithm *Find_Simpson_* III(Non-Uniform Demand) **Begin 1:** $i = 1$. Set $S^{(i)} = P$ **2: while** $(A(S^{(i)}) > \varepsilon A(P))$ **do** { **2.1 (Vertex reduction loop)** If ($n^{(i)}>8$, i.e., $S^{(i)}$ has more than 8 vertices) { **2.1.1** Let $\delta^{(i)} = \lfloor (n^{(i)} - 9)/3 \rfloor$. **2.2.2** Find $\delta^{(i)}$-*Core*($S^{(i)}$). Choose $x^{(i+1)}$ as any point in the interior of this $\delta^{(i)}$-*Core*. **2.1.2** Find $H(x^{(i+1)})$. **2.1.3** Set $S^{(i+1)} = \{S^{(i)} \cap H(x^{(i+1)})\}$ } **2.2 (Area reduction loop)** If ($n^{(i)} \le 8$) { **2.2.1** Choose $x^{(i+1)}$ as the centroid of $S^{(i)}$. **2.2.2** Find $H(x^{(i+1)})$. **2.2.3** Set $S^{(i+1)} = \{S^{(i)} \cap H(x^{(i+1)})\}$ } **2.3** i=i+1 } **End.**

To analyse the time complexity of algorithm *Find_Simpson*_III, assume that $n \ge 10$. Then for the initial few, say i', iterations of the while-do loop of step 2, only the Vertex Reduction Loop of step 2.1 will be executed, until $n^{(i'+1)} \le 9$. Hence, we will now estimate i' and the time taken by the algorithm in the first i' steps. To that end, note that by our choice of $\delta^{(i)} = \lfloor (n^{(i)} - 3)/3 \rfloor$, and by Lemma 3.3, it is assured that for steps 1 through i',

$$n^{(1)} = n \text{ and for } 1 \le i \le i'$$

$$n^{(i+1)} \le n^{(i)} - \delta^{(i)} + 1 = n^{(i)} - \lfloor (n^{(i)} - 3)/3 \rfloor + 1$$

$$\le (2(n^{(i)} + 9))/3$$

Given the above recurrence relation for i' such that $n^{(i'+1)} \le 9$, obviously $i' = O(\log n)$. Hence, $O(\log n)$ iterations of the while-do loop will be necessary to obtain a *Simpson* polygon with at most nine vertices. Since each iteration will take $O(\tau(n) \log n^{(i)}) = O(\tau(n) \log n)$ time, the total time spent by the algorithm in the first i' steps is $O(\tau(n) \log^2 n)$.

However, at iteration $(i' + 1)$, we are guaranteed that the *Simpson_Polygon* will have no more than nine vertices. From then on, the analysis of the algorithm will be identical to that of algorithm *Find_Simpson*_II, i.e., it will be assured that for every two consecutive iterations of the while-do loop of step 2 (which will take $O(\tau(n))$ time), at least 4/9ths of the area of the *Simpson* polygon will be removed from future consideration. Thus, even if we conservatively assume that $A(S^{(i'+1)}) = A(S^{(1)}) = A(P)$, the total amount of time taken by algorithm *Find_Simpson*_III to produce a*Simpson* polygon of area no more than $\varepsilon A(P)$ is $O(\tau(n)(\log^2 n + \log 1/\varepsilon))$. This leads to the following proposition.

Proposition 3.10

Given a convex polygon P with a non-uniform demand distribution, and an error bound ϵ*, algorithm Find_Simpson_*III *produces a polygon inside P of area no more than* ϵA*, that is guaranteed to contain x*, in* $O(\tau(n)(\log^2 n + \log 1/\varepsilon))$ *time.*

When the demand of the region $\mu(H(x^*)) > 1/2$, Lemma 4.3 in the next section easily combines with Proposition 3.10 to guarantee pointwise convergence of the algorithm, but the rate of convergence is not as explicit as one would like, because it depends on the size of the gap $(\mu(H(x^*)) - 1/2)$, which is not known in advance.

One practical approach to handling non-uniform demand distributions in practical application is to approximate the non-uniform demand by a set of convex sub-polygons of *P* with uniform demands. The principal effect of this approximation will be to speed up the implementation algorithm *Find_Simpson_*III by reducing $\tau(n) = \Omega(n)$, which, in turn, will speed up the calculation of *H*(*x*). The improvement in computing time will depend on the number of such approximating sub-polygons; nonetheless, in practice, we would expect their numbers to not be excessive. As expected, such an approximation would result in an approximate, rather than, optimal, Simpson Point. Therefore, in practical applications, a decision maker can make trade-offs between the accuracy desired and the number of approximating sub-polygons that would be needed. Finally, another advantage of this approximation is that it will not affect the convergence results in Section 4.

4. Analysis of convergence

In this section, we show that both our algorithm and Diaz and O'Rourke's original *Find_Simpson_*I algorithm have good convergence properties. We have already seen that the area of the search regions $S^{(i)}$ decreases geometrically. As Diaz and O'Rourke point out, however, the trouble is that the regions might be long and thin. They showed that for even the very simple case of a rectangle, the $S^{(i)}$ produced by their algorithm may converge to a line segment rather than a point.

It turns out that their algorithm's convergence is better than they claimed. First, *only in the case* $A(H(x^*)) = A(P)/2$ can it occur that the $S^{(i)}$ does not converge to a point (Lemma 4.1). This simple case could be detected and dealt with separately. Second, even when the $S^{(i)}$ fails to converge, *the sequence of best candidate points from among* $x^{(1)} \ldots x^{(i)}$ *converges to the Simpson point* (Theorem 4.1). The same convergence properties hold for our enhanced algorithm.

4.1. Notation

P and *S* will denote convex polygons in the plane. As usual, the line segment with endpoints *x*, *y* is denoted $\overline{xy}$ and the line containing it is denoted $\overleftrightarrow{xy}$. If *H* is a hyperplane (line) the two halfspaces it defines are denoted H^+ and H^-. $A(S)$ as usual denotes the area of polygon S; $L(\overline{xy})$ denotes the length of a line segment $\overline{xy}$; $D(S) \equiv \max_{x \in S, y \in S} L(\overline{xy})$ denotes the diameter of polygon S.

If voters are uniformly distributed on polygon P, recall that the largest customer demand that can be mustered against $x \in P$ is the polygon $H(x)$. Define

$$\alpha_P(x) \equiv \frac{A(H(x))}{A(P)}$$

Our algorithm (either *Find_Simpson*_I or *Find_Simpson*_II) seeks x^*, the Simpson point of P, i.e., the point $x \in P$ that minimizes $\alpha_P(x)$. Our key definition follows.

Definition 4.1

Make the convention for any line G defining a side of polygon $S \subseteq P$ that the normal to G points away from S, i.e., $S \subseteq G^-$. A convex polygon S is an α-Simpson polygon of convex polygon P if $S \subseteq P$ and each side of polygon S is defined by a line G such that $A(G^- \cap P) \geq \alpha A(P)$. The definition applies to our algorithm in the following way. As usual let $x^{(1)} \ldots x^{(i)}$ be the sequence of points generated in the first i steps. Let

$$\hat{\alpha}^{(i)} \equiv \min_{1 \leq j \leq i} \alpha_P(x^{(j)}).$$

Let $\hat{x}(i)$ be the point at which the minimum is attained – thus $\alpha(\hat{x}^{(i)}) = \hat{\alpha}^{(i)}$. If $H(x^{(j)})$ is defined by line G then by definition,

$$\frac{A(G^- \cap P)}{A(P)} = \alpha_P(x^{(j)}) \geq \hat{\alpha}^{(i)}.$$

Therefore, the algorithm produces a sequence of $\hat{\alpha}^{(i)}$-Simpson polygons $S^{(i)}$. It also produces a sequence $\hat{x}^{(i)} \in S^{(i)}$ of associated candidate solutions. (The candidate point is contained in the polygon – actually it is on the boundary – because it cannot be cut off by another line G with smaller value $A(G^+ \cap P)/A(P) \leq \hat{\alpha}^{(i)}$. (Note that the sequence is not necessarily the sequence of candidates $x^{(j)} : j = 1 \ldots$; rather it is the sequence of the best candidates found so far.) We summarize the preceding observations.

Claim 4.1

Find_Simpson_I and Find_Simpson_II each produces a sequence of α^i – Simpson polygons i=1…, and candidate points $\hat{x}^i$ contained in the respective polygons $S^{(i)}$, with $\alpha_P(\hat{x}^i) = \alpha^i$.

The rest of the analysis depends only on these properties, so we can drop the superscript *i*.

4.2. Claims and lemmas

Claim 4.2

For any points *x*, *y* in convex polygon *S* s.t. $L(\overline{xy}) = D(S)$ and any distinct points *p*, *q* in *S* such that $\overline{pq}$ is perpendicular to $\overleftrightarrow{xy}$, we have $L(\overline{pq}) \leq (2A(S)/D(S))$. In particular, for any point *r*∈*S* the distance from *r* to $\overline{xy}$ is at most $(2A(S)/D(S))$.

Proof

The quadrilateral $xpyq \subseteq S$ by convexity. It has an area $(1/2)L(\overline{pq})L(\overline{xy})$.

Claim 4.3

Let *x*, *y* in convex polygon *S* be s.t. $L(\overline{xy}) = D(S)$. Then there exists a side $\overline{x_i x_{i+1}}$ of *S* contained in $\overleftrightarrow{xy}^+$ such that $\overleftrightarrow{x_i x_{i+1}}$ makes acute angle less than $\theta = \arctan\{(4A(S)/D(S)^2)\}$ with line $\overleftrightarrow{xy}$. Symmetrically *S* has a different side $\overline{y_j y_{j+1}}$ in $\overleftrightarrow{xy}^-$ whose defining line makes acute angle less than θ with line $\overleftrightarrow{xy}$.

Proof

Denote the extreme points of *S* as $\{x = x_0, x_1, \ldots, x_m = y = y_0, y_1, \ldots, y_n = x\}$ travelling clockwise from*x*. It is possible that *m*=1 (or later symmetrically *n*=1) in which case the side $\overline{xy}$ makes angle 0 with itself. Otherwise*m*>1. Let x_t be an x_i at maximum distance to $\overleftrightarrow{xy}$. By Claim 4.2 this distance is at most $(2A(S)/D(S))$. Without any loss of generality, assume x_t is closer to *y* than to *x*. Then the angle formed by points x_t, *x*, *y* has tangent at most

$$\frac{(2A(S)/D(S))}{1/2L(\overline{xy})}.$$

Finally, by convexity of *S*, $\overline{xx_t} \subset \overleftrightarrow{x_{t-1}x_t}^-$ (informally, the point x_{t-1} lies ‘above’ the segment $\overline{xx_t}$), hence the line defining side $\overline{x_{t-1}x_t}$ makes at least as small an angle with $\overleftrightarrow{xy}$ as possible. Repeat the argument to find a side of*S* on the other side of $\overleftrightarrow{xy}$.

Claim 4.4

Let *F* and *G* be lines defining edges of *S* as proved to exist in Claim 4.3. Let $Q = P \cap F^- \cap G^-$. Then

$$A(Q) \leq 4A(S)\left(\frac{D(P)}{D(S)}\right)^2$$

Proof

Let *F* (resp. *G*) intersect $\overleftrightarrow{xy}$ at *f* (resp. *G*). Define triangle F_Δ as follows: it has vertex *f*, and incident to *f* it has two sides, one lying in *F* and containing x_t, the other lying in $\overleftrightarrow{xy}$ and containing *y*, each of length *D*(*P*). Similarly define triangle G_Δ. Visually there are two cases, depending on whether *f* and *g* are both on the same side of $\overline{xy}$ or whether $\overline{xy} \subseteq \overline{fg}$, but in either case $Q \subseteq F_\Delta \cup G_\Delta$. Hence, $A(Q) \leq A(F_\Delta) + A(G_\Delta)$. Now $\sin\theta \leq \tan\theta$, so

$$A(F_\Delta) = \frac{1}{2}D(P)^2 \sin\theta \leq \frac{1}{2}D(P)^2 \frac{4A(S)}{D(S)^2} = 2A(S)\frac{D(P)^2}{D(S)^2}.$$

The area of G_Δ is bounded by the same quantity and the claimed bound follows.

Lemma 4.1

Let $(1/2) \leq \hat{\alpha} < 1$ *and let S be an* α̂*- Simpson polygon of P. Then*

$$D(S)\sqrt{\hat{\alpha} - \frac{1}{2}} \leq D(P)\sqrt{2\frac{A(S)}{A(P)}}.$$

Proof

Let *F* and *G* be lines according to claim 4.3. As in Claim 4.4, let $Q = P \cap F^- \cap G^-$. By inclusion–exclusion,

$$A(Q) \geq A(P) - A(P \cap F^+) - A(P \cap G^+) \geq A(P)(1 - (1 - \hat{\alpha}) - (1 - \hat{\alpha}))$$
$$= 2\left(\hat{\alpha} - \frac{1}{2}\right)A(P).$$

(The second inequality holds because *F* and *G* are lines defining sides of *S*, and *S* is an α̂-Simpson polygon of *P*). Combining this inequality with the upper bound on *A*(*Q*) from Claim 4.4 gives the result.

Lemma 4.1 tells us that the diameter of the inner (Simpson) polygon shrinks as its area shrinks, at rate square root of the reduction in area. However, Lemma 4.1 only guarantees this if the candidate value α̂ is bounded away from 1/2, and we do not know α* in advance. It is very easy to check for the case $\hat{\alpha} = (1/2)$: determine the horizontal and vertical lines that bisect the area of P (any two non-parallel normal vectors will do). Let y be the intersection of these two lines. Then y is the only possible solution. In other words, either $\alpha_P(y) = 1/2 = \alpha^*$ or $\alpha^* > (1/2)$. Even when $\alpha^* > 1/2$, Lemma 4.1 does not give a satisfactory convergence rate because α* may be very close to (1/2). We will need another lemma to get rid of the $(\hat{\alpha} - (1/2))$ term.

Lemma 4.2

Let S be any convex polygon in the plane. Let line H divide S into regions each with area at least t times the area of S, i.e., let H be such that $A(H^+ \cap S) \geq tA(S)$ *and* $A(H^- \cap S) \geq tA(S)$ *where* $0 < t \leq (1/2)$. *Then*

$$L(H \cap S) \geq \frac{tA(S)}{D(S)}. \qquad (1)$$

Proof

Consider the set of lines parallel to H, which have non-empty intersection with S. Parameterize these lines by $H_x : 0 \leq x \leq d_H$, where d_H is the length of the projection of S onto the normal (line perpendicular to H). Note $d_H \leq D(S)$. Define $f(x) = L(H_x \cap S)$, the length of the line segment formed by line H_x and S. Thus

$$A(S) = \int_0^{d_H} f(x)\mathrm{d}x. \qquad (2)$$

Let $0 < \hat{x} < d_H$ be the value for which $H = H_{\hat{x}}$. Thus $f(\hat{x}) = L(H \cap S)$ and

$$tA(S) \leq \int_0^{\hat{x}} f(x)\mathrm{d}x \leq (1-t)A(S) \Longrightarrow \int_{\hat{x}}^{d_H} f(x)\mathrm{d}x \geq tA(S). \qquad (3)$$

Observe that the function $f(x)$ is concave. This follows from the convexity of S. We consider two cases.

Case 1

$\hat{x} \geq \arg\max_x f(x)$. In this case, by concavity of f, $f(\hat{x}) \geq f(x) \forall x > \hat{x}$. Hence,

$$tA(S) \leq \int_{\hat{x}}^{d_H} f(x)\mathrm{d}x \leq (d_H - \hat{x}) f(\hat{x}) \leq d_H f(\hat{x})$$

$$\Longrightarrow f(\hat{x}) \geq \frac{tA(S)}{d_H}$$

Case 2

$\hat{x} \leq \arg\max_x f(x)$. By concavity of *f*, $f(\hat{x}) \geq f(x) \forall x < (\hat{x})$. Hence,

$$tA(S) \leq \int_0^{\hat{x}} f(x)\mathrm{d}x \leq (\hat{x} - 0) f(\hat{x}) \leq d_H f(\hat{x})$$

$$\implies f(\hat{x}) \geq \frac{tA(S)}{d_H}$$

4.3. Convergence

Theorem 4.1

Let $\alpha_P(\hat{x}) = \hat{\alpha} \leq \beta < 1$ *and let S containing* $\hat{x}$ *be an* αˆ*-Simpson polygon of P. Then the distance between* $\hat{x}$ *and x*, the Simpson point of P, is at most*

$$\sqrt[3]{\frac{4}{1-\beta}} D(P) \sqrt[3]{\frac{A(S)}{A(P)}}.$$

Proof

As given in the statement of the theorem, $\alpha_P(\hat{x}) = \hat{\alpha}$ and *x** is the optimum point with $\alpha_P^* = \alpha_P(x^*)$. Consider the line segment $T = \overline{\hat{x}x^*}$. Let H_1 (resp. H_2) be the line perpendicular to *T* through $\hat{x}$ (resp. *x**). Denote by H_i^+ the halfspace not containing *T*, for *i*=1, 2. Then $A(H_1^+ \cap P) \geq (1-\hat{\alpha})A(P)$ and $A(H_2^+ \cap P) \geq (1-\alpha^*)A(P)$. Therefore, every line *H* perpendicularly intersecting *T* satisfies the condition of Lemma 4.2 with $t = 1 - \hat{\alpha}$. By that lemma, for each such *H*, $L(H \cap P) \geq (1-\hat{\alpha})(A(P)/D(P))$. Therefore, the area in *P* ‘between’ H_1 and H_2 satisfies

$$A(P \cap H_1^- \cap H_2^-) \geq L(T)(1-\hat{\alpha})\frac{A(P)}{D(P)}.$$

Hence,

$$\hat{\alpha}A(P) \geq A(H_1^- \cap P) \geq A(H_2^+ \cap P) + L(T)(1-\hat{\alpha})\frac{A(P)}{D(P)} \geq (1-\alpha^*)A(P) + L(T)(1-\hat{\alpha})\frac{A(P)}{D(P)}.$$

From this and $\alpha^* \leq \hat{\alpha} \leq \beta$, we obtain

$$L(T) \le \frac{2}{1-\beta}\left(\hat{\alpha} - \frac{1}{2}\right) D(P). \qquad (4)$$

Because in our successive relaxation algorithm *S* always contains both $\hat{x}$ and *x**, we have $L(T) \le D(S)$. Combining the square of this inequality with the inequality (4), and applying Lemma 4.1, we have,

$$(L(T))^3 \le \frac{2}{1-\beta}(D(S))^2 D(P)\left(\hat{\alpha} - \frac{1}{2}\right) \le \frac{2}{1-\beta}(D(S))^2 D(P)\frac{2D(P)^2 A(S)}{A(P)D(S)^2}$$
$$= \frac{4}{1-\beta}(D(P))^3 \frac{A(S)}{A(P)}.$$

Theorem 4.1 tells us that any algorithm which generates a sequence of Simpson polygons, with area converging to 0, must also converge pointwise. To apply Theorem 4.1 to our algorithm, we can take β=(5/9) because the initial triangle provides a value no larger. Given that after 2*m* iterations, $A(S) \le (5/9)^m A(P)$, we conclude the following.

Corollary 4.4

Algorithm Find_Simpson_II pointwise converges at rate

$$L(\overline{x, x^*}) \le \sqrt[3]{9}D(P)\sqrt[3]{\frac{A(S)}{A(P)}} \le \sqrt[3]{9}D(P)\left(\frac{5}{9}\right)^{m/6},$$

where m is the number of iterations.

Similarly, Theorem 1 applies to the original Diaz-O'Rourke algorithm. We can take β=(5/9) again because of the centroid property stated earlier.

Corollary 4.5

Algorithm Find_Simpson_I pointwise converges at rate

$$L(\overline{x, x^*}) \le \sqrt[3]{9}D(P)\sqrt[3]{\frac{A(S)}{A(P)}} \le \sqrt[3]{9}D(P)\left(\frac{5}{9}\right)^{m/3},$$

where m is the number of iterations.

4.4. Extensions to the non-uniform case

Some of the convergence properties extend readily to the case of non-uniform convergence. The main idea is to use some of the results about area unchanged and to alter some other results using bounds on voter population density.

Assume that a continuous probability measure $\mu(x)$ is defined on *P*. By compactness of *P*, μ attains a maximum value $\mu_{\max}/A(P)$. Retaining the definitions of length *L*() and area *A*(), define the demand of a region as $V(S) \equiv \int_S \mu(x)\mathrm{d}x$. The definition of α_P() changes to

$$\alpha_P(x) \equiv \frac{V(H(x))}{V(P)} = V(H(x)),$$

since *V*(*P*)=1. The definition of Simpson polygon changes to the following.

Definition 4.2

A convex polygon S is an α*-Simpson polygon of convex polygon P if S*⊆ *P and each side of polygon S is defined by a line G such that* $V(G^- \cap P) \geq \alpha$*. Claims* 4.2, 4.3, *and* 4.4 *are all in terms of A*() *and remain unchanged. Lemma* 4.1 *and its proof change as follows*:

Lemma 4.3

Let $(1/2) < \hat{\alpha} < 1$ *and let S be an* $\hat{\alpha}$*-Simpson polygon of P. Then*

$$D(S)\sqrt{\hat{\alpha} - \frac{1}{2}} \leq D(P)\sqrt{2\mu_{\max}\frac{A(S)}{A(P)}}.$$

Proof

Let *F* and *G* be lines according to Claim 3. As in Claim 4, let $Q = P \cap F^- \cap G^-$. By inclusion–exclusion,

$$V(Q) \geq V(P) - V(P \cap F^+) - V(P \cap G^+) \geq (1 - (1 - \hat{\alpha}) - (1 - \hat{\alpha})) = 2\left(\hat{\alpha} - \frac{1}{2}\right).$$

(The second inequality holds because *F* and *G* are lines defining sides of *S*, and *S* is a $\hat{\alpha}$-Simpson polygon of *P*). On the other hand, $V(Q) \leq \mu_{\max}A(Q)/A(P)$.

Combining these two inequalities with the upper bound on *A*(*Q*) from Claim 4.4 gives the result.

If $\mu(x)>0$, there are finite non-zero bounds on minimum density and ratio of densities as well. However, this does not seem to be sufficient to generalize Lemma 4.2 and Theorem 4.1.

5. Conclusions and future research

We have investigated the problem of finding the Simpson Point, also called the (1| 1)- Centroid, of a convex polygon where the demand is continuously distributed over it. We have presented an enhancement of the Diaz and O'Rourke's iterative search algorithm for finding the Center of Area that solves the version of the problem when the demand distribution is uniform, reducing the time requirements from $O(|\log \varepsilon|^2 + n|\log \varepsilon|)$ to $O(n|\log \varepsilon|)$. We have also proved pointwise convergence (at unchanged time order cost) for both our version and Diaz and O'Rourke's original algorithm. Last, we outlined a high level modification of our enhanced algorithm for the problem when the demand distribution is continuous but non-uniform, which runs in time $O(\tau(n)(\log^2 n + |\log \varepsilon|))$ and pointwise converges if the value is bounded away from 1/2.

One immediate avenue for future research is to address the problem of finding the Simpson Point of a uniformly distributed population on a non-convex polygon. This problem is asymptotically close to a special case of non-uniform continuous distributions on convex polygons. Although Diaz and O'Rourke have developed an algorithm to find the Center of Area of non-convex polygons, it is not known yet if that also resolves the Simpson Point problem in the uniform and/or the non-uniform demand distribution case. Another avenue is to develop representations of non-uniform distributions and associated low-level subroutines to calculate centroids, $H(x)$, etc. Yet another open question is whether pointwise convergence, with a rate independent of the gap $(\mu(H(x*)) - 1/2)$, holds in these more general settings.

References

1. Brass, P., Heinrich-Litan, L. and Morin, P. 2003. Computing the center of area of a convex polygon. *Internat. J. Comput. Geom. Appl.*, 13: 439–445.

2. Carrizosa, E., Conde, E., Munoz-Marquez, M. and Puerto, J. 1997. Simpson points in planar problems with locational constraints: the round-norm case. *Math. Oper. Res.*, 22(2): 276–290.

3. Carrizosa, E. J., Conde, E., Munoz-Marquez, M. and Puerto, J. 1997. Simpson points in planar problems with locational constraints: the polyhedral-gauge case. *Math. Oper. Res.*, 22(2): 291–300.

4. Diaz, M. and O'Rourke, J. 1994. Algorithms for computing the center of area of a convex polygon. *Visual Comput.*, 10: 432–442.

5. Drezner, Z. 1982. Competitive location strategies for two facilities. *Regional Sci. Urban Econ.*, 12: 485–493.

6. Drezner, T. 1995. *Facility Location: A Survey of Applications and Methods*, Edited by: Drezner, Z. New York, Inc: Springer-Verlag.

7. Drezner, Z. and Zemel, E. 1992. Competitive location in the plane. *Ann. Oper. Res.,* 40(1): 173–190.

8. Durier, R. 1989. Continuous location theory under majority rule. *Math. Oper. Res.*, 14(2): 258–274.

9. Eiselt, H. A. and Laporte, G. 1997. Sequential location problems. *Eur. J. Oper. Res.*, 96(2): 217–231.

10. Grünbaum, B. 1960. Partitions of mass distributions and of convex bodies by hyperplanes. *Pacific J. Math.*, 10: 1257–1261.

11. Grünbaum, B. 1963. "*Measures of symmetry for convex sets*". In *Pro. Symposia in Pure Math.: Convexity,* Edited by: Klee, V. Vol. 7, 223–270. RI: American Mathematical Society, Providence.

12. Hakimi, S. L. 1990. "*Locations with spatial interaction: competitive locations and games*". In *Discrete Location Theory*, Edited by: Mirchandani, P. B. and Francis, R. L. New York: John Wiley & Sons.

13. Hammer, P. C. 1960. *Volumes cut form convex bodies by planes*, unpublished, made available by T.J. Smith, Department of Mathematics, University of Michigan, Kalamazoo

14. Hansen, P. and Labbé, M. 1988. Algorithms for voting and competitive location on a network. *Transport. Sci.*, 27(4): 278–288.

15. Hogatt, V. E. Jr. 1962. forbidden area. *The American Monthly Journal*, 62(2): 98–104.

16. Hotelling, H. 1929. Stability in competition. *Econ. J.*, 39: 41–57.

17. Langerman, S. and Steiger, W. *Optimization in arrangements*. Proceedings of the 20th International Symposium on Theoretical Aspects of Computer Science. Berlin, Germany.

18. Michelot, C. 1993. The mathematics of continuous location. *Stud. Locat. Anal.*, (5): 59–83.

19. Plastria, F. 2001. Static competitive facility location: an overview of optimisation approaches. *Eur. J. Oper. Res.*, 129(3): 461–470.

20. Preparata, F. P. and Shamos, M. I. 1985. *Computational Geometry: An Introduction*, New York, Inc: Springer Verlag.

21. Simpson, P. 1969. On defining areas of voter choice: Professor Tullock on stable voting. *Quart. J. Econ.*, 83: 478–487.

22. Singmaster, D. S. and Singmaster, G. 1967. Forbidden regions are convex. *Am. Math. Monthly*, 74: 184–186.

23. Tovey, C. A. 1993. “Some foundations for empirical study in the Euclidean spatial model of social choice, Ch. 7”. In *Political Economy: Institutions, Competition, and Representation*, Edited by: William, Barnett, Hinich, Melvin and Norman, Schofield. 175–194. New York: Cambridge University Press.

24. Yaglom, I. M. and Boltyanskii, V. G. 1961. *Convex figures*, *Library of the Mathematical Circle*, Vol. 4, New York, NY: Holt Rinehart and Winston. (Translated by Paul Kelly and Lewis Walton)