Combinatorics of Beacon-Based Routing and Coverage

Michael Biro* Jie Gao† Justin Iwerks* Irina Kostitsyna† Joseph S. B. Mitchell*

Abstract

We consider combinatorial problems motivated by sensor networks for *beacon*-based point-to-point routing and covering. A beacon b is a point that can be *activated* to effect a 'magnetic pull' toward itself everywhere in a polygonal domain P. The routing problem asks how many beacons are required to route between any pair of points in a polygonal domain P. In simple polygons with n vertices we show that $\lfloor \frac{n}{2} \rfloor - 1$ beacons are sometimes necessary and always sufficient. In polygons with h holes, we establish that $\lfloor \frac{n}{2} \rfloor - h - 1$ beacons are sometimes necessary while $\lfloor \frac{n}{2} \rfloor + h - 1$ beacons are always sufficient. Loose bounds for simple orthogonal polygons are also shown. We consider art gallery problems where beacons function as guards. Loose bounds are given for covering simple polygons, polygons with holes and simple orthogonal polygons.

1 Introduction

The model of beacon-based routing in this paper is an analog of geographical greedy routing in sensor networks in the continuous setting. A comparison of our model with others found in the literature can be found in the the current authors' related paper on beacon-based algorithms [1]. We consider two main questions in this paper: "How many beacons are required to allow for point-to-point routing between any two points in a polygonal domain P?" and "How many beacons are required to cover P?"

In our model, a beacon can occupy a point location on the interior or the boundary of P, ∂P. When a beacon is *activated*, an object p in P moves along a straight line toward b until either it reaches b or makes contact with ∂P. If contact is made with ∂P, p will follow along ∂P as long as its straight line distance to b decreases monotonically. p may alternate between moving in a straight line path toward b on the interior of P and following along ∂P. If p is unable to move so that its distance to b decreases monotonically, we say p is 'stuck' and has reached a local minimum at a *dead point*. If an object p originating at a point q reaches b we say that b *attracts* q. A point s is *routed* to t if there is a sequence of beacons that can be activated and then deactivated, one at a time in order, such that an object beginning at a source s visits each beacon in the sequence after it is activated and terminates at a destination t, which we always assume to be a beacon itself, but is not counted. We restrict each beacon to be activated at most one time during a routing. We say that a polygon P is covered by a set of beacons if every point of P is attracted by at least one beacon in the set.

*Department of Applied Mathematics and Statistics, Stony Brook University, mbiro@ams.stonybrook.edu, jiwerks@ams.stonybrook.edu, jsbm@ams.stonybrook.edu

†Department of Computer Science, Stony Brook University, jgao@cs.stonybrook.edu, ikost@cs.stonybrook.edu

2 Routing in Simple Polygons

Suppose first that P is a simple polygon. Then we show tight bounds on the number of beacons necessary to route between a pair s, t of points in P and the number of beacons sufficient to route between *any* pair of points s, t in P.

Theorem 1 *Given a simple polygon P, $\lfloor \frac{n}{2} \rfloor - 1$ beacons are sometimes necessary and always sufficient to route between any pair of points in P.*

Proof. We can see from Figure 1 that $\lfloor \frac{n}{2} \rfloor - 1$ beacons are sometimes necessary to route between a specific pair s and t.

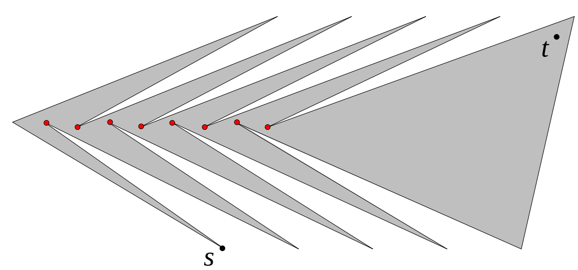

Figure 1: $\lfloor \frac{n}{2} \rfloor - 1$ beacons are sometimes necessary to route between a pair of points in a simple polygon. Here, $n = 19$ and 8 beacons are required to route from s to t.

To establish the upper bound, we first triangulate P and construct the dual graph G of the resulting triangulation, rooted at an arbitrary triangle. Beginning with a lowest leaf node of G, we begin to peel off adjacent triangles. Suppose the leaf triangle is σ_1 and its neighbor is σ_2. The analysis depends on the degree of σ_2 in the triangulation.

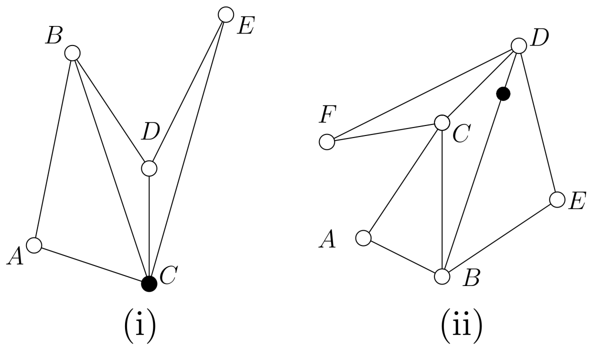

Figure 2: (i) The beacon b is placed at a vertex common to three triangles; (ii) The beacon b is placed appropriately on the edge BD. Any point in the four triangles can then navigate to b and vice-versa without any additional beacons needed.

(i) σ_2 has only one additional adjacent triangle σ_3. Suppose $\sigma_1 = \triangle ABC$, $\sigma_2 = \triangle BCD$. σ_3 is then either $\triangle BDE$ or $\triangle CDE$. If $\sigma_3 = \triangle CDE$, then we place a beacon b at the vertex C and otherwise we place b at B. In either case, since b is contained in each of the three triangles, any point p in these three triangles can navigate to b and vice-versa (see Figure 2 (i)).

(ii) σ_2 has two additional adjacent triangles σ_3, σ_4. Assume that $\sigma_1 = \triangle ABC$, $\sigma_2 = \triangle BCD$, $\sigma_3 = \triangle BDE$, $\sigma_4 = \triangle CDF$. Suppose that the path from σ_1 to the root passes through triangle σ_3. Since $\sigma_1 = \triangle ABC$ was a lowest leaf, $\sigma_4 = CDF$ is also a leaf. We place a beacon on the diagonal BD. The location b along BD is chosen so the pentagon $ABDFC$ is visible to b. This is always possible, by placing b on the correct side of lines CF and AC. Then, any point in triangles $\triangle ABC, \triangle BCD, \triangle CDF$ can be routed to or from b as b is visible to each point in those triangles. Hence, the claim is true (see Figure 2 (ii)).

Given the basic steps as shown above, we will place beacons in a recursive manner: We take any lowest leaf triangle σ_1 of the triangulation of P and place a beacon at points described above.

1. If P is a single triangle, we do nothing. If P has at most one more triangle besides σ_1 and σ_2 or P has only two more triangles but both are adjacent to σ_2, then we are done after placing one more beacon (see Figure 3 (ii) or (iii)). By the above arguments we can navigate from any starting point to any destination point by using the single beacon: First route from the start to the beacon and then route from the beacon to the destination (which is always a beacon).

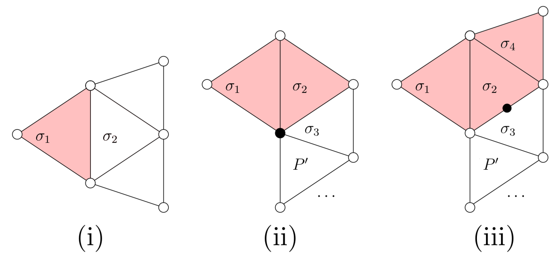

Figure 3: Inductive placement of beacons. (i): Base case; (ii): Peeling off σ_1 and σ_2 leaves a simple polygon; (iii): Peeling off σ_1, σ_2, and σ_4, leaves a simple polygon.

2. Otherwise, we peel off σ_1 at least one more triangle. There are two subcases to consider:

   (a) σ_2 is only adjacent to one more triangle σ_3 (i.e., σ_2 has degree 2 in the dual graph; see Figure 3 (ii)). In this case peeling off σ_1 and σ_2 will still leave a simple polygon P'. We can recursively 'beaconize' P'. Now we argue that one can navigate with the union of these beacons. In particular, if the start and destination pair are both in $\sigma_1 \cup \sigma_2$, we may route from s to b and from b to t as b is visible to both s and t. If both s and t are in P', then we can navigate by induction hypothesis. If the start and destination pair are separated in $\sigma_1 \cup \sigma_2$ and P', we can route from s to the beacon b and then from b to t by the induction hypothesis and the analysis above. Thus navigation works in this case.

   (b) σ_2 is adjacent to two other triangles σ_3 and σ_4, with σ_4 also a leaf. Thus peeling off σ_1, σ_2, and σ_4 will still leave a simple polygon P'; see Figure 3 (iii)). We can recursively 'beaconize' P'. Now we argue that one can navigate with the union of these beacons. In particular, if the start and destination pair are both in $\sigma_1 \cup \sigma_2 \cup \sigma_4$, we may route from s to b and then from b to t as t is visible to both s and t. If both s and t are in P', we can navigate by the induction hypothesis. If the start and destination pair are separated, we route from s to b and then from b to t, again by the induction hypothesis and the analysis above. Thus, navigation works in this case as well.

With the algorithm, we can see that each time we place a beacon we peel off at least two triangles. There are $n-2$ triangles in any triangulation of a simple polygon with n vertices. Thus the total number of beacons we place would be at most $\lfloor\frac{n-2}{2}\rfloor = \lfloor\frac{n}{2}\rfloor - 1$. □

3 Routing in Polygons with Holes

If P has n vertices and h holes, then we give bounds on the number of beacons that are sometimes necessary and always sufficient to route between any pair of points in P.

Theorem 2 *Given a polygon P with n vertices and h holes, $\lfloor\frac{n}{2}\rfloor - h - 1$ beacons are sometimes necessary to route between a pair of points in P. Conversely, $\lfloor\frac{n}{2}\rfloor + h - 1$ beacons are always sufficient to route between any pair of points in P.*

Proof. Figure 5 illustrates that $\lfloor\frac{n}{2}\rfloor - h - 1$ beacons are sometimes necessary to route between a specific pair of points s and t. The figure shows the polygon from Figure 1, with another copy of that polygon placed where s was originally in Figure 1, as in Figure 4. The original polygon requires 19 vertices and the additional hole polygon has 19 vertices, plus one extra to close the hole, so 20 vertices, and 39 total. We have that 8 beacons are required for the original, and an additional 9 for the hole, so 17 beacons are required to route from s to t in this polygon. That is $\lfloor\frac{n}{2}\rfloor - h - 1 = \lfloor\frac{39}{2}\rfloor - 1 - 1 = 19 - 2 = 17$. This process may be iteratively repeated to achieve the $\lfloor\frac{n}{2}\rfloor - h - 1$ bound for large numbers of holes and vertices.

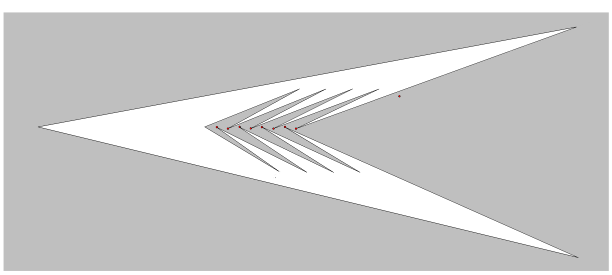

Figure 4: Closing a copy of the polygon in Figure 1 with an additional vertex to create a hole with 20 vertices that requires 9 beacons.

To establish the upper bound, we first triangulate P and construct the dual graph G of the resulting triangulation. Since P is not simple, there may be cycles in the dual graph and so, for each cycle in G we remove an edge. This leaves the dual graph connected and is equivalent to cutting a thin channel in the polygon to remove a hole, thus adding 2 vertices. After h cuts, the resulting polygon is simple and has $n + 2h$ vertices. We then utilize Theorem 1, to say that the resulting polygon may be routed with $\lfloor\frac{n+2h}{2}\rfloor - 1 = \lfloor\frac{n}{2}\rfloor + h - 1$ beacons, as the beacons placed do not depend on the rigidity of the edges of the polygon. Then, a valid routing sequence in the modified simple polygon corresponds to a valid routing sequence in the original polygon P. □

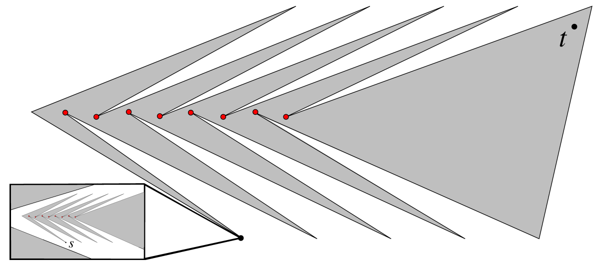

Figure 5: $\lfloor\frac{n}{2}\rfloor - h - 1$ beacons are sometimes necessary to route between a pair of points. Here, $n = 39$, $h = 1$, and 17 beacons are required to route from s to t.

Conjecture 1 *Given a polygon P with n vertices and h holes, $\lfloor\frac{n}{2}\rfloor - h - 1$ beacons are always sufficient to route between any pair of points in P.*

4 Routing in Simple Orthogonal Polygons

In simple orthogonal polygons we give loose bounds for the number of beacons necessary to route between a pair of points s, t in P and the number of beacons sufficient to route between *any* two points s, t in P. We were unsuccessful in attempting to mimic the proof of Theorem 1 with a peeling process on a convex quadrilateralization of P in order to improve the upper bound.

Theorem 3 *Given a simple orthogonal polygon P with n vertices, $\lfloor\frac{n}{4}\rfloor - 1$ beacons are sometimes necessary while $\lfloor\frac{n}{2}\rfloor - 1$ beacons always sufficient to route between any pair of points in P.*

Proof. We can see that $\lfloor\frac{n}{4}\rfloor - 1$ beacons are sometimes necessary to route between any pair of points in P by constructing an orthogonal 'zig-zag' polygon as in Figure 6. The upper bound carries over from Theorem 1 for general simple polygons. □

5 Coverage in Simple Polygons and Polygons with Holes

Define the *attraction region* of a beacon b to be the locus of points in P that can reach b when b is activated. Also, let the *inverse attraction region* of point p be the locus of points that can attract p. Then we say that a set of beacons B covers a polygon P if P is entirely contained in the union of the attraction regions of the beacons in B. In this section, we give bounds on the number of beacons that are sometimes necessary and

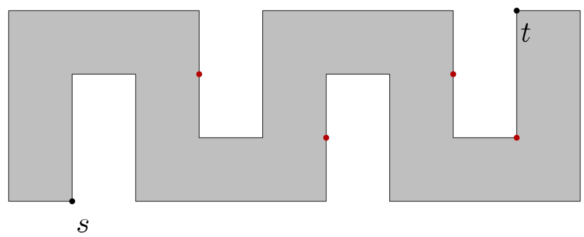

Figure 6: $\lfloor \frac{n}{4} \rfloor - 1$ beacons are sometimes necessary to route between a pair of points in an orthogonal polygon. Here, $n = 20$ and 4 beacons (light filled circles) are required to route from s to t.

always sufficient to cover a simple polygon.

Figure 7 depicts a small polygon with 9 vertices that requires 3 beacons to cover. Due to the angles involved, this small example cannot easily be extended to larger n.

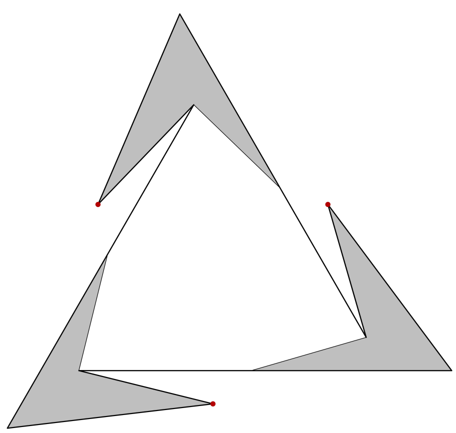

Figure 7: Here, $n = 9$ and $\frac{9}{3} = 3$ beacons are required to cover. The (red) independent witnesses and their disjoint inverse attraction regions are shaded.

In order to get around the angle issue, we try to make multiple copies of the above figure in a linear pattern. This yields a polygon with a repeating spike gadget, requiring $\lfloor \frac{3n}{10} \rfloor$ beacons to cover.

In order to further improve this lower bound, we iteratively 'glue' spikes together. We distinguish between spike trunks that are 'angled-in' and those that are 'angled-out', and proceed to glue two copies of an angled-in spike to form an angled-out spike, as in Figure 9. The next operation takes two such angled-out spikes and glues them together to form a new angled-in spike. This process can then be iterated to generate larger and larger examples.

Note the additional barb added when creating the angled-in spike, which should be removed before starting the next iteration. The procedure takes an angled-in spike (with barb) and proceeds as follows:

1. Remove the barb.
2. Glue two copies of the 'angled-in' (barb-less) spike together by merging a corresponding pair of angled-in edges and adding new angled-out edges to make an 'angled-out' spike.
3. Glue two copies of the 'angled-out' spike together by merging a corresponding pair of angled-out edges and adding new angled-in edges to make an 'angled-in' spike.
4. Add the barb.

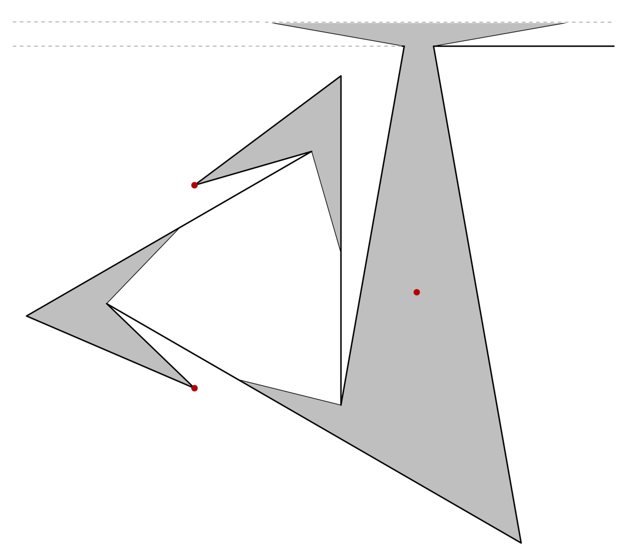

Figure 8: Here, we have a repeatable spike with $n = 10$ and $\frac{3(10)}{10} = 3$ beacons are required to guard. The (red) independent witness points and their disjoint inverse attraction regions are shaded.

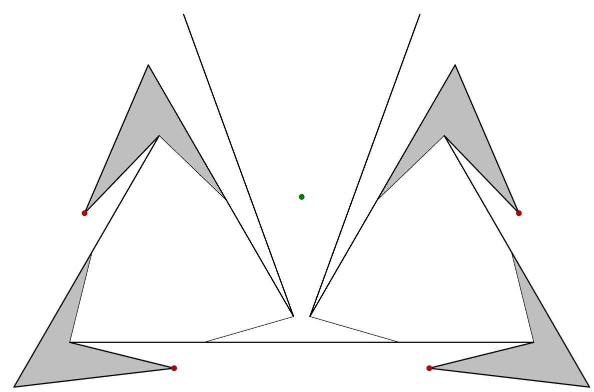

Figure 9: Two copies of the 'angled-in' spike from Figure 8 glued together to form an 'angled-out' spike.

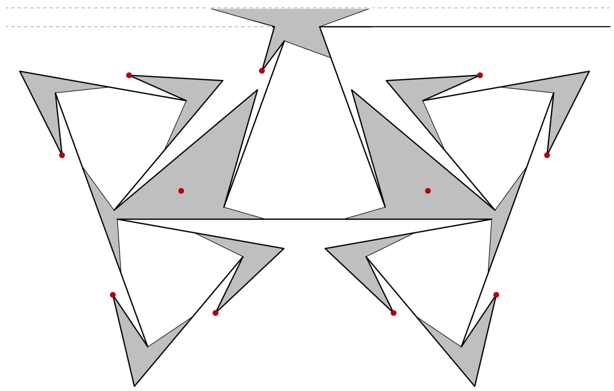

Figure 10: We took two copies of the 'angled-out' spike from Figure 9, and glued them together with new vertices to form an 'angled-in' spike. Note the extra barb added where the spike attaches to the shaft. It has 36 vertices and requires 11 beacons and may be arbitrarily repeated along the line. Iterating the procedure with this spike as input yields a new spike with 140 vertices that requires 43 beacons.

With respect to the number of vertices, we have that step 1 removes two vertices; step 2 doubles the number of vertices, then merges two edges (deleting two vertices) and adds two new vertices; step 3 doubles the number of vertices again, merges edges (deleting two vertices), then adds two new vertices; step 4 adds two new vertices. Altogether, if the initial spike has n vertices, then after the above procedure we are left with a new closed spike with $4n - 4$ vertices.

With respect to the number of independent witness points, we have that step 1 deletes one witness point, step 2 doubles the number of witness points, step 3 doubles the number of witness points then adds two new witness points, and step 4 adds a new witness point. Altogether, if the original spike has b independent witness points, the new spike has $4b-1$ independent witness points.

We can now analyze the number of beacons required to cover the above polygons, starting with the closed spike in Figure 8.

Lemma 4 *Starting with the spike depicted in Figure 8, after k iterations of the operation, we are left with an angled-in spike having $\frac{1}{3}(26 \cdot 4^k + 4)$ vertices and $\frac{1}{3}(8 \cdot 4^k + 1)$ independent witness points.*

Proof. Define a function $T(k)$ to be the number of vertices after k iterations. We start with the above spike, so $T(0) = 10$. Using the observations above, we have the recursion $T(k) = 4T(k-1) - 4$. Solving this recursion yields $T(k) = \frac{1}{3}(26 \cdot 4^k + 4)$ vertices.

Define a function $W(k)$ to be the number of independent witness points after k iterations. We start with the above spike, so $W(0) = 3$. Using the observations above, we have the recursion $W(k) = 4W(k-1) - 1$. Solving this recursion yields $W(k) = \frac{1}{3}(8 \cdot 4^k + 1)$ independent witness points. □

Using the preceding lemma we may now give an asymptotic lower bound on the number of beacons sometimes necessary to cover arbitrarily large polygons.

Theorem 5 *For an arbitrary polygon P with n vertices and h holes (possibly 0), we may need arbitrarily close to $\lfloor \frac{4n}{13} \rfloor$ beacons to cover P. Conversely, $\lfloor \frac{n+h}{3} \rfloor$ beacons are always sufficient to cover P.*

Proof. We have shown a gluing approach that gives $\frac{1}{3}(8 \cdot 4^k + 1)$ independent witness points and $\frac{1}{3}(26 \cdot 4^k + 4)$ vertices for arbitrary k. The ratio of these as k goes to infinity is $\lim_{k \to \infty} \frac{\frac{1}{3}(8 \cdot 4^k + 1)}{\frac{1}{3}(26 \cdot 4^k + 4)} = \frac{4}{13}$. In simple polygons, we can then arrange an arbitrary number of angled-in spikes in a line, whereas for polygons with h holes, h angled-out spikes may be closed with an additional vertex and then placed in a convex h-gon. Therefore, we can display a family of polygons that display a requirement for a number of beacons arbitrarily close to $\lfloor \frac{4n}{13} \rfloor$.

The proof of the always sufficient bound is derived from standard art gallery theorems: $\lfloor \frac{n}{3} \rfloor$ or $\lfloor \frac{n+h}{3} \rfloor$ beacons are always sufficient since if a set of beacons 'sees' the entire polygon they must also cover the polygon [4, 7]. □

6 Coverage in Orthogonal Polygons

In this section, we show that, unlike in arbitrary polygons, beacons seem significantly stronger than standard visibility guards in orthogonal polygons. Specifically, we show that if P is an orthogonal polygon, then $\lfloor \frac{n+4}{8} \rfloor$ beacons are sometimes necessary, while $\lfloor \frac{n}{4} \rfloor$ beacons always suffice, due to the standard art gallery bound.

The following figure displays an example of a family of orthogonal polygons that require $\lfloor \frac{n+4}{8} \rfloor$ beacons to cover.

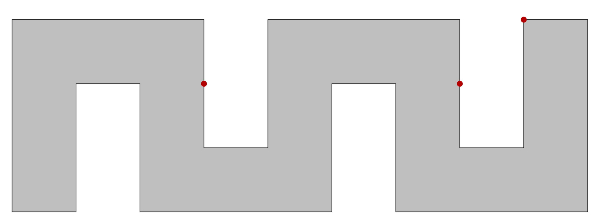

Figure 11: $\lfloor \frac{n+4}{8} \rfloor$ beacons are sometimes necessary to guard an orthogonal polygon. Here, $n = 20$ and $\frac{20+4}{8} = 3$ beacons are required to guard.

There is a gap between the sometimes necessary and always sufficient bounds proved in the above theorem.

We conjecture that the sometimes necessary bound is also sufficient.

Conjecture 2 *Given a simple orthogonal polygon P with n vertices, $\left\lfloor \frac{n+4}{8} \right\rfloor$ beacons are sometimes necessary and always sufficient to cover P.*

Acknowledgements

This research has been partially supported by the National Science Foundation (CCF-1018388) and the US-Israel Binational Science Foundation (project 2010074).

References

[1] M. Biro, J. Iwerks, I. Kostitsyna, J. S. B. Mitchell. Beacon-Based Algorithms for Geometric Routing. *Proc. of the 13th Algorithms and Data Structures Symposium (WADS 2013)*, London, Ontario, Canada, August 2013. To appear.

[2] M. Biro, J. Gao, J. Iwerks, I. Kostitsyna, J. S. B. Mitchell. Beacon-based structures in polygonal domains. In *Abstracts of the 1st Computational Geometry: Young Researchers Forum (CG:YRF 2012)*, 2012.

[3] M. Biro, J. Gao, J. Iwerks, I. Kostitsyna, J. S. B. Mitchell. Beacon-based routing and coverage. In *21st Fall Workshop on Computational Geometry (FWCG 2011)*, 2011.

[4] V. Chvátal. A combinatorial theorem in plane geometry. *Journal of Combinatorial Theory Series B*, 18:39–41, 1975.

[5] E. Györi, F. Hoffman, K. Kriegel, T. Shermer. Generalized guarding and partitioning for rectilinear polygons. *Computational Geometry: Theory and Applications*, 6(1):21–44, 1996.

[6] J. Kahn, M. Klawe, D. Kleitman. Traditional Galleries Require Fewer Watchman. *SIAM J. on Algebraic and Discrete Methods*, 4(2):194–206, 1983.

[7] F. Hoffman, M. Kaufmann, K. Kriegel The art gallery theorem for polygons with holes. *Proceedings of the 32nd Annual Symposium on Foundations of Computer Science (FOCS 1991)*, 39–48.