Parallel Delaunay Refinement with Off-Centers

Daniel A. Spielman[1], Shang-hua Teng[2], and Alper Üngör[3]

[1] Dept. of Mathematics, M.I.T., Cambridge, MA 02139, USA
spielman@math.mit.edu
[2] Dept. of Computer Science, Boston Univ., Boston, MA 02215, USA
steng@cs.bu.edu
[3] Dept. of Computer Science, Duke Univ., Durham, NC 27708, USA
ungor@cs.duke.edu

Abstract. Off-centers were recently introduced as an alternative type of Steiner points to circum-centers for computing size-optimal quality guaranteed Delaunay triangulations. In this paper, we study the depth of the off-center insertion hierarchy. We prove that Delaunay refinement with off-centers takes only $O(\log(L/h))$ parallel iterations, where L is the diameter of the domain, and h is the smallest edge in the initial triangulation. This is an improvement over the previously best known algorithm that runs in $O(\log^2(L/h))$ iterations.

Keywords: Delaunay refinement, parallel algorithms, triangulations.

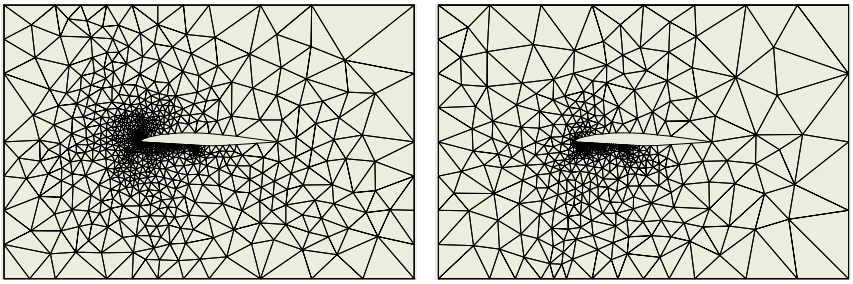
1 Introduction

Mesh generation problems ask for the discretization of an input domain into small and simple elements. These discretizations are essential in many applications including physical simulations, geographic information systems, computer graphics, and scientific visualization [9]. In addition to having the mesh conforming the input domain, most applications require that the mesh elements are of good quality and that the size of the mesh is small. A mesh element is considered good if its smallest angle is bounded from below. A bad element is likely to cause interpolation errors in the applications. Hence, mesh quality is critical for the accuracy and the convergence speed of the simulations. Mesh size, naturally, is also a factor in the running time of the applications algorithms.

Two main approaches for solving the mesh generation problem are the *quadtree methods* [2, 10] and the *Delaunay refinement methods* [5, 13]. Both methods compute quality-guaranteed size-optimal triangular meshes. However, Delaunay refinement methods are more popular than the quadtree methods mostly due to their superior performance in practice in generating smaller meshes. Over the years, many versions of the Delaunay refinement have been suggested in the literature [5, 7, 11, 13, 14, 16].

Delaunay triangulation of a given input is likely to have bad elements. Delaunay refinement iteratively adds new points, called *Steiner points*, into the domain to improve the quality of the mesh. A sequential Delaunay refinement

M. Danelutto, D. Laforenza, M. Vanneschi (Eds.): Euro-Par 2004, LNCS 3149, pp. 812–819, 2004.



(a) (b)

Fig. 1. Circumcenter vs. off-center insertion on an airfoil model. The smallest angle in both meshes is 32°. Delaunay refinement with circumcenters inserts 731 Steiner points and results in a mesh with 1430 triangles (a). On the other hand, Delaunay refinement with off-centers inserts 441 points and generates a mesh with 854 triangles (b).

algorithm typically adds one new vertex in each iteration. Each new vertex is chosen from a set of candidates – the circumcenters of bad triangles (to improve mesh quality) and the mid-points of input segments (to conform to the domain boundary). Ruppert [13] was the first to show that proper application of Delaunay refinement produces well-shaped meshes in two dimensions whose size is within a small constant factor of the best possible. Recently, we introduced a new type of Steiner points, called *off-centers* and proposed a new Delaunay refinement algorithm [16]. We proved that this new Delaunay refinement algorithm has the same theoretical guarantees as the Ruppert's refinement, and hence, generates quality-guaranteed size-optimal meshes. Moreover, experimental study indicates that our Delaunay refinement algorithm with off-centers inserts 40% fewer Steiner points than the circumcenter insertion algorithms and results in meshes 30% smaller in the number of elements. This implies substantial reduction not only in mesh generation time, but also in the running time of the application algorithms. Fig. 1 illustrates the performance difference between off-center and circumcenter insertion in meshing a region around an airplane wing. See [16] for further analysis.

Parallelization of Delaunay refinement methods is important for large scale applications. Recently, we gave the first parallel complexity analysis of the Delaunay refinement with circumcenters [15]. A main ingredient of this parallel algorithm is a notion of independence among candidate Steiner points for insertion at each iteration. The parallel algorithm consists of two main steps at each iteration. First, we generate an independent set of points for parallel insertion and then update the Delaunay triangulation in parallel. The independent sets have some nice properties. Insertion can be realized sequentially by Ruppert's Delaunay refinement method. Hence, an algorithm that inserts all the independent points in parallel will inherit the size and quality guarantees of Ruppert's method. The independent sets can be generated efficiently in parallel. In addition, they are "large enough" so that the number of parallel iterations needed is shown to be $O(\log^2(L/h))$, where L is the diameter of the domain and h is the smallest edge in the input triangulation [15]. In this paper, we show

that by replacing the circumcenters with the off-centers, we improve the bound on the number of iterations of the parallel Delaunay refinement algorithm to $O(\log(L/h))$. As a result, the work of our parallel Delaunay refinement algorithm is improved to $O(m \log m \log(L/h))$, where m is the output size. This is close to $O((n \log(L/h) + m) \log m)$ time bound of Miller's sequential algorithm [11], where n is the input size.

2 Delaunay Refinement with Off-Centers

In two dimensions, the input domain Ω is represented as a *planar straight line graph* (PSLG) – a proper planar drawing in which each edge is mapped to a straight line segment between its two endpoints [13]. Due to space limitation we present our parallelization results only on periodic point sets, a special type of PSLG. If P is a finite set of points in the half open unit square $[0,1)^2$ and $\mathbb{Z}^2$ is the two dimensional integer grid, then $S = P + \mathbb{Z}^2$ is a *periodic point set* [6]. The periodic set S contains all points $p + v$, where $p \in P$ and v is an integer vector. As P is contained in the unit square, the diameter of P is $L \leq \sqrt{2}$. It is worth to note that some of the pioneering theoretical mesh generation work, such as sliver removal algorithms, are first studied on periodic point sets [4].

Let P be a point set in $\mathbb{R}^d$. A simplex τ formed by a subset of P points is a *Delaunay simplex* if there exists a circumsphere of τ whose interior does not contain any points in P. This empty sphere property is often referred to as the *Delaunay property*. The Delaunay triangulation of P, denoted $Del(P)$, is a collection of all Delaunay simplices. If the points are in general position, that is, if no $d+2$ points in P are co-spherical, then $Del(P)$ is a simplicial complex. The Delaunay triangulation of a periodic point set is also periodic. The Delaunay triangulation of a point set can be constructed in $O(n \log n)$ time in two dimensions [6].

Radius-edge ratio of a triangle is the ratio of its circumradius to the length of its shortest side. A triangle is considered *bad* if its radius-edge ratio is larger than a pre-specified constant $\beta \geq \sqrt{2}$. This quality measure is equivalent to other well-known quality measures, such as smallest angle and aspect ratio, in two dimensions [13].

The line that goes through the midpoint of an edge of a triangle and its circumcenter is called the *bisector* of the edge. Given a bad triangle pqr, suppose that its shortest edge is pq. Let c denote the circumcenter of pqr. We define the *off-center* to be the circumcenter of pqr if the radius-edge-ratio of pqc is smaller than or equal to β. Otherwise, the *off-center* is the point on the bisector, which makes the radius-edge ratio of the triangle based on p, q and the off-center itself β (Figure 2). The circle that is centered at the off-center and goes through the endpoints of the shortest edge is called the *off-circle*. In the first case, off-circle is same as the circumcircle of the triangle.

The description of the Delaunay refinement algorithm with off-centers is very simple for the periodic point set input. We maintain the Delaunay triangulation of the point set. As long as there exists a bad triangle in the triangulation, we

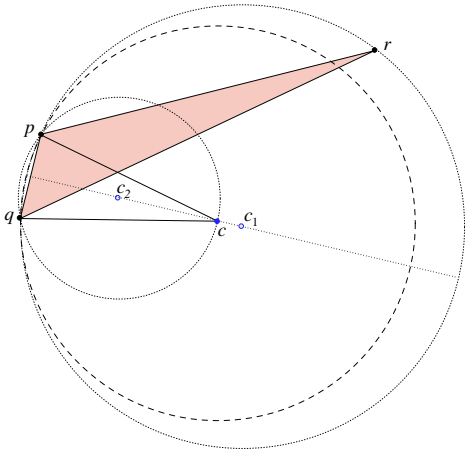


Fig. 2. The off-center and the circumcenter of triangle pqr is labeled c and c_1 respectively. The circumcenter of pqc is labeled as c_2. The off-circle of pqr is shown dashed.

insert its off-center as a Steiner point and update the Delaunay triangulation. We refer to [16] for a detailed description of the algorithm for PSLGs and also for the termination and size-optimality proofs.

3 Parallel Delaunay Refinement with Off-Centers

When parallelizing a Delaunay refinement algorithm, at each parallel iteration we would like to insert as many Steiner points as possible. However, some off-centers can be arbitrarily close to each other, hence neither a sequential nor a parallel refinement algorithm insert them all and can still provide termination guarantee. Furthermore, we would like to insert a set of points that has a sequential realization, i.e. there exist a provably good sequential algorithm that inserts the same set of points. We select the set of insertion points in a parallel iteration based on the following definition of *independence* among candidate off-centers.

Definition 1. *Two off-centers $\dot{c}_a$ and $\dot{c}_b$ (and also the corresponding off-circles c_a and c_b) are said to* conflict *if both c_a and c_b contain each other's off-center. Otherwise, $\dot{c}_a$ and $\dot{c}_b$ (respectively c_a and c_b) are said to be* independent.

Our parallelization of the off-center insertion is based on the same edge classification that we used in parallelizing the Ruppert's refinement [15]. Let h be the shortest edge length in the initial Delaunay triangulation. The class $\mathcal{E}_i$ contains the edges whose length are in the half open interval $\left[\sqrt{2}^{i-1}h, \sqrt{2}^{i}h\right)$. Therefore, there are at most $\lceil \log_{\sqrt{2}}(L/h) \rceil$ edge classes. A triangle is said to be associated with edge class $\mathcal{E}_i$ if its shortest edge is in $\mathcal{E}_i$. The ith iteration of the outer loop in the following algorithm removes all the bad triangles associated with $\mathcal{E}_i$.

Algorithm 1 PARALLEL DELAUNAY REFINEMENT WITH OFF-CENTERS.

Input: A periodic point set P in $\mathbb{R}^2$
  Let T be the Delaunay triangulation of P
  for i=1 to $\lceil \log_{\sqrt{2}}(L/h) \rceil$ **do**
    Let $\dot{\mathcal{C}} = \{\dot{c}_1, \ldots, \dot{c}_n\}$ be the set of off-centers of bad triangles associated with $\mathcal{E}_i$
    while $\dot{\mathcal{C}}$ is not empty **do**
      Let $\mathcal{I}$ be a maximal independent subset of $\dot{\mathcal{C}}$
      Update Delaunay triangulation inserting all points in $\mathcal{I}$ in parallel
      Update $\dot{\mathcal{C}}$
    end while
  end for

Lemma 1. *Suppose all the triangles associated with edge classes $\mathcal{E}_j$, $\forall j < i$ are good. Then, off-circle of every bad triangle associated with class $\mathcal{E}_i$ is empty of other points.*

Proof. Suppose the off-circle is not empty. Then, there exists a vertex inside off-circle but outside the circumcircle (Delaunay property). Consider a morph of the circumcircle into off-circle, where the morphing circle passes through the endpoints of the shortest edge pq and hence its center moves from the circumcenter to the off-center along the bisector. Let w be the vertex that the morphing circle hits first. The triangle pqw is bad because its circumradius is larger than the radius of the off-circle. Moreover, its shortest edge is less than $|pq|/\sqrt{2}$. This implies that pqw is associated with an edge class $\mathcal{E}_j$ where $j < i$. This contradicts the assumption. ◻

Lemma 2. *Suppose c_a and c_b are two conflicting off-circles of two triangles associated with class $\mathcal{E}_i$ at a parallel iteration, and let r_a and r_b be their radii. Then, $r_b/2 < r_a < 2r_b$.*

Proof. Off-circle c_a contains $\dot{c}_b$ and (by Lemma 1) no vertices of the triangulation. As some vertex on c_a lies outside c_b, the diameter of c_a is greater than the radius of c_b. Thus, $r_a > r_b/2$. A symmetric argument implies $2r_b > r_a$. ◻

We next show that the mesh generated by Algorithm 1 is realizable by the sequential algorithm introduced in [16] and described in Section 2.

Theorem 1. *Suppose M is a mesh produced by an execution of the* PARALLEL DELAUNAY REFINEMENT WITH OFF-CENTERS. *Then, M can be obtained by some execution of the sequential* DELAUNAY REFINEMENT WITH OFF-CENTERS.

Proof. Let $\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_k$ be the sets of vertices inserted by the PARALLEL DELAUNAY REFINEMENT WITH OFF-CENTER at iterations $1, \ldots, k$, respectively. We describe a sequential execution that inserts all the points in $\mathcal{I}_i$ before any point of $\mathcal{I}_j$ for $i < j$. In other words, first all the points in set $\mathcal{I}_1$ are inserted sequentially, then all the points in set $\mathcal{I}_2$, and so on. To determine the order within each maximal independent set, we use a dependency graph. For any two

off-centers $\dot{a}, \dot{b} \in \mathcal{I}_i$, $\dot{a}$ has to wait for the insertion of $\dot{b}$ if $\dot{a}$ is inside the corresponding off-circle b of $\dot{b}$. In the dependency graph we put an edge from $\dot{a}$ to $\dot{b}$. Because each edge is directed from a smaller circle to a larger circle, this dependency graph is acyclic. Topological sorting of the vertices of the dependency graph gives us a valid sequential insertion within each $\mathcal{I}_i$. Notice that in a sequential realization an off-center $\dot{a} \in \mathcal{I}_i$ remains a candidate until it is inserted. This is because the shortest edge of the corresponding triangle remains to be the shortest edge of a bad triangle (not necessarily the same one).

Furthermore, in any sequential execution, a point $p \in \mathcal{I}_i$ cannot eliminate a point $q \in \mathcal{I}_j$ for any $i < j$. Otherwise, q would not be inserted in the jth iteration of the parallel execution. Finally, the parallel and sequential executions terminate after inserting exactly the same set of points. An extra element in the insertion set of one would indicate that the execution of the other one is not terminated, as this implies existence of a bad element. Since the same points are inserted effectively in the same order, the output mesh is the same. ◻

We recall that β is the threshold of the ratio of the radius to shortest edge-length defining a bad triangle. Thus, for $\beta \geq \sqrt{2}$, inserting the off-center of a bad triangle whose shortest edge length is l, introduces new Delaunay edges of length at least $\sqrt{2}l$.

Lemma 3. *Suppose the shortest edge associated with any bad triangle is in $\mathcal{E}_i$. Let $e \in \mathcal{E}_i$ be the shortest edge of a bad triangle that exists before the first iteration of the inner loop. Then, after $O(1)$ iterations, either e does not exist anymore or all the triangles associated with e are good.*

Proof. Suppose e exists and still is the shortest edge of a bad triangle pqr after iteration 51 of the inner loop. This implies that the off-center $\dot{c}$ of pqr is not inserted because it was in conflict with another vertex at each of the iterations 1 through 51 of the inner loop during the ith iteration of the outer loop. So, for each iteration $k = 1, \ldots, 51$, an off-center $\dot{c}'_k$ in conflict with $\dot{c}$ is inserted. Moreover, by Lemma 2 the radius r'_k of the corresponding off-circle c'_k is at least half the size of c, i.e. $r'_k > r/2$, where r is the radius of the off-circle associated with e. Let $\dot{C}' = \{\dot{c}'_1, \dot{c}'_2, \ldots, \dot{c}'_{51}\}$ be the set of off-centers that were inserted in iterations $k = 1, \ldots, 51$. Let C' be the corresponding set of off-circles. The radius of each one of these circles in C' is at least $r/2$. By Lemma 1, each circle $c'_k \in C'$ is empty of all the centers in C' inserted prior to $\dot{c}'_k$. So the centers in $\dot{C}'$ are pairwise at least $r/2$ apart from each other. This in turn implies that the circles of radius $r/4$ on the centers in $\dot{C}'$ do not overlap. These circles can spread into an area of size at most $2\pi(r+r/4)^2$ because each of the corresponding off-centers is in conflict with an off-center on either side of the edge e. One can fit at most

$$\left\lfloor \frac{2\pi(r + r/4)^2}{\pi(r/4)^2} \right\rfloor = 50$$

circles of radius $r/4$ in that region. Therefore, the number of centers in $\dot{C}'$ is at most 50, which is a contradiction. ◻

Our algorithm, one at a time, handles the bad triangles associated with each edge class. In the next Lemma we justify the double loop structure of our parallel refinement algorithm. We prove that during and after the ith iteration of the outer loop no bad triangle associated with an edge in $\mathcal{E}_i$ is introduced.

Lemma 4. *Suppose the shortest edge associated with any triangle is in $\mathcal{E}_i$. During and after iteration i of the outer loop of Algorithm 1, the following are true:*

I *if an edge $e \in \mathcal{E}_i$ disappears it never appears again;*
II *no new edges are introduced to edge class $\mathcal{E}_j$, $\forall j \leq i$;*
III *the radius-edge ratio of a triangle associated with $\mathcal{E}_i$ does not increase.*

Proof. (I) If there is no empty circle containing e before a Steiner point insertion, clearly there is no such circle after the insertion. (II) During and after the ith iteration of the outer loop the smallest edge that can be introduced has length at least $\sqrt{2}^i h$. (III) Consider an edge $pq \in \mathcal{E}_i$ and a triangle pqr associated with pq. We claim that the quality of a triangle pqu replacing pqr is better than that of pqr. The new vertex u must be inside the circumcircle of pqr. Otherwise, pqr is intact. On the other hand, u can not be too close to edge pq, i.e., $\min\{|pu|, |qu|\} \geq |pq|$. Otherwise, a shorter edge than pq would be introduced contradicting (II). The radius of the circumcircle of pqu is smaller than that of pqr when u is inside the circumcircle of pqr but outside the diametral circle of pq. This in turn, implies that the radius-edge ratio of a triangle associated to an edge in $\mathcal{E}_i$ can only be improved through refinement. ◻

Theorem 2. Parallel Delaunay Refinement with Off-centers *algorithm takes $O(\log(L/h))$ iterations to generate a size-optimal well-shaped uniform mesh, where L is the diameter of the input and h is the length of the shortest edge in the initial triangulation.*

Proof. By Lemma 3 and 4, the ith outer loop of the algorithm takes $O(1)$ parallel iterations to fix all the triangles associated with $\mathcal{E}_i$. Overall the algorithm takes at most $51 \log_{\sqrt{2}}(L/h)$ parallel iterations. The size optimality and quality guarantee of the parallel algorithm follows from Theorem 1. ◻

4 Discussions

We should note that in our design and analysis of the parallel Delaunay refinement, we focused and gave a bound on the number of parallel iterations. To complete the parallel time complexity analysis at each iteration we employ the parallel maximal independent set algorithm presented in [15] and one of the parallel Delaunay triangulation algorithms given in [1, 12]. As a future research we plan to extend the off-center algorithm to three dimensions and explore its benefits both in theoretical and practical fronts.

Acknowledgements

Research of the first author is partially supported by NSF under grant CCR-0112487. Research of the second author is partially supported by NSF under grants CCR-0311430 and CCR-0325630. Research of the third author is partially supported by NSF under grant CCR-00-86013.

References

1. N. M. Amato, M. T. Goodrich, and E. A. Ramos. Parallel algorithms for higher-dimensional convex hulls. *Proc. 35th IEEE Symp. Foundations of Computer Science*, 683–694, 1994.
2. M. Bern, D. Eppstein, and J. Gilbert. Provably good mesh generation. *J. Comp. System Sciences* 48:384–409, 1994.
3. M. Bern, D. Eppstein, and S.-H. Teng. Parallel construction of quadtrees and quality triangulations. *Int. J. Comp. Geometry & Applications* 9(6):517–532, 1999.
4. S.-W. Cheng, T. K. Dey, H. Edelsbrunner,M. A. Facello, and S.-H. Teng. Sliver exudation. *Proc.* 15*th ACM Symp. Comp. Geometry*, 1–13, 1999.
5. P. Chew. Guaranteed-quality triangular meshes. TR-89-983, Cornell Univ., 1989.
6. H. Edelsbrunner. *Geometry and Topology for Mesh Generation*. Cambridge Univ. Press, 2001.
7. H. Edelsbrunner and D. Guoy. Sink insertion for mesh improvement. *Proc.* 17*th ACM Symp. Comp. Geometry*, 115–123, 2001.
8. W.H. Frey. Selective refinement: a new strategy for automatic node placement in graded triangular meshes. *Int. J. Numerical Meth. Eng.*, 24, 2183–2200, 1987.
9. P. Frey, and P.-L. George. *Mesh Generation*. Oxford, Paris, 2000.
10. S. Mitchell and S. Vavasis. Quality mesh generation in three dimensions. *Proc.* 8*th ACM Symp. Comp. Geometry*, 212–221, 1992.
11. G. L. Miller. A time efficient Delaunay refinement algorithm. *Proc. ACM-SIAM Symp. on Disc. Algorithms*, 400–409, 2004.
12. J. H. Reif and S. Sen. Parallel computational geometry: An approach using randomization. *Handbook of Comp. Geometry*, 765–828, Elsevier, Amsterdam, 1999.
13. J. Ruppert. A new and simple algorithm for quality 2-dimensional mesh generation. *Proc.* 4*th ACM-SIAM Symp. on Disc. Algorithms*, 83–92, 1993.
14. J. R. Shewchuk. *Delaunay Refinement Mesh Generation*. Ph.D. thesis, Carnegie Mellon University, 1997.
15. D. A. Spielman, S.-H. Teng, and A. Üngör. Parallel Delaunay refinement: Algorithms and analyses. *Proc.* 11*th Int. Meshing Roundtable*, 205–217, 2002.
16. A. Üngör. Off-centers: A new type of Steiner points for computing size-optimal quality-guaranteed Delaunay triangulations. *Proc. of Latin American Theoretical Informatics (LATIN)*, 151–162, 2004,