How to Place Efficiently Guards and Paintings in an Art Gallery

Christodoulos Fragoudakis * Euripides Markou †
Stathis Zachos ‡*

(cfrag, emarkou, zachos)@cs.ntua.gr

November 13, 2005

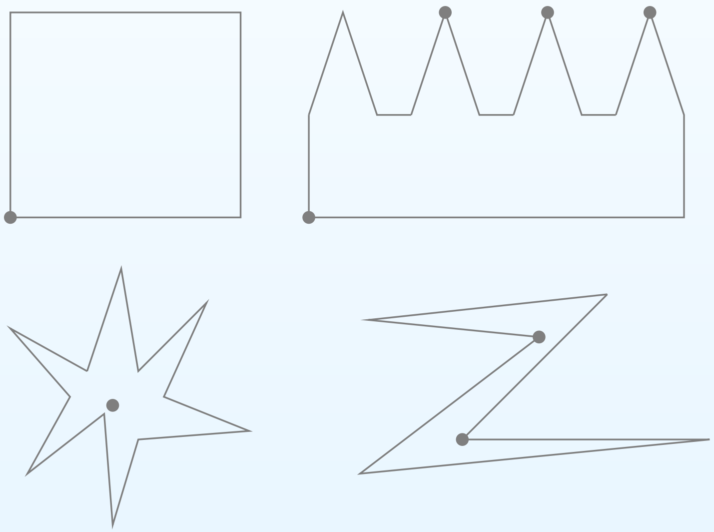
* Computer Science, ECE, National Technical University of Athens
† Department of Informatics and Telecommunications, National University of Athens
‡ CIS Department, Brooklyn College, CUNY, USA

Outline

Introduction

The Art Gallery (AG) Problem

A polygon is given and the goal is to place as few as possible guards in the polygon, so that the polygon is covered.



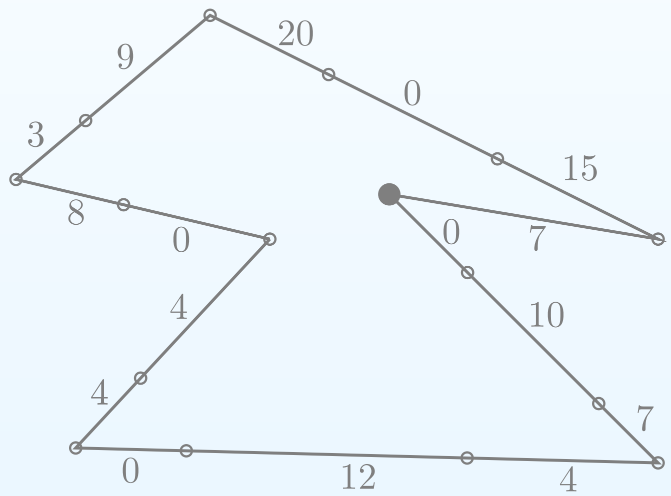
In a variation a number of guards is given and the goal is to cover as many parts of the polygon as possible.

Variations of the AG Problem

- Polygons with holes.
- The points that must be covered lie:
  - on the boundary of the polygon and of its holes,
  - inside the polygon.
- The guards can be placed:
  - on vertices,
  - inside edges,
  - generaly inside the polygon.
- The guards are realized as:
  - vertices (vertex guards),
  - whole edges (edge guards).

Our Problem

How to place exhibits, like paintings, and guards in an art gallery so that the total value of guarded paintings is a maximum?

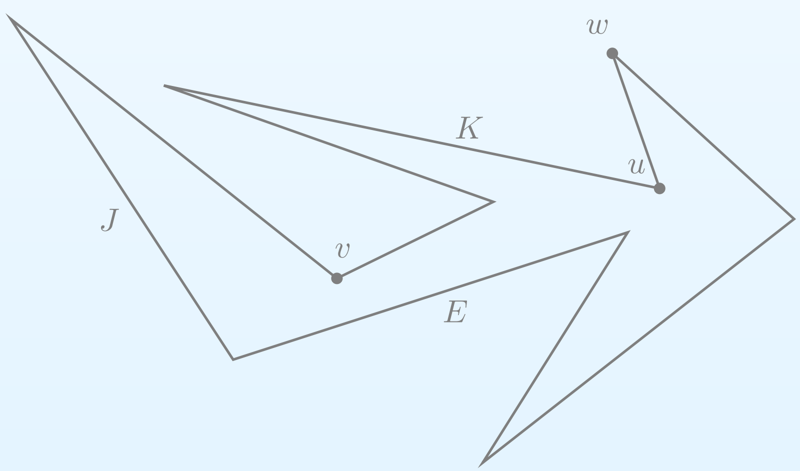


This is the *MAXIMUM VALUE VERTEX GUARD with PAINTING PLACEMENT (MVVGPP)* problem which has variations where the polygon might have holes, or the guards are realized as whole edges (edge guards).

Visibility Predicates

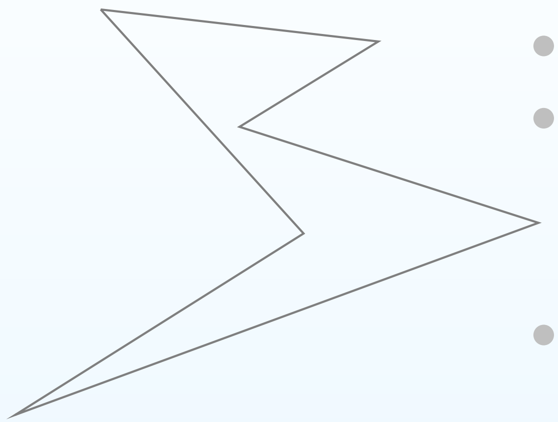
Let P be a polygon, $v, u \in P$ points in P and $E, J, K \subseteq P$ edges of P. We define the following visibility predicates:

- $sees(v, u) : \forall x \in \overline{vu} : x \in P$
- $oversees(v, E) : \forall x \in E : sees(v, x)$
- $oversees(E, J) : \forall x \in J : \exists y \in E : sees(x, y)$
- $watches(u, J) : \exists x \in J : sees(u, x)$
- $watches(E, K) : \exists x \in K : \exists y \in E : sees(x, y)$



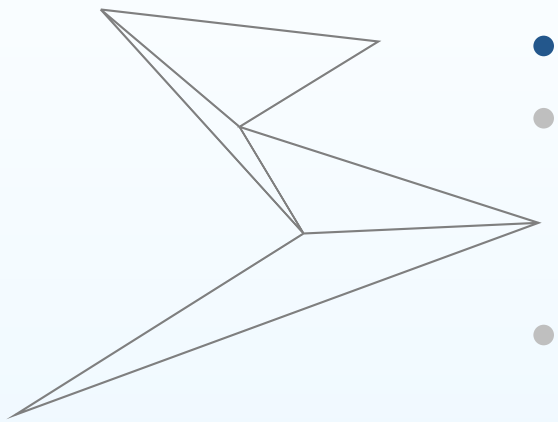
Finest Visibility Segmentation

A Simple Construction



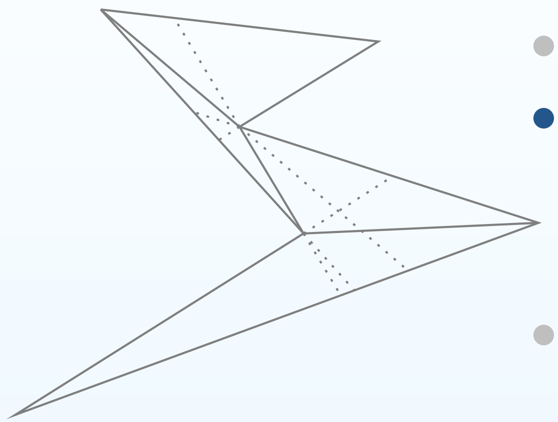
- Start with an arbitrary polygon.
- Find the Visibility Graph.
- Extend the visibility graph's edges, inside and up to the boundary of the polygon.
- $O(n^2)$ points are generated that include all vertices of the polygon.

A Simple Construction



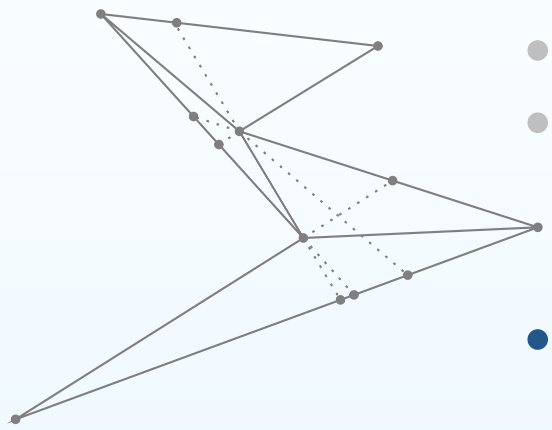
- Start with an arbitrary polygon.
- Find the Visibility Graph.
- Extend the visibility graph's edges, inside and up to the boundary of the polygon.
- $O(n^2)$ points are generated that include all vertices of the polygon.

A Simple Construction



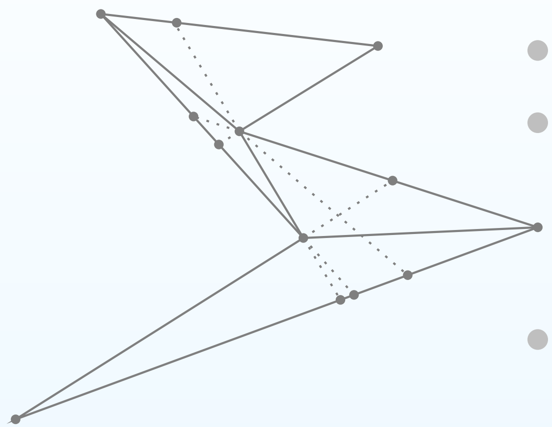
- Start with an arbitrary polygon.
- Find the Visibility Graph.
- Extend the visibility graph's edges, inside and up to the boundary of the polygon.
- $O(n^2)$ points are generated that include all vertices of the polygon.

A Simple Construction



- Start with an arbitrary polygon.
- Find the Visibility Graph.
- Extend the visibility graph's edges, inside and up to the boundary of the polygon.
- $O(n^2)$ points are generated that include all vertices of the polygon.

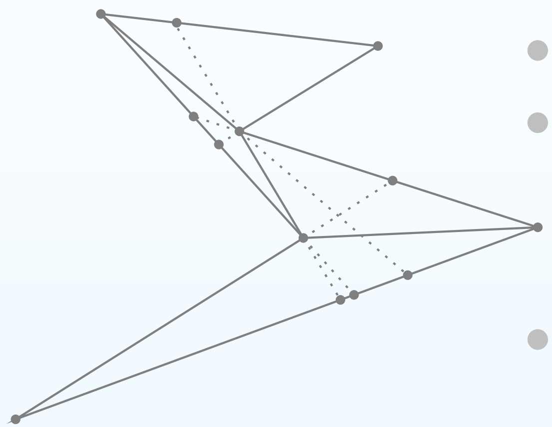
A Simple Construction



- Start with an arbitrary polygon.
- Find the Visibility Graph.
- Extend the visibility graph's edges, inside and up to the boundary of the polygon.
- $O(n^2)$ points are generated that include all vertices of the polygon.

We managed to discretize the boundary of the polygon with respect to visibility: we obtained a set FVS of points on the boundary of P such that any segment (a, b) defined by consecutive FVS points (FVS segment) cannot be **only partly** visible from a vertex or an edge.

A Simple Construction



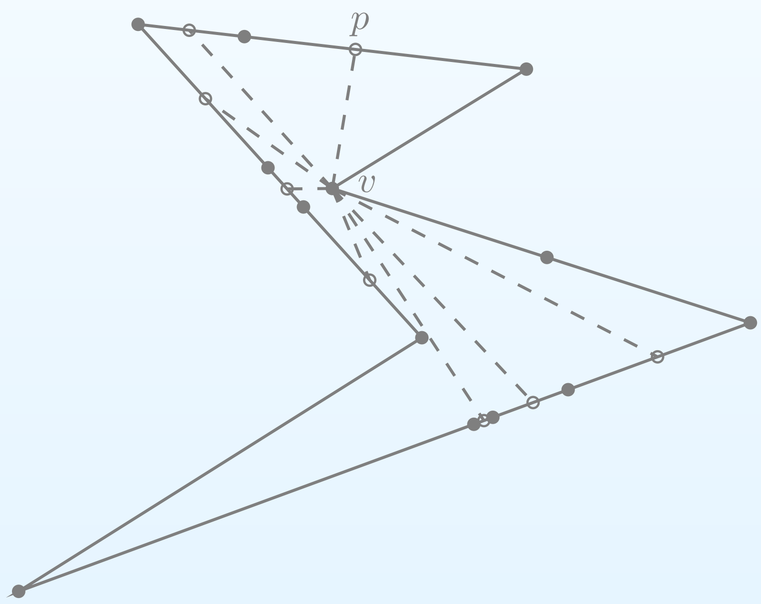
- Start with an arbitrary polygon.
- Find the Visibility Graph.
- Extend the visibility graph's edges, inside and up to the boundary of the polygon.
- $O(n^2)$ points are generated that include all vertices of the polygon.

We managed to discretize the boundary of the polygon with respect to visibility: we obtained a set FVS of points on the boundary of P such that any segment (a, b) defined by consecutive FVS points (FVS segment) cannot be **only partly** visible from a vertex or an edge.

Theorem: Any vertex (edge) of P sees a FVS segment if and only if watches the FVS segment.

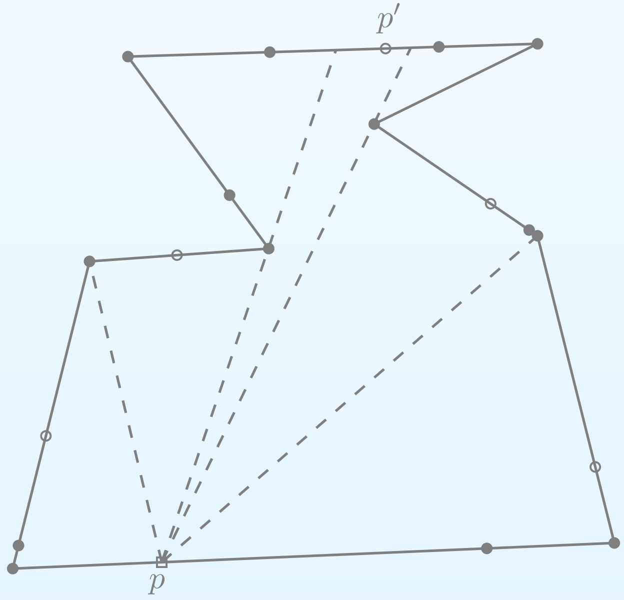
The *FVS(v)* set

In order to find the set $FVS(v)$ of all overseen FVS segments from a polygon vertex v, it suffices to pick an arbitrary point p in every FVS segment and check if vp is everywhere inside the polygon.



The *FVS(e)* set

In order to find the set $FVS(e)$ of all overseen FVS segments from a polygon edge e, it suffices to pick an arbitrary point p in every FVS segment and check if there exists a point $p' \in e$ such that pp' is everywhere inside the polygon.



The MVVGPP Problem

The Multiple Knapsack Problem

We are given a set of n items and m knapsacks such that each item i has a profit $p(i)$ and a size $s(i)$, and each knapsack j has a capacity $c(j)$. The goal is to find a subset of items of maximum profit such that they have a feasible packing in the knapsacks.

- The MK problem is a generalization of both the knapsack and bin packing problems and is strongly NP-hard.
- 2-approximation algorithm (Shmoys and Tardos, 1993)
- PTAS (Chekuri and Khanna, 2000)

The MVVGPP Problem

Given is a polygon P (art gallery), a set of ordered pairs (x, y) (paintings with length x and value y), and an integer $k > 0$.
The goal of the *MAXIMUM VALUE VERTEX GUARD with PAINTING PLACEMENT PROBLEM* is to place k vertex guards as well as place paintings on the boundary of P so that the total weight of the overseen paintings is maximum.

The MVVGPP Problem

Given is a polygon P (art gallery), a set of ordered pairs (x, y) (paintings with length x and value y), and an integer $k > 0$.
The goal of the *MAXIMUM VALUE VERTEX GUARD with PAINTING PLACEMENT PROBLEM* is to place k vertex guards as well as place paintings on the boundary of P so that the total weight of the overseen paintings is maximum.

Proposition: The MVVGPP problem is NP-hard

The MVVGPP Problem

Given is a polygon P (art gallery), a set of ordered pairs (x, y) (paintings with length x and value y), and an integer $k > 0$.
The goal of the *MAXIMUM VALUE VERTEX GUARD with PAINTING PLACEMENT PROBLEM* is to place k vertex guards as well as place paintings on the boundary of P so that the total weight of the overseen paintings is maximum.

Proposition: The MVVGPP problem is NP-hard

Proof: The decision version of *MINIMUM VERTEX GUARD* problem for a polygon P reduces to the corresponding decision version of *MVVGPP* problem for the same polygon P:

The MVVGPP Problem

Given is a polygon P (art gallery), a set of ordered pairs (x, y) (paintings with length x and value y), and an integer $k > 0$.
The goal of the *MAXIMUM VALUE VERTEX GUARD with PAINTING PLACEMENT PROBLEM* is to place k vertex guards as well as place paintings on the boundary of P so that the total weight of the overseen paintings is maximum.

Proposition: The MVVGPP problem is NP-hard

Proof: The decision version of *MINIMUM VERTEX GUARD* problem for a polygon P reduces to the corresponding decision version of *MVVGPP* problem for the same polygon P: construct the FVS set and take a painting for each FVS segment that has length the length of the segment and value also the length of the segment. The total value that the *MVVGPP* problem asks for is the total sum of values of the segments.

An Approximation Algorithm for Vertex Guards

- Set $SOL = \emptyset$ and calculate the FVS points

An Approximation Algorithm for Vertex Guards

- Set $SOL = \emptyset$ and calculate the FVS points
- For every $v \in V(P)$ calculate the set $FVS(v)$

An Approximation Algorithm for Vertex Guards

- Set $SOL = \emptyset$ and calculate the FVS points
- For every $v \in V(P)$ calculate the set $FVS(v)$
- During each iteration, for any vertex that hasn't been assigned a guard yet:

An Approximation Algorithm for Vertex Guards

- Set $SOL = \emptyset$ and calculate the FVS points
- For every $v \in V(P)$ calculate the set $FVS(v)$
- During each iteration, for any vertex that hasn't been assigned a guard yet:
  - calculate the set of the visible segments not previously overseen: $FVS(v) \setminus SOL \cap FVS(v)$.

An Approximation Algorithm for Vertex Guards

- Set $SOL = \emptyset$ and calculate the FVS points
- For every $v \in V(P)$ calculate the set $FVS(v)$
- During each iteration, for any vertex that hasn't been assigned a guard yet:
  - calculate the set of the visible segments not previously overseen: $FVS(v) \setminus SOL \cap FVS(v)$.
  - For every such set the *MULTIPLE KNAPSACK* problem is solved (the knapsacks are the segments in the set and the capacity of a knapsack is the length of the corresponding segment). The vertex that maximizes the total value of the fitted paintings is then assigned a guard.

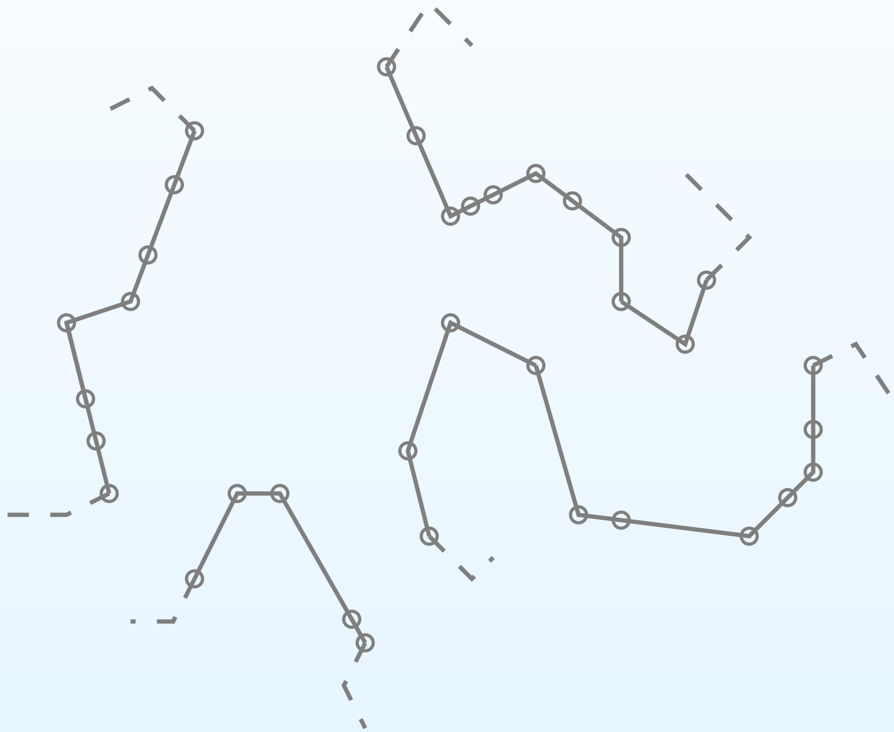
An Approximation Algorithm for Vertex Guards

- Set $SOL = \emptyset$ and calculate the FVS points
- For every $v \in V(P)$ calculate the set $FVS(v)$
- During each iteration, for any vertex that hasn't been assigned a guard yet:
  - calculate the set of the visible segments not previously overseen: $FVS(v) \setminus SOL \cap FVS(v)$.
  - For every such set the *MULTIPLE KNAPSACK* problem is solved (the knapsacks are the segments in the set and the capacity of a knapsack is the length of the corresponding segment). The vertex that maximizes the total value of the fitted paintings is then assigned a guard.
  - Update SOL: add the new FVS segments along with the fitted paintings.

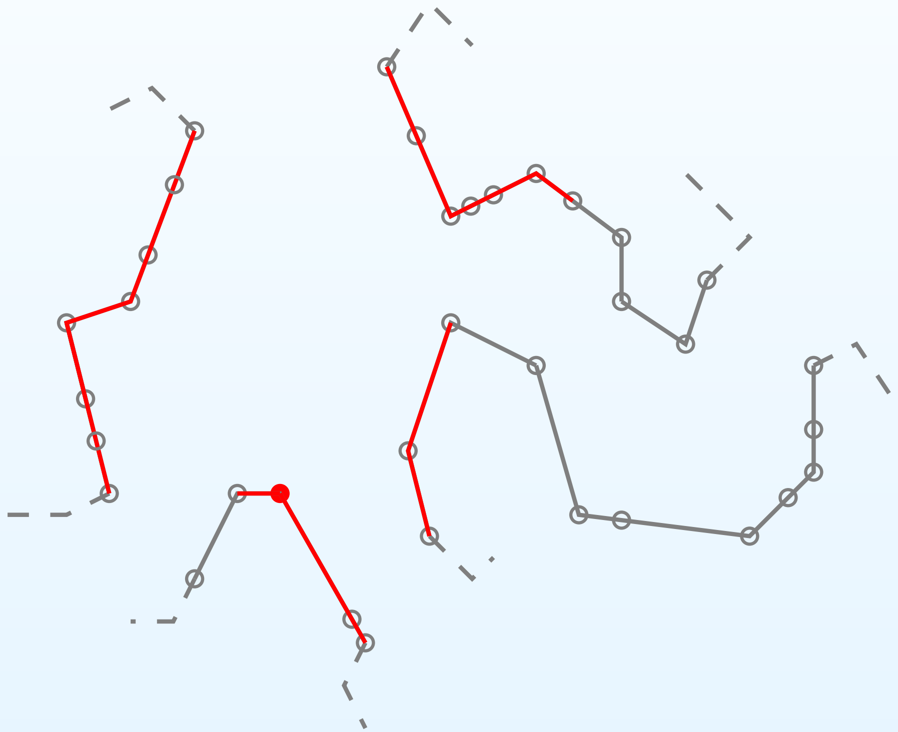
An Approximation Algorithm for Vertex Guards

- Set $SOL = \emptyset$ and calculate the FVS points
- For every $v \in V(P)$ calculate the set $FVS(v)$
- During each iteration, for any vertex that hasn't been assigned a guard yet:
  - calculate the set of the visible segments not previously overseen: $FVS(v) \setminus SOL \cap FVS(v)$.
  - For every such set the *MULTIPLE KNAPSACK* problem is solved (the knapsacks are the segments in the set and the capacity of a knapsack is the length of the corresponding segment). The vertex that maximizes the total value of the fitted paintings is then assigned a guard.
  - Update SOL: add the new FVS segments along with the fitted paintings.
- Return the total value of all the fitted paintings in SOL

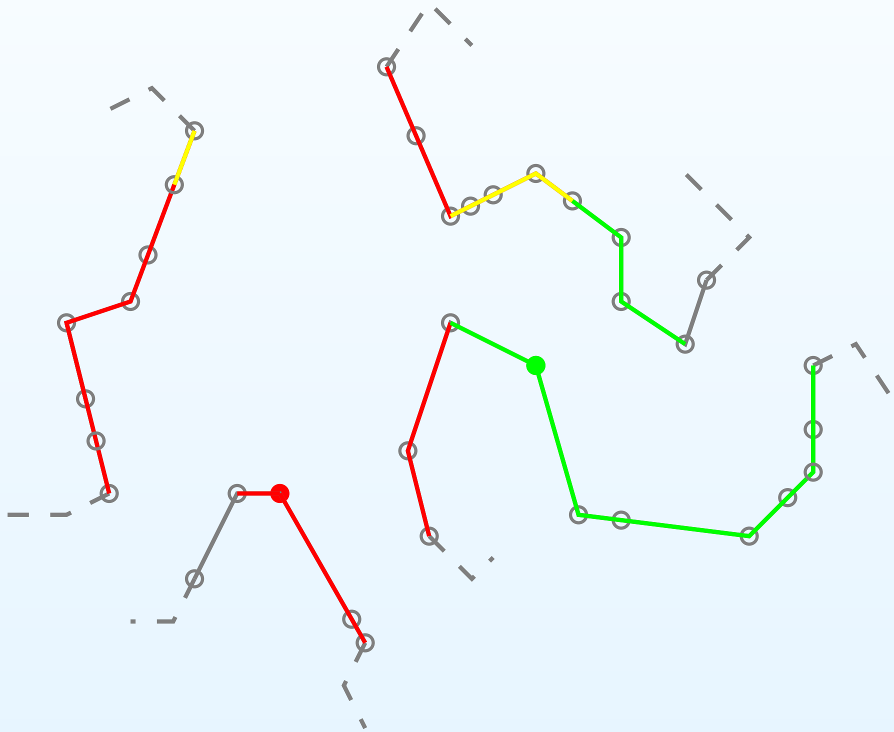
Example



Example



Example



Remarks

- Let $W(SOL)$ the value returned from our algorithm and $W(OPT)$ the value of the collection of the set of paintings in an optimal solution. It holds: $W(SOL) > (1 - \frac{1}{e^a})\, W(OPT)$, where a is the approximation of the *MK* problem.

Remarks

- Let $W(SOL)$ the value returned from our algorithm and $W(OPT)$ the value of the collection of the set of paintings in an optimal solution. It holds: $W(SOL) > (1 - \frac{1}{e^a})\, W(OPT)$, where a is the approximation of the *MK* problem.
- As $a \longrightarrow 1$, due to the existence of the PTAS for the MK problem, $1 - \frac{1}{e^a} \longrightarrow 0.633$

Remarks

- Let $W(SOL)$ the value returned from our algorithm and $W(OPT)$ the value of the collection of the set of paintings in an optimal solution. It holds: $W(SOL) > (1 - \frac{1}{e^a})\, W(OPT)$, where a is the approximation of the *MK* problem.
- As $a \longrightarrow 1$, due to the existence of the PTAS for the MK problem, $1 - \frac{1}{e^a} \longrightarrow 0.633$

- Similar to *MVVGPP* problem the *MAXIMUM VALUE EDGE GUARD with PAINTING PLACEMENT* problem is NP-hard.
- The same algorithm applies except from the calculation of the $FVS(e)$ set.
- Our results are also applicable for polygons with holes.

Conclusion

References

[1] Lee, D., Lin, A., Computational complexity of art gallery problems, IEEE Trans. Inform. Theory 32, 276-282, 1986.

[2] Shmoys, D., Tardos, E., An approximation algorithm for the generalized assignment problem, Mathematical Programming A, 62:461-74, 1993.

[3] Eidenbenz, S., Inapproximability Results for Guarding Polygons without Holes, Lecture notes in Computer Science, Vol. 1533 (ISAAC'98), p. 427-436, 1998.

[4] Ghosh, S., Approximation algorithms for Art Gallery Problems, Proc. of the Canadian Information Processing Society Congress, pp. 429-434, 1987.

[5] Hochbaum, D., Approximation Algorithms for NP-Hard Problems, PWS Publishing Company, 1996.

[6] Hochbaum, D., Approximating Covering and Packing Problems in: Approximation Algorithms for NP-Hard Problems (ed. Dorit Hochbaum), pp. 94-143, PWS Publishing Company, 1996.

[7] Avis, D., Rappaport, D., Computing the largest empty convex subset of a set of points, Proc. 1st Ann. ACM Symposium Computational Geometry, pp. 161-167, 1985.

[8] Dobkin, D., Edelsbrunner, H., Overmars, H., Searching for Empty Convex Polygons, Algorithmica 5, pp. 561-571, 1990.

[9] Edelsbrunner, H., Guibas, L., Topologically sweeping an arrangement, J. Comput. System Sci. 38, pp. 165-194, 1989.

[10] Chekuri, C., Khanna, S., A PTAS for the Multiple Knapsack problem, Proc. 11th ACM Symp on Discrete Algorithms, 213-222, 2000.