Maximizing the Guarded Boundary of a Dynamic Art Gallery

Mihir Patel

May 9, 2025

Abstract

We study several boundary-maximization variants of the classic Art Gallery problem, where the goal is to place guards at vertices of a simple polygon in order to maximize the portion of the boundary that is fully visible. Specifically, we consider two variants: Maximum Length Vertex Guard, which seeks to maximize the total boundary length visible to k vertex guards, and Maximum Value Vertex Guard, which generalizes this to weighted boundary segments. For both problems, we prove that the objective functions are monotone and submodular, allowing the application of the standard greedy algorithm to obtain a $(1 - 1/e)$-approximation in $O(kn^2)$ time, improving upon the prior $O(k^2n^2)$-time algorithm. We also show that this approximation factor is the best possible, unless P=NP. We also introduce a new variant, Budgeted Maximum Value Vertex Guard, which institutes vertex costs and imposes a budget constraint rather than a cardinality limit for guard placement. We show that several intuitive greedy strategies fail to provide approximation guarantees for this setting. Drawing on techniques from budgeted Max k-Coverage literature, we propose an algorithm that may yield a $\frac{1}{2}(1 - 1/e)$-approximation for Budgeted Maximum Value Vertex Guard, opening the door for further theoretical advances in cost-constrained guarding strategies.

1 Introduction

In this paper, we study a variant of the classic Art Gallery problem, where the goal is not to minimize the number of guards required to cover a region, but instead to maximize the portion of the polygon's boundary that is guarded by a budgeted number of guards. Specifically, given a simple polygon P and $k \in \mathbb{N}$, we want to find the k vertex guards which maximize the length of the boundary of P that is watched. We define the set of vertices of P as V_P, and $L(S)$ as the length of the boundary seen by the set of vertex guards/vertices S. Note that $L(S)$ is necessarily at most the perimeter of P.

Maximum Length Vertex Guard

Input: A simple polygon P and a positive integer $k \in \mathbb{N}$.
Task: Find a set of vertices $S \subseteq V_P$ of size at most k such that $L(S)$ is maximized.

We also study a weighted version of the problem, where the polygon P is composed of (possibly collinear) boundary segments, each assigned a non-negative integer weight. An example of such a polygon is shown in Figure 1. This models a more realistic art gallery, where some sections of the wall (e.g., those holding more valuable paintings) are more important to guard than others, especially when the constrained number of guards may prevent us from guarding the entire gallery. In this variant, the goal is to maximize the total weight of boundary segments that are *fully* visible to the vertex guards. We define $W(S)$ as the weighted analog of $L(S)$: the sum of the weights of all boundary segments that are completely seen by the guard set S.

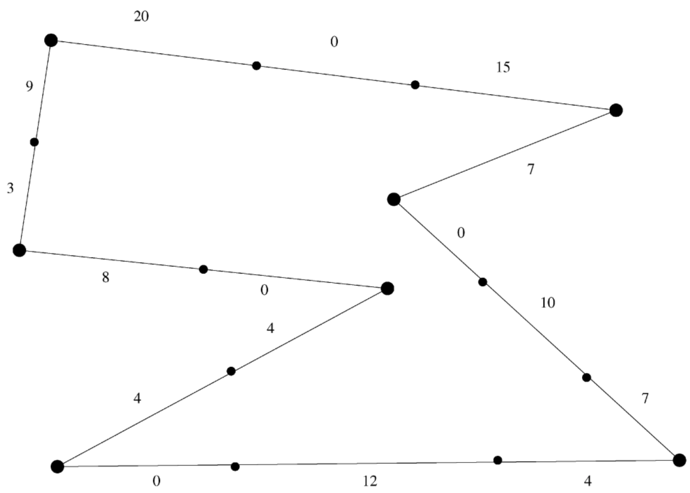


Figure 1: A polygon composed of weighted line segments, the input for Maximum Value Vertex Guard.

Maximum Value Vertex Guard

Input: A weighted polygon P and a positive integer $k \in \mathbb{N}$.
Task: Find a set of vertices $S \subseteq V_P$ of size at most k such that $W(S)$ is maximized.

Finally, we consider a budgeted variant of Maximum Value Vertex Guard, where each vertex has an associated placement cost. This is in contrast to Maximum Value Vertex Guard, where the constraint was the number of guards, not the total guard cost. Depending on the setting, it may be more difficult or expensive to position a guard at certain locations of the art gallery, and this problem formulation captures that added difficulty. Now, instead of selecting up to k guards, we are given a total budget B, and each vertex v has a cost $c(v) \leq B$. The goal is to choose a set of guards whose total cost does not exceed B, while maximizing the total weight of boundary segments they fully guard.

Budgeted Maximum Value Vertex Guard

Input: A weighted polygon P and a positive integer $B \in \mathbb{N}$.
Task: Find a set of vertices $S \subseteq V_P$ with $\sum_{s \in S} c(s) \leq B$ such that $W(S)$ is maximized.

1.1 Related Works

The Art Gallery problem, originally posed by Victor Klee to Václav Chvátal in 1973, is a foundational problem in computational geometry with numerous variations. In its original form, the problem asks for the minimum number of guards required to fully guard the interior of a simple polygon P with n vertices. A comprehensive overview of early results is provided in [11], which includes both foundational combinatorial results and a survey of the state of the field (as of 1987).

Subsequent research has explored the problem through the lens of approximation algorithms. It is now known that the Art Gallery problem is $O(\log n)$-approximable and APX-complete, meaning that no polynomial-time approximation scheme (PTAS) exists unless $\mathsf{P} = \mathsf{NP}$. Recent work, such as [3], introduces novel approximation strategies and includes a comprehensive literature review of prior approximation results for the Art Gallery problem. Additional inapproximability results are presented in [1], which considers restricted visibility models where guards have a limited field of view.

This paper focuses on a boundary-maximization variant of the Art Gallery problem, where the objective is to maximize the length of the polygon boundary that is fully guarded, rather than minimizing the number of guards. This formulation was introduced by Fragoudakis et al. in a series of papers [4, 7, 6], where they introduce and study both the Maximum Length Vertex Guard and Maximum Value Vertex Guard problems. In [7], they show that both problems are APX-complete. In [4], they present a $(1 - 1/e)$-approximation algorithm for maximizing the vertex-guarded interior area of a polygon, based on partitioning the polygon into what they call the Finest Visibility Segmentation. Their algorithm runs in $O(k^2n^2)$ time.

Our work also draws heavily on the relationship between the Set Cover problem and various Art Gallery formulations. Specifically, we use techniques from the Max k-Coverage problem to inform our approach to Maximum Length Vertex Guard. [9] analyzes the Max k-Coverage problem and provides tight approximation bounds. These connections also position our work within the broader study of optimizing monotone submodular functions under cardinality constraints, where $(1 - 1/e)$-approximation factors are both the best known and provably optimal (assuming $\mathsf{P} \neq \mathsf{NP}$) for cardinality-constrained maximization [2, 5]. In addition to cardinality constraints, we also study a budgeted version of Maximum Value Vertex Guard, where each vertex has an associated cost. This leads us to consider the budgeted Max k-Coverage, which was introduced in [10] and also admits a $(1 - 1/e)$-approximation algorithm under a total cost constraint.

1.2 Our Contributions

For both Maximum Length Vertex Guard and Maximum Value Vertex Guard, we improve upon the $(1 - 1/e)$-approximation algorithm with $O(k^2n^2)$ running time from [4, 7, 6] by proving that the objective function in each case is monotone and submodular. This structural property allows us to apply the standard greedy algorithm for submodular maximization, yielding a $(1 - 1/e)$-approximation that runs in $O(kn^2)$ time. We further show that this approximation ratio is optimal unless $\mathsf{P} = \mathsf{NP}$, aligning with their prior result that both problems are APX-complete.

We also introduce a budgeted variant of Maximum Value Vertex Guard, called Budgeted Maximum Value Vertex Guard, where guards have associated costs and the goal is to maximize the total weight guarded under a fixed budget, rather than a cardinality constraint. We demonstrate that three natural greedy strategies fail to form an approximation:

1. Greedily selecting the vertex which provides the highest marginal gain in total weight guarded.
2. Greedily selecting the cheapest vertex.
3. Greedily selecting the vertex which provides the highest ratio of marginal gain in total weight guarded to vertex cost.

Using these counterexamples as motivation, we propose an algorithm inspired by [10] that may achieve a $\frac{1}{2}(1 - 1/e)$-approximation for Budgeted Maximum Value Vertex Guard.

2 Submodularity of Maximum Value Vertex Guard

In this section, we prove that the objective functions of both Maximum Length Vertex Guard and Maximum Value Vertex Guard are monotone and submodular. This then implies that greedily maximizing the objective will yield a $(1 - \frac{1}{e})$ approximation for both problems. We also provide a brief analysis of the running time of these greedy algorithms, which is an improvement on the approximation algorithms proposed for these problems in [4, 7, 6].

We start with the objective function of Maximum Length Vertex Guard, L. Recall that given a simple polygon P and a set of vertex guards $S \subseteq V_P$, $L(S)$ denotes the length of the guarded boundary of P, and is necessarily less than or equal to the permiter of P. It is fairly immediate to see that L is monotone.

Observation 1. *For any $S \subseteq T \subseteq V$, $L(S) \leq L(T)$.*

Proof. Consider a point p on the portion of the boundary of the polygon counted by $L(S)$, then p must be seen by some $v \in S$. As $S \subseteq T$, $v \in T$ as well, implying p is counted by $L(T)$. Every point on the boundary counted by $L(S)$ is also counted by $L(T)$, and thus $L(S) \leq L(T)$. □

We can also show that L is submodular with a little bit more work.

Claim 1. *For any $S \subseteq T \subseteq V$ and $v \notin T$, $L(S \cup \{v\}) - L(S) \geq L(T \cup \{v\}) - L(T)$.*

Proof. First observe that, by Observation 1, $L(S \cup \{v\}) - L(S) \geq 0$, meaning this claim is trivial if $L(T \cup \{v\}) - L(T) \leq 0$. So we only consider the case where $L(T \cup \{v\}) - L(T) > 0$, or in other words, there is some portion of the boundary of the polygon guarded by $T \cup \{v\}$, but not T. Then for every point p in this portion, p is visible to v, but not T. As $S \subseteq T$, by Observation 1, p is visible to S either.

In short, we know that p is visible to $S \cup \{v\}$ but not S, and as this is true for every point on the boundary seen by the set of guards $T \cup \{v\}$, but not T, $L(S \cup \{v\}) - L(S) \geq L(T \cup \{v\}) - L(T)$, as desired. □

These two facts also lift quickly to the objective function of Maximum Value Vertex Guard, W. Given a simple polygon P made up of weighted segments and a set of vertex guards $S \subseteq V_P$, $W(S)$ denotes the total weight of the guarded boundary of P. Note that a weighted segment must be *completely* visible to a set of guards to be considered "guarded" and contribute to the total weight. The proofs of monotonicity and submodularity for W proceed exactly the same as those of L (we only need to swap L and W).

Observation 2. *For any $S \subseteq T \subseteq V$, $W(S) \leq W(T)$.*

Claim 2. *For any $S \subseteq T \subseteq V$ and $v \notin T$, $W(S \cup \{v\}) - W(S) \geq W(T \cup \{v\}) - W(T)$.*

Equipped with these properties of L and W, we can quickly derive greedy algorithms for Maximum Length Vertex Guard and Maximum Value Vertex Guard. Start with an empty solution set S. While the size of S is less than k, find the vertex v which provides L with the most marginal gain. That is, the vertex v which maximizes $L(S \cup \{v\}) - L(S)$. Then add v to S and repeat, until $|S| = k$. Then, because L and W are both monotone submodular, we get the following facts about maximizing monotone submodular functions under cardinality constraints (i.e. the type of problem Maximum Length Vertex Guard and Maximum Value Vertex Guard both are).

Theorem 1 ([2])**.** *The greedy algorithm is a $(1-1/e)$-approximation algorithm for Maximum Length Vertex Guard.*

Theorem 2 ([5])**.** *For any $\epsilon > 0$, there is no $(1 - 1/e + \epsilon)$-approximation algorithm for Maximum Length Vertex Guard, unless* P=NP*.*

The same approximation guarantees hold for the greedy algorithm applied to Maximum Value Vertex Guard, which maximizes the marginal gain of W. As for the runtime of these algorithms, they proceed in k iterations, selecting one vertex per step. In each iteration,

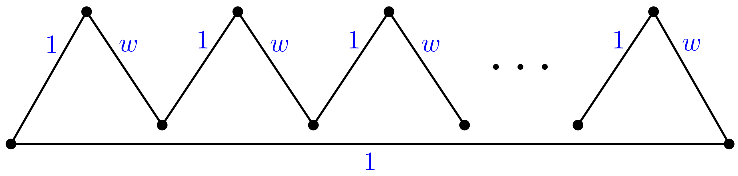


Figure 2: A polygon composed of w cones, where the right edge of each cone has weight w and all other edges in the polygon have weight 1. The apicies of each cone have cost w, and everything else has cost w^2. We are given a budget of $B = w^2$.

we evaluate the marginal gain for up to n candidate vertices. To compute the marginal gain of a vertex v, we maintain the portion of the boundary currently guarded by the selected set S, compute the portion guarded by v, and take their union to determine the new coverage. Since computing the visibility region of a single vertex can be done in $O(n)$ time [8], and union operations can be managed efficiently with appropriate data structures, each marginal gain evaluation takes $O(n)$. Thus, the total running time of the greedy algorithm is $O(kn^2)$ in the worst case.

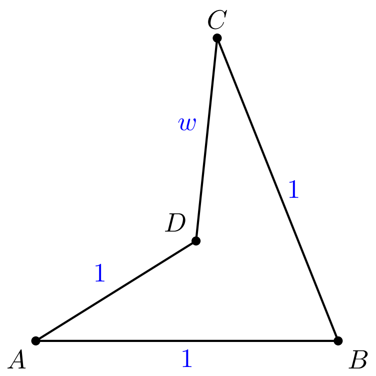
By leveraging the monotonicity and submodularity of the objective functions in Maximum Length Vertex Guard and Maximum Value Vertex Guard, we obtain simple greedy algorithms that achieve a $(1 - 1/e)$-approximation. While previous work [7, 4, 6] also achieves this approximation ratio, we additionally prove that this is the best possible approximation ratio unless P=NP. Moreover, our approach offers a slight improvement in efficiency: our algorithms run in $O(kn^2)$ time, whereas their algorithms have runtimes of $O(n^4)$ and $O(k^2n^2)$.

3 Intuition-Building Examples for Budgeted Maximum Value Vertex Guard

In this section, we show that three natural greedy strategies for Budgeted Maximum Value Vertex Guard may all perform arbitrarily badly compared to an optimal solution. In other words, these strategies will not automatically form approximations for Budgeted Maximum Value Vertex Guard, as there exists a problem instance where the total weight they achieve is *not* boundedly close to the optimal. Note that this is in contrast to Maximum Length Vertex Guard and Maximum Value Vertex Guard, where the natural greedy strategy formed a constant-factor approximation, validating the added difficulty of Budgeted Maximum Value Vertex Guard.

A first attempt could be to greedily add the vertex which maximizes the marginal gain in total weight guarded. To see why this fails, fix a $w \in \mathbb{N}$ such that $w \geq 3$. Then construct the polygon depicted in Figure 2, with w many "cones". The right edge of each cone has weight w, while all other edges have weight 1. The apex of each cone has cost w, while all other vertices have cost w^2. The budget B is equal to w^2, and now we have an instance of Budgeted Maximum Value Vertex Guard, with Figure 2 and $B = w^2$.

The porposed greedy strategy would start by choosing the vertex that maximizes the total weight guarded. Initially, this is one of the "valleys" in the polygon, as these vertices guard $2w + 2$ total weight, whereas all other vertices guard at most $w + 1$. Any one of these vertices has cost w^2, so adding it to our solution uses our entire budget. Thus the algorithm which greedily maximizes the marginal gain in total weight guarded ends up guarding $2w + 2$ total weight on Figure 2.



Vertex	Cost
A	2
B	$w+4$
C	$w+4$
D	$w+4$

Figure 3: Weights of each edge are represented in blue next to the edge, and vertex costs are denoted in the table on the right. The given budget is $B = w + 4$.

However, an optimal strategy would be to place a guard at each apex. There are w many apices and each one has cost w, so to use all of them would cost w^2 (which is exactly our budget). This guard placement also guards our entire polygon boundary, meaning the total weight guarded by the optimal set of guards is $w(w) + w(1) + 1 = w^2 + w + 1$. To compare the total weight achieved between these two algorithms, the weight achieved by the greedy divided by the weight achieved by the optimal is equal to $\frac{2w+2}{w^2+w+1}$. We can make this value arbitrarily small as we increase w, meaning the greedy algorithm which maximizes total weight guarded performs arbitrarily badly compared to an optimal solution and will not achieve any constant factor approximation ratio.

Now we consider two alternative greedy strategies and show that they also fail, using Figure 3. The first would be to greedily add the vertex with cheapest cost. The second would be to greedily add the vertex which maximizes the ratio of marginal gain in total weight to vertex cost. Starting with the first strategy, consider Figure 3 where weights and costs are displayed, and the budget is $w + 4$. The cheapest vertex is A, which has cost of 2. The vertex A guards total weight of 2, and once we have placed a guard there, we cannot afford to place guards at any of the remaining vertices. Thus this greedy strategy achieves total weight of 2 on this example. An optimal strategy would be to place a guard on either D or B, this stays within budget and guards the entire polygon, achieving total weight of $w + 3$. Comparing the weight achieved by each strategy, we have $\frac{2}{w+3}$, which we can again make arbitrarily small as we increase w.

Similarly, if we wanted to greedily choose the vertex which maximizes the ratio of marginal gain in total weight to vertex cost (i.e., provides the best "bang-for-your-buck"), we would choose vertex A. It guards weight of 2 and costs 2, a ratio of 1. Any other vertex can guard at most $w + 3$, while costing $w + 4$, and $\frac{w+3}{w+4} < 1$ for all w. Again, choosing A prevents adding any more guards, achieving a total guarded weight of 2, while an optimal strategy could guard $w + 3$, and $\frac{2}{w+3}$ can be arbitrarily small for large w.

[10] studies a related problem where the goal is to cover as much total weight as possible using sets whose combined cost does not exceed a given budget. They show that the greedy strategy of repeatedly selecting the set that maximizes the ratio of marginal gain in weight to cost does not guarantee any approximation ratio on its own. However, they prove that if you consider both the collection of sets selected by this greedy method and the single set that covers the most weight on its own, and return whichever of the two covers more weight, then the result is a $\frac{1}{2}(1 - 1/e)$-approximation. They further refine this strategy to

obtain a full $(1 - 1/e)$-approximation.

This approach offers a direction for tackling Budgeted Maximum Value Vertex Guard. Specifically, we could try applying the same strategy: run the greedy algorithm that selects vertices by maximizing the ratio of marginal gain in boundary weight to cost, and also identify the single vertex that guards the most weight on its own. Then, return whichever of the two guard sets achieves greater coverage. A natural question is whether this method yields a provable approximation guarantee for Budgeted Maximum Value Vertex Guard, perhaps even matching the $\frac{1}{2}(1 - 1/e)$ bound shown in the set cover context.

4 Conclusion

This paper explored boundary-focused variants of the Art Gallery problem, where the goal is to maximize the total length or weight of the polygon boundary that is fully guarded by a limited number of vertex guards. We studied three key variants: Maximum Length Vertex Guard, Maximum Value Vertex Guard, and a novel budgeted version, Budgeted Maximum Value Vertex Guard. For Maximum Length Vertex Guard and Maximum Value Vertex Guard, we proved that the objective functions are monotone and submodular, allowing the standard greedy algorithm to achieve a $(1 - 1/e)$-approximation. We also improved the runtime over prior work [4] and established that no better approximation is possible unless P=NP. For Budgeted Maximum Value Vertex Guard, we demonstrated that several natural greedy strategies fail to yield approximations, and we proposed a slightly more nuanced algorithm, which we conjecture achieves a $\frac{1}{2}(1 - 1/e)$-approximation.

4.1 Future Work

The analysis of Maximum Length Vertex Guard and Maximum Value Vertex Guard is essentially complete, thanks to their monotone and submodular structure. However, Budgeted Maximum Value Vertex Guard still need further theoretical development — proving that our proposed algorithm achieves the conjectured approximation ratio, and narrowing the gap between this bound and the best possible. Beyond these static settings, our results could also be expanded to dynamic and online settings (similar to the current focus of most research into covering problems). For example, if guards represent fixed cameras in a gallery, the polygon's structure or edge weights may change over time, reflecting layout changes or shifting priorities within an art gallery. In such settings, reconfiguring the camera placement can be expensive. Designing algorithms that can maintain near-optimal coverage with minimal guard movement at each time step would be both practically useful and theoretically interesting, and would reinforce the connection between our problem to ongoing work on dynamic and online Set Cover.

References

[1] A. Abdelkader, A. Saeed, K. Harras, and A. Mohamed. The inapproximabiliy of illuminating polygons by α-floodlights. *Proceedings of the Canadian Conference on Computational Geometry 2015*, August 2015.

[2] G. Cornuejols, M. L. Fisher, and G. L. Nemhauser. Exceptional paper—location of bank accounts to optimize float: An analytic study of exact and approximate algorithms. *Management Science*, 23:789–810, April 1977.

[3] R. Das, O. Filtser, M. J. Katz, and J. S. Mitchell. Robustly guarding polygons. volume 293, pages 1–47, June 2024.

[4] I. Emiris, C. Fragoudakis, and E. Markou. Maximizing the guarded interior of an art gallery. *European Workshop on Computational Geometry*, pages 165–169, March 2006.

[5] U. Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45: 634–652, July 1998.

[6] C. Fragoudakis, E. Markou, and S. Zachos. How to place efficiently guards and paintings in an art gallery. *Lecture Notes in Computer Science*, 3746:145–154, November 2005. doi: 10.1007/11573036_14.

[7] C. Fragoudakis, E. Markou, and S. Zachos. Maximizing the guarded boundary of an art gallery is apx-complete. *Computational Geometry*, 38:170–180, October 2007.

[8] H. E. Gindy and D. Avis. A linear algorithm for computing the visibility polygon from a point. *Journal of Algorithms*, 2:186–197, June 1981.

[9] D. S. Hochbaum and A. Pathria. Analysis of the greedy approach in problems of maximum k-coverage. *Naval Research Logistics*, 45:615–627, September 1998.

[10] S. Khuller, A. Moss, and J. S. Naor. The budgeted maximum coverage problem. *Information Processing Letters*, 70:39–45, April 1999.

[11] J. O'Rourke. *Art Gallery Theorems and Algorithms.* Oxford University Press, Inc., USA, 1987. ISBN 0195039653.