Cluster Connecting Problem inside a Polygon

Sanjib Sadhu* Arijit Bishnu† Subhas C. Nandy† Partha P. Goswami‡

Abstract
The cluster connecting problem inside a simple polygon is introduced in this paper. The problem is shown to be NP-complete. A $\log n$-factor approximation algorithm is proposed.

1 Introduction

The visibility graph of the vertices of a simple polygon plays important role in several geometric optimization problems. A pair of points p, p' inside a simple polygon P are said to be *mutually visible* if the line segment $[p, p']$ lies entirely inside P. The visibility graph of a simple polygon P is a graph $G = (V, E)$, where the members in V are the vertices of the polygon P. An edge $e = (v_i, v_j) \in E$ exists if and only if v_i and v_j are mutually visible. The first output sensitive algorithm for computing the visibility graph of the vertices of a simple polygon P appeared in [5]; an $O(n + k)$ time algorithm is proposed, where n is the number of vertices in P, and $k = |E|$. Later the problem is studied in depth and several efficient algorithms are proposed [3, 10, 11]. An useful variation of the visibility graph problem is studied recently in the literature [1], where a set of points Q is given inside a simple polygon P, and the objective is to compute the visibility graph of the point set Q. This problem has applications in sensor network, where a set of sensors is already deployed in a polygonal region, and the objective is to establish/check connectivity among the guards. An $O(n + m \log m \log mn + k)$ time algorithm for this problem is proposed in [1], where $m = |Q|$ and $n = |P|$. The same paper also deals with the problem of computing the range restricted visibility graph, where each point $q \in Q$ is assigned a range $\rho(q)$. A point q can see another point $q' \in Q$, $q' \neq q$ if $d(q, q') \leq \rho(q)$ and the line segment $[q, q']$ lies entirely inside P. As a result, here the visibility graph $G'(V, E)$ is a directed graph. The proposed algorithm for computing G' runs in $O(n + m^{3/2} + k)$ time and $O(n + m \log m)$ space.

Note that, both the visibility graphs with infinite and finite visibility ranges of the points in Q may not be connected. We call each connected component a cluster. In order to get a connected visibility graph, we will consider the following problem.

*National Institute of Technology, Durgapur, India
†Indian Statistical Institute, Kolkata, India
‡Institute of Radiophysics and Electronics, University of Calcutta, India

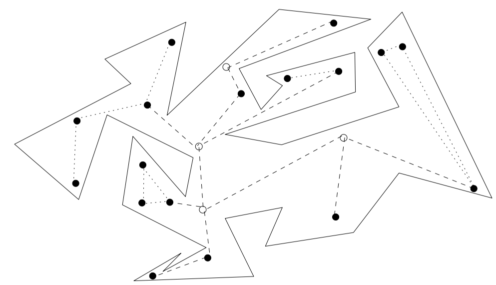

Figure 1: Example of clusters inside a polygon

Cluster connecting problem: We are given a set of m points $Q = \{q_1, q_2, \ldots, q_m\}$ inside a simple polygon P with n vertices. If the visibility graph $G = (V, E)$ corresponding to the set of points Q is *not connected*, then compute the positions of placing a set Q' of Steiner (extra) points inside P such that (i) the visibility graph of $Q \cup Q'$ is connected, and (ii) $|Q'|$ is minimum among all possible placements of the Steiner point set Q' satisfying condition (i).

The problem is useful in sensor network applications where a set of sensors are distributed to monitor a polygonal region. It may happen that the placed sensors can not form a connected network either due to the obstructions of the polygonal boundary or due to the malfunction of some sensors. The objective is to place minimum number of extra sensors to have a connected network.

The points in Q' are called the *Steiner guards*. The connected components of the visibility graph G of the points in Q are called *clusters*. If G is connected, then we have one single cluster with all the points in Q, and no Steiner guard is required; in other words, $Q' = \emptyset$. However, if G is not connected, $Q' \neq \emptyset$. An example of the *cluster connecting problem* is shown in Figure 1. Here *black* shaded points form the set Q, and the Steiner guards are shown by empty *circles*. The dotted edges are intra-cluster edges, and dashed edges are the connections of the cluster with the Steiner guards.

We show that a constrained version of the *cluster connecting problem* is NP-complete, where the Steiner guards are to be placed at the vertices of P. From now onwards, we will use the term *CCV-problem* to denote this constrained version of *cluster connecting problem*. We also show that the *CCV-problem* can be mapped to the node weighted Steiner tree problem. Thus, a $\log n$-factor approximation algorithm is easy to achieve.

A very simple version of the *CCV-problem* is studied long ago [2]. It is shown that, for a given instance of the *CCV-problem*, testing whether there exists of a solution with $|Q'| = 1$ can be solved in $O(n+m)$ time.

2 Proof of NP-completeness

Consider a simple polygon P with n vertices $\{p_1, p_2, \ldots, p_n\}$ in clockwise order. An angle $\angle p_{i-1}p_ip_{i+1}$ is said to be convex (resp. reflex) if it is less (resp. greater) than 180^o inside the polygon. In order to prove the NP-completeness of the *CCV-problem*, we choose the *cooperative guard placement problem* (*CVG-problem* in short), stated below, and propose a polynomial time reduction of an arbitrary instance of *CVG-problem* to an instance of *CCV-problem*.

CVG-problem: Given a simple polygon P, identify the minimum number of vertices $\Theta \subset P$ such that if guards are placed at the vertices of Θ then each point of the polygon P is visible to at least one member in Θ, and the visibility graph of the members in Θ is connected.

The decision version of *CVG-problem* is known to be NP-complete [9]. It follows from the NP-completeness proof of the decision version of the art gallery problem using vertex guards [8].

It is easy to show that the decision version of our *CCV-problem* is in NP, since (i) for any subset S of vertices of P, we can compute the visibility graph of $S \cup Q$ in polynomial time [1] and (ii) the connectedness of a graph can be checked in polynomial time. We now propose a polynomial time reduction of an arbitrary instance of *CVG-problem* to an instance of *CCV-problem*.

2.1 Polynomial time reduction

For the sake of simplicity, we assume that no three vertices of the polygon P are collinear. We compute the visibility graph $G = (V, E)$ of the given polygon P using the algorithm in [5]. We create an instance P' of *CCV-problem* from the polygon P by creating notches at vertices and edges of P, and placing one cluster inside each notch. A vertex v_i may satisfy at least one of the following three properties with respect to the vertices visible to it.

Π: Here an edge (v_i, v_j) exists in the visibility graph of the polygon P such that the extension of the line segment $[v_i, v_j]$ beyond v_i goes outside P.

Π^ℓ: Here an edge (v_i, v_j) exists in the visibility graph of P such that a finite extension of the line segment $[v_i, v_j]$ beyond v_i remains inside P, and v_j is to the left side of the bisector of the angle $\angle v_{i-1}v_iv_{i+1}$.

Π^r: Here an edge (v_i, v_j) exists in the visibility graph of P such that a finite extension of the line segment $[v_i, v_j]$ beyond v_i remains inside P, and v_j is to the right of the bisector of the angle $\angle v_{i-1}v_iv_{i+1}$.

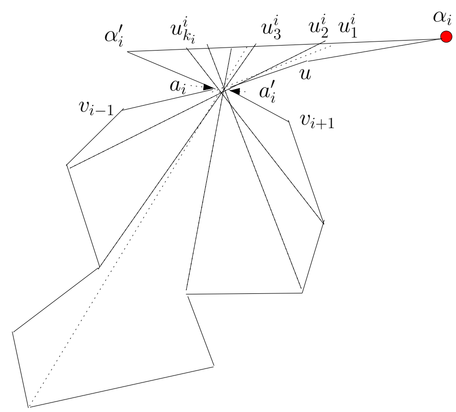

Figure 2: Modification of P at a convex vertex

Notice that, a convex vertex always satisfies only the property Π. But a reflex vertex may simultaneously satisfy one or more of the properties listed above. We separately explain the modifications needed in the polygon P for a vertex satisfying the above three properties.

The first step of our reduction algorithm is to ensure that if we draw line parallel to each edge of P at a (suitably chosen) small distance ϵ from the original edge of P outside P, the generated polygon is simple, and is an enlarged version of P. If this condition fails, we enlarge the region of the plane containing P by a constant amount (independent of n) such that the aforesaid property is satisfied. Both the checking and enlarging of P (if necessary) need $O(n)$ time.

The vertex v_i satisfies the property Π: Let the visibility edges incident at v_i be $e^i_1, e^i_2, \ldots e^i_{k_i}$, where k_i is the number of visibility edges incident at v_i; e^i_1 and $e^i_{k_i}$ correspond to the visibility edges (v_i, v_{i-1}) and (v_i, v_{i+1}) respectively.

Here, for each $j = 1, 2, \ldots k_i$, if we extend each e^i_j beyond v_i, it goes outside the polygon P. We make a very small hole $[a_i, a'_i]$ at the vertex v_i and create a region A_i bounded by a poly-chain $[a_i\alpha'_iu^i_{k_i} \ldots u^i_2u^i_1\alpha_iua'_i]$ drawn outside P. The line segment $[\alpha_i, \alpha'_i]$ is at a distance ϵ from v_i, and u^i_j is the point of intersection of the visibility edge $e^i_j = (v_i, v_j)$ and the line segment $[\alpha_i, \alpha'_i]$. The choice of the hole $[a_i, a'_i]$ at the vertex v_i is done as follows. Let v^*_j and v^{**}_j be the vertices visible to v_i corresponding to the visibility edges e^i_{j-1} and e^i_{j+1}. We join u^i_j with v^*_j and v^{**}_j, and mark its intersection with the edges (v_{i-1}, v_i) and (v_i, v_{i+1}) of P. These are referred to as the *mark points*. We compute *mark-points* for each u^i_j for $j = 1, 2, \ldots, k_i$. The point a_i (resp. a'_i) on the edge (v_{i-1}, v_i) (resp. (v_i, v_{i+1})) is closer than the closest mark point to the left (resp. right) of v_i. This ensures that for each $j = 1, 2, \ldots, k_i$, u^i_j can see no other vertex of P, except v_j. We place a single point cluster (darkly shaded) at α_i. The concave corner at u in the poly-chain bounding A_i is required to prevent the direct visibility of the cluster at α_i from the vertex

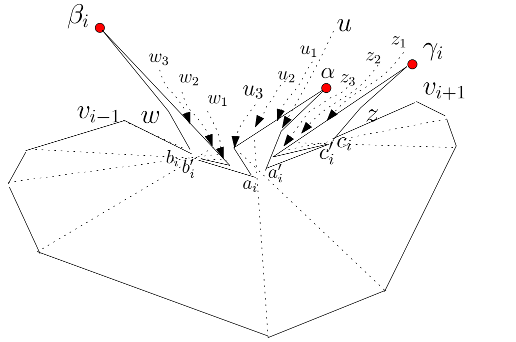

Figure 3: Modification of P at a reflex vertex

v_i (or equivalently at a_i or a'_i). Figure 2 demonstrates the updating of P around a convex vertex v_i. The same updating is done at each convex vertex of P. This modification works if v_i satisfies Π, irrespective of whether it is convex or reflex.

The vertex v_i satisfies the property Π^ℓ: Let $\mathcal{E}^i = \{e^i_1, e^i_2, \ldots, e^i_k\}$ be the visibility edges incident at v_i that satisfies the property Π^ℓ. Let b^* be the point on the edge (v_{i-1}, v_i) closest to v_{i-1}, that is visible to all the vertices corresponding to the edges in $\mathcal{E}^i$. We choose a point b on the edge (v_{i-1}, v_i) which is closer to v_i than b^*, and create a small hole $[b_i, b'_i]$ on the edge (v_{i-1}, v_i) that contains b. The width of this hole is determined in the same way as we have created the hole $[a_i, a'_i]$ around a vertex satisfying the property Π. We choose a line λ parallel to the polygonal edge (v_{i-1}, v_i) outside P and at a distance ϵ from (v_{i-1}, v_i). Let w^i_j is the point of intersection of λ and the half-line $[v_j, b)$. We create a polygonal region B_i with vertices $\{b_i w \beta_i \ldots w^i_3 w^i_2 w^i_1 \beta'_i b'_i\}$, where β_i and β'_i are also on the line λ, and $\angle b_i w \beta_i > 180^o$. The choice of the interval $[b_i, b'_i]$ says that w^i_j can not see any vertex of P other than v_j. Finally, we place a single point cluster (colored with "black") at the vertex β_i of B_i. The reason of creating the concave angle at w is same as that for the vertex u, that arrived while handing a convex vertex v_i.

The vertex v_i satisfies the property Π^r: Here, an exactly similar notch C_i is to be added on the edge (v_i, v_{i+1}) of the polygon P. The vertices of this notch will be named as $[c_i z \gamma_i z^i_1 z^i_2 z^i_3 \ldots \gamma'_i c'_i]$, where $\angle c_i z \gamma_i > 180^o$.

In Figure 3, the vertex v_i satisfies all the three properties Π, Π^ℓ and Π^r. The modifications necessary for all these properties are incorporated in the figure.

Observation 1 *If a vertex v_i satisfies Π, then in order to see the cluster at α_i, one needs to place a Steiner guard at either v_i (i.e. at a_i) or some vertex v_j (at a_j if v_j satisfies Π) such that (v_i, v_j) is an edge in the visibility graph G.*

If a vertex v_i satisfies Π^ℓ (resp. Π^r), then in order to see the cluster at β_i (resp. γ_i), one needs to place a Steiner guard at a_i or at some vertex v_j (or at a_j if v_j satisfies Π) such that (v_i, v_j) is an edge in the visibility graph G, and v_j is to the left (resp. right) of the perpendicular bisector of the angle at v_i.

Observation 2 *If (v_i, v_j) is a visibility edge in P, then if a Steiner guard is placed at a vertex u^i_j or w^i_j or z^i_j of P' (depending on whether v_i satisfies Π or Π^ℓ or Π^r with respect to v_j),*

- *it can see the vertex u^j_i (resp. w^j_i or z^j_i) if v_j satisfies Π (resp. Π^ℓ or Π^r) with respect to v_i,*
- *it can not see any other vertex in the notch(es) attached to the vertex v_j,*
- *moreover, it can not see any vertex in the notch(es) attached to any other vertex of P.*

Proof. Let the vertex v_i satisfy the property Π (resp. Π^ℓ or Π^r) with respect to the vertex v_j. The first part follows from the fact that u^i_j (resp. w^i_j or z^i_j) lies on the line $[v_i, v_j]$.

The choice of the length of $[a_i, a'_i]$ (resp. $[b_i, b'_i]$ or $[c_i, c'_i]$) depending on whether v_i satisfies Π (resp. Π^ℓ or Π^r) with respect to v_j ensures that the cone $\angle a_i u^i_j a'_i$ (resp. $\angle b_i w^i_j b'_i$ or $\angle c_i z^i_j c'_i$) does not contain any other vertex in the notch of P' attached to the vertex v_j in P. Thus, the second part follows.

Regarding the third part, if $(v_i, v_k) \notin E$, then the question of seeing u^i_j (resp. w^i_j or z^i_j, which one is appropriate) to any vertex inside A_k or B_k or C_k does not arise. $\square$

Lemma 1 *The polygon P is completely visible by K number of cooperative guards if and only if the clusters in P' can be connected by $K + n^* + n_\ell + n_r$ number of Steiner guards, where n^*, n_ℓ and n_r are number of vertices satisfying the properties Π, Π^ℓ and Π^r respectively.*

Proof. Note that, we have $n^* + n_\ell + n_r$ number of notches in P'.

[Only if part] Let K cooperative guards can see all the vertices and edges of the polygon P. None of them can see any of the clusters put at the notches attached to the vertices and edges in P. But they can see all the vertices of the polygon P. As a result they can see at least one vertex in the notch(es) of P' corresponding to each vertex in P. We need to put a Steiner guard in one of the vertices (which is visible from some cooperative guard) in each notch to see the cluster present in that notches. Thus, the clusters are connected by $K + n^* + n_\ell + n_r$ number of Steiner guards.

[If part] Suppose all the single point clusters in the polygon P' are connected using $K + n^* + n_\ell + n_r$ number of Steiner guards. If we ignore all the Steiner guards in the notches, the remaining K Steiner guards are positioned at the vertices of P, since we have solved the *CCV-problem*. Moreover, they can see all the vertices in P since no Steiner guard in a notch can see the cluster placed in the notch of some other vertex unless a Steiner guard is put there. □

2.2 Complexity result

Lemma 2 *Generation of P' from the given polygon P needs $O(n \log n + |E|)$ time, where n is the number of vertices in P and E is the set of edges in the visibility graph of P.*

Proof. Construction of the visibility graph of P needs $O(|E| + n)$ time [5]. The number of vertices inside the notch(es) attached to each vertex is at most $|E_v| + 12$, where E_v is the set of visibility edges incident at the vertex v of P. Thus the computation of the notches is linear in the number of edges of the visibility graph of P. Surely, we assume that each arithmetic operation for computing the notches in P' can be done in $O(1)$ time. The assumption is valid since ϵ is so chosen that $\frac{\epsilon}{n}$ can be stored in the machine supported real number format. □

Theorem 3 *The* CCV-problem *is NP-complete.*

3 Approximation algorithm

In this section, we propose a $\log n$-factor approximation algorithm for the *CCV-problem*. Consider an instance of the *CCV-problem*, where P is the given polygon and Q is the set of given points. We identify the clusters (connected components) in Q by executing the algorithm in [1]. Let $\mathcal{C} = \{C_1, C_2, \ldots, C_m\}$ be the set of clusters. Next, we construct a visibility graph $G = (V, E)$, where V consists of two types of nodes, namely *terminal* (V_T) and *non-terminal* (V_N). Each cluster contributes a node in V_T, and each vertex in P contributes a node in V_N. The edge set $E = E_1 \cup E_2$, where E_1 = set of edges of the visibility graph of P, and each edge in E_2 connectes a vertex in V_T with a vertex in V_N. An edge $(v_\mu, v_\nu) \in E_2$ implies that the polygonal vertex corresponding to the node $v_\nu \in V_N$ can see at least one point in the cluster corresponding to the node $v_\mu \in V_T$. This can easily be computed using the same algorithm [1] with the point set $Q \cup \mathcal{P}$, where $\mathcal{P}$ is the set of vertices of the polygon P.

We attach an weight "1" to each node in $V_T \cup V_N$. Now, the *CCV-problem* reduces to finding a Steiner tree of minimum cost, where the cost of a Steiner tree T is the number of terminal nodes and Steiner nodes in the tree T. Since all terminal nodes are present in T, the minimum cost Steiner tree corresponds to one having minimum number of Steiner nodes. Guha and Khullar [4] proposed a polynomial time $\log k$-factor approximation algorithm for the node weighted Steiner tree problem where each node is assigned a weight "1", and K is the number of terminal nodes in the graph. We use this algorithm for our *CCV-problem* to get a solution with cost at most $\log K \times OPT$, where K is the number of clusters originally present in Q and OPT is the minimum number of Steiner guards needed.

4 Conclusion

The cluster connecting problem is introduced in this paper. The problem is shown to be NP-complete. A $\log n$-factor approximation algorithm is proposed using the approximation algorithm for the node weighted Steiner tree problem. It seems that getting a constant factor approximation algorithm is possible exploiting the geometric properties of the problem.

References

[1] B. Ben-Moshe, O. Hall-Holt, M. J. Katz and J. S. B. Mitchell, *Computing the visibility graph of points within a polygon*, Proc. 20th. Annual ACM Symposium on Computational Geometry, pp. 27-35, 2004.

[2] S. K. Ghosh, *Computing a viewpoint of a set of points inside a polygon*, Proc. Foundation of Software Technology and Theoretical Computer Science, pp. 18-29, 1988.

[3] S. K. Ghosh and D. M. Mount, *An output-sensitive algorithm for computing visibility graphs*, SIAM Journal on Computing, vol. 20, pp. 888-910, 1991.

[4] S. Guha and S. Khuller, *Improved methods for approximating node weighted steiner trees and connected dominating sets*, Information and Computation, vol. 150, pp. 57-74, 1999.

[5] J. Hershberger, *An optimal visibility graph algorithm for triangulated simple polygons*, Algorithmica, vol. 4, pp. 141155, 1989.

[6] P. N. Klein, R. Ravi, *A nearly best-possible approximation algorithm for node-weighted steiner trees*, Journal of Algorithms, vol. 19, pp. 104-115, 1995.

[7] D. T. Lee *Visibility of a simple polygon*, Computer Vision, Graphics and Image Processing, vol. 22, pp. 207-221, 1983.

[8] D. T. Lee and A. K. Lin, *Computational complexity of art gallery problems*, IEEE Trans. on Information Theory, vol. IT32, pp. 276-282, 1986.

[9] B. C. Liaw, N. F. Huang, R. C. T. Lee, *The minimum cooperative guards problem on k-spiral polygons*, Proc. Canadian Conf. on Computational Geometry, pp. 97-101, 1993.

[10] M. H. Overmars and E. Welzl, *New methods for computing visibility graphs*, Proc. 4th Annual ACM Symposium on Computational Geometry, pp. 164-171, 1988.

[11] M. Pocchiola and G. Vegter, *Computing the visibility graph via pseudo-triangulation*, Proc. 11th. Annual ACM Symp. on Computational Geometry, pp. 248-257, 1995.