

The Generalized Heron Problem: A Computational Perspective Using Line Search and Conic Reformulation

Manohar Choudhary

Department of Mathematics and Statistics,
Dr. Hari Singh Gour Vishwavidyalaya, Sagar, India
Email: manoharfbg@gmail.com

Triloki Nath*

Department of Mathematics and Statistics,
Deen Dayal Upadhyaya Gorakhpur University, Gorakhpur, India
*Corresponding Author E-mail: tverma07@gmail.com

Ram K. Pandey

Department of Mathematics and Statistics,
Dr. Hari Singh Gour Vishwavidyalaya, Sagar, India
Email: pandeywavelet@gmail.com

(Received January 19, 2026, Accepted March 06, 2026)

Abstract: The classical Heron problem and its numerous generalizations have long stood as a fundamental *geometric optimization problem* in convex analysis and optimization. In this paper, we investigate the Generalized Heron Problem (GHP) and propose two new computational strategies: (i) a projected subgradient method with *exact line search* (PSM-ELS), and (ii) a second-order cone programming (SOCP) reformulation. The PSM-ELS enhances the classical subgradient framework by replacing diminishing step sizes with adaptive line search along feasible rays, achieving significantly faster convergence and greater effectiveness in nonsmooth settings. The SOCP reformulation uses modern interior-point solvers to provide exact benchmark solutions whenever the feasible region and target sets admit conic representations. Extensive numerical experiments across diverse examples in $\mathbb{R}^2$, $\mathbb{R}^3$, and $\mathbb{R}^{25}$ demonstrate that PSM-ELS consistently attains minimum root-mean-square error (RMSE) in the fewest iterations, with rapid monotone error decay and competitive runtimes. Compared with the majorization–minimization (MM) algorithm, SM-ELS offers broader applicability; compared with the basic subgradient method, it delivers notable convergence improvement, and compared with SOCP solvers, it provides a lightweight and scalable alternative. These findings establish PSM-ELS as an efficient, stable, and versatile approach for nonsmooth

convex optimization problems of distance-sum type, with potential extensions to large-scale and real-world applications.

Keywords: Heron Problem, Geometric Optimization, Distance Minimization, Convexity.

1. Introduction

The classical *Heron problem*, attributed to Heron of Alexandria (10–75 AD), asks for the point on a given line in the Euclidean plane that minimizes the sum of its distances to two fixed points. This geometric problem, which can be solved by elementary geometry or calculus, has inspired numerous extensions. Fermat posed a related problem for three points in the plane, leading to the celebrated Fermat–Torricelli problem solved by Torricelli in the seventeenth century. Simpson, Steiner, and others later proposed weighted and higher-dimensional generalizations. In the twentieth century, the problem evolved into the Weber location problem, and Weiszfeld introduced an influential iterative algorithm for its solution¹. These connections reveal a deep interplay between geometry, analysis, and optimization.

In the framework of modern convex analysis, the Heron problem has been substantially generalized. Let $C_1, \dots, C_m \subset \mathbb{R}^n$ be nonempty closed convex sets, and let $S \subset \mathbb{R}^n$ be a nonempty closed convex feasible region. The *generalized Heron problem* is formulated as

$$(1) \quad \min_{x \in S} F(x) := \sum_{i=1}^m d_{C_i}(x),$$

where $d_{C_i}(x) := \inf_{y \in C_i} \|x - y\|$ denotes the distance from x to C_i , and $\|\cdot\|$ is the Euclidean norm.

Problem (1) is a constrained convex optimization problem, typically nonsmooth because of the distance functions. It includes many well-known models: if $m = 2$, each C_i is a singleton and S is a line in $\mathbb{R}^2$, then (1) reduces to the classical Heron problem. If m consists of finitely many singletons in $\mathbb{R}^2$ and $S = \mathbb{R}^2$, one recovers the Fermat–Weber problem. In particular, the sets C_i may represent geometric regions such as balls, polytopes, or subspaces, while the feasible set S represents operational or geometric constraints. Applications arise in facility location, signal processing, imaging, and data science, where the objective is to identify a representative point that minimizes distances to multiple convex models subject to feasibility requirements.

Related Work

The generalized Heron problem was first rigorously investigated by Mordukhovich, Nam, and Salinas^{2,3}, who applied convex and variational analysis to establish existence results, derive necessary and sufficient

optimality conditions, and design a convergent subgradient algorithm. Their work extended the classical Fermat–Weber setting to arbitrary families of convex sets and provided geometric solution characterizations via normal cones and projections⁴. Subsequently, Chi and Lange¹ revisited the problem from a computational viewpoint and introduced a majorization–minimization (MM) algorithm. Their approach generalized Weiszfeld’s iteration and demonstrated a significant acceleration compared with the standard subgradient method for the Euclidean case. However, the MM algorithm depends on building a surrogate function that always stays above the original objective. This is relatively easy in certain cases, such as when the objective is quadratic or when it can be split into independent parts (a separable structure). In more general problems, though, finding suitable surrogates becomes technically difficult and often requires problem-specific tricks. Together, these contributions positioned the generalized Heron problem as both a mathematically rich and computationally challenging benchmark problem for nonsmooth convex optimization.

Our Contributions

This paper develops two new computational strategies for solving (1) in $\mathbb{R}^n$ with m convex sets:

1. A *projected subgradient method with exact line search along feasible rays*, which leverages one-dimensional convexity of the distance-sum function to obtain improved step-size selection compared to classical diminishing-step subgradient schemes.
2. A *second-order cone programming (SOCP) reformulation* of (1), valid whenever the sets C_i and S admit second-order cone representations, thereby enabling the use of modern interior-point solvers.

We present theoretical justifications for both methods, establish descent properties and optimality characterizations, and conduct extensive numerical experiments. In particular, we compare the proposed approaches with the subgradient algorithm of the paper² and the MM algorithm of the paper¹, thus providing the first comprehensive benchmark across four distinct algorithmic paradigms for the generalized Heron problem. Our results demonstrate the comparison of efficiency, accuracy, and convergence behavior of each method and suggest promising directions for large-scale nonsmooth optimization in higher dimensions.

The paper is organized as follows. Section 2 reviews necessary concepts from convex analysis and convex optimization. Section 3 summarizes existing computational methods, Section 4 details our new approaches, Section 5 presents numerical comparisons, and Section 7 concludes with final remarks and future directions.

2. Preliminary

This section provides necessary mathematical preliminaries and optimization tools essential for understanding the computational methodologies developed in subsequent sections. Our goal is to ensure that the exposition is self-contained and accessible to a wide-class of readers familiar with optimization. We refer the publications⁵⁻⁹ to for a comprehensive study of these materials and additional backgrounds.

The theoretical foundation of the Generalized Heron Problem (GHP) lies in convex analysis and conic optimization. In particular, the fundamental properties of convex sets and convex functions—such as the fact that every local minimum of a convex function is global, or that the distance function to a convex set is convex—play a crucial role. Additionally, second-order cone programming (SOCP) provides a powerful framework to reformulate the GHP when the problem admits conic representations.

Accordingly, we divide this section into two parts. The first subsection reviews key notions and results from convex analysis and optimization, while the second introduces basic preliminaries of second-order cone programming.

2.1 Convex Analysis: Key Concepts

We recall the key definitions and results from convex analysis that form the basis of the GHP. The focus is on convex sets, convex functions, epigraphs, distance functions, and projections, together with essential tools from nonsmooth analysis such as subgradients and subdifferentials.

These concepts not only establish existence and optimality guarantees but also provide the foundation for algorithmic strategies, including projected subgradient methods.

Definition 2.1.1 (Convex set). A set $C \subset \mathbb{R}^n$ is *convex* if

$$\lambda x + (1 - \lambda)y \in C \quad \text{for all } x, y \in C \text{ and } \lambda \in [0, 1].$$

In other words, every line segment joining two points in C lies entirely within C .

Definition 2.1.2 (Effective domain). For a function $f: \mathbb{R}^n \rightarrow (-\infty, +\infty]$, the *effective domain* is

$$\text{dom} f := \{x \in \mathbb{R}^n \mid f(x) < +\infty\}.$$

Definition 2.1.3 (Convex function). A function $f: \mathbb{R}^n \rightarrow (-\infty, +\infty]$ is *convex* if $\text{dom} f$ is convex and

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y)$$

for all $x, y \in \text{dom} f$ and $\lambda \in [0, 1]$. It is *strictly convex* if the inequality is strict whenever $x \neq y$ and $\lambda \in (0, 1)$.

Definition 2.1.4 (Epigraph). The *epigraph* of $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is

$$\text{epi } f := \{(x, t) \in \mathbb{R}^n \times \mathbb{R} \mid f(x) \leq t\}.$$

It is easy to verify that a function f is convex if and only if $\text{epi } f$ is a convex set.

Definition 2.1.5 (Distance function). For a nonempty set $C \subset \mathbb{R}^n$, the *distance function* is

$$d_C(x) := \inf_{y \in C} \|x - y\|.$$

If C is convex, then d_C is convex and Lipschitz continuous.

Definition 2.1.6 (Euclidean projection). If $C \subset \mathbb{R}^n$ is nonempty, then *projection* of $x \in \mathbb{R}^n$ onto C is

$$\text{proj}_C(x) := \underset{y \in C}{\text{argmin}} \|x - y\|.$$

For a closed convex set C , the projection is unique, and Moreover, $d_C(x) = \|x - \text{proj}_C(x)\|$.

Definition 2.1.7 (Indicator function). For a set $C \subset \mathbb{R}^n$, the *indicator function* is

$$\delta_C(x) := \begin{cases} 0, & x \in C, \\ +\infty, & x \notin C. \end{cases}$$

Definition 2.1.8 (Normal cone). Let $C \subset \mathbb{R}^n$ be convex, and let $x \in C$. The *normal cone* to C at x is

$$N_C(x) := \{v \in \mathbb{R}^n \mid \langle v, y - x \rangle \leq 0 \quad \forall y \in C\},$$

where $\langle \cdot, \cdot \rangle$ denotes the standard Euclidean inner product. Geometrically, $N_C(x)$ consists of all vectors that form an obtuse angle with every feasible direction from x within C .

Since the distance function is not differentiable when $x \in C$, we require tools from non-smooth convex analysis. The central object is the subgradient, which generalizes the gradient.

Definition 2.1.9 (Subgradient and subdifferential). Let $f: \mathbb{R}^n \rightarrow (-\infty, +\infty]$ be convex. A vector $v \in \mathbb{R}^n$ is a *subgradient* of f at $x \in \text{dom } f$ if

$$f(y) \geq f(x) + \langle v, y - x \rangle \quad \forall y \in \mathbb{R}^n.$$

The set of all subgradients is called the *subdifferential* of f at x , denoted $\partial f(x)$.

For the indicator function δ_C of a convex set C , the subdifferential at any $x \in C$ coincides with the normal cone to C at x ¹⁰:

$$(2) \quad \partial \delta_C(x) = N_C(x), \quad x \in C.$$

Investigating the GHP, the subdifferential of the distance function plays a crucial role. For algorithmic purposes, the computation of subdifferentials is essential.

Proposition 2.1.10 (Subdifferential of the distance function). *Let $C \subset \mathbb{R}^n$ be nonempty, closed, and convex. Then for any $x \in \mathbb{R}^n$,*

$$\partial d_C(x) = \begin{cases} \left\{ \frac{x - \text{proj}_C(x)}{\|x - \text{proj}_C(x)\|} \right\}, & \text{if } x \notin C, \\ N_C(x) \cap \mathbb{B}, & \text{if } x \in C, \end{cases}$$

where $\mathbb{B} := \{z \in \mathbb{R}^n \mid \|z\| \leq 1\}$ is the closed unit ball. Proof can be found in the paper⁸.

In the special case where $C = \{y\}$ is a singleton, we have $\text{proj}_C(x) = \{y\}$ and $d_C(x) = \|x - y\|$. Then, for $x \neq y$,

$$(3) \quad \partial d_C(x) = \left\{ \frac{x-y}{\|x-y\|} \right\}.$$

Theorem 2.1.11 (Moreau–Rockafellar Theorem¹⁰). *Let $\psi_i: \mathbb{R}^n \rightarrow (-\infty, +\infty]$, $i = 1, \dots, k$, be closed convex functions. Suppose there exists $\bar{x} \in \bigcap_{i=1}^k \text{dom} \psi_i$ at which all but at most one of the functions are continuous. Then for all $x \in \bigcap_{i=1}^k \text{dom} \psi_i$, $\partial(\sum_{i=1}^k \psi_i)(x) = \sum_{i=1}^k \partial \psi_i(x) := \{\sum_{i=1}^k w_i \mid w_i \in \partial \psi_i(x), i = 1, \dots, k\}$.*

2.2 Second-Order Cone Programming: Key Concepts

This subsection provides a comprehensive background on Second-Order Cone Programming (SOCP) and the concept of Second-Order Cone Representability (SOC-representability), which are fundamental to reformulating and solving the Generalized Heron Problem.

Definition 2.2.12 (Lorentz Cone / Second-Order Cone). *For $n \geq 1$, the Lorentz cone (also called the second-order cone) in $\mathbb{R}^{n+1}$ is defined by*

$$\mathcal{L}^{n+1} := \{(x, t) \in \mathbb{R}^n \times \mathbb{R} \mid \|x\| \leq t\}.$$

Equivalently,

$$\mathcal{L}^{n+1} = \{(x_1, \dots, x_n, t) \in \mathbb{R}^{n+1} : \sqrt{x_1^2 + \dots + x_n^2} \leq t\}.$$

The Lorentz cone is a closed, convex, self-dual cone¹¹. The second-order cone or Lorentz cone is often referred to by several alternative names. It is called the *quadratic cone* because it is defined through a quadratic inequality, and it is also known as the *ice-cream cone*.

Conic Optimization and SOCP

Conic optimization is a broad class of convex optimization problems in which a linear function is minimized subject to membership in a convex cone. A general conic problem can be expressed as

$$\begin{aligned} & \text{minimize} && c^\top z \\ & \text{subject to} && Az + b \in \mathcal{K}, \end{aligned}$$

where $\mathcal{K}$ is a closed convex cone (such as the nonnegative orthant, the second-order cone, or the semidefinite cone).

A particularly important subclass is the *second-order cone program* (SOCP)¹¹, which takes the form

$$(4) \quad \begin{aligned} & \min_{z \in \mathbb{R}^d} f^\top z \\ & \text{s.t.} \quad \|A_i z + b_i\| \leq c_i^\top z + d_i, \quad i = 1, \dots, q, \\ & \quad \quad Fz = g, \end{aligned}$$

where each inequality constraint represents membership in a second-order (Lorentz) cone. Optimization problems involving linear functions over the intersection of affine constraints and products of Lorentz cones are called *second-order cone programs* (SOCPs). SOCP is a robust class of convex problems that strictly generalizes Linear Programming (LP), convex Quadratic Programming (QP), and Quadratically-Constrained Quadratic Programming (QCQP)¹². The SOCPs are efficiently solvable by modern interior-point methods and are widely supported in optimization modeling software.

Definition 2.2.13 (SOC-Representable Set¹²⁻¹³). A convex set $C \subset \mathbb{R}^n$ is SOC-representable if it can be defined as the projection of the solution set of a finite system of second-order cone constraints onto the x -space, potentially utilizing auxiliary variables $u \in \mathbb{R}^k$. The formal representation (SOCR) is:

$x \in C \Leftrightarrow$ there exist u such that

$$(5) \quad \left\| A_j \begin{pmatrix} x \\ u \end{pmatrix} + b_j \right\| \leq c_j^\top \begin{pmatrix} x \\ u \end{pmatrix} + d_j, \quad j = 1, \dots, N.$$

Definition 2.2.14 (SOC-Representable Function¹²). A convex function $f: \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ is SOC-representable if its epigraph, $\text{epi } f := \{(x, t) \in \mathbb{R}^n \times \mathbb{R} \mid f(x) \leq t\}$, is an SOC-representable set.

If a function is SOC-representable, then its level sets are also SOC-representable, as they inherit their representation from the function's epigraph¹². Moreover, SOC-representability is preserved under operations such as intersection, direct product, inverse affine image, and nonnegative weighted sum¹³.

Theorem 2.2.15 (SOC-representability conditions¹²⁻¹³).

1. A set $C \subset \mathbb{R}^n$ is SOC-representable if and only if it is convex and can be expressed as the projection of the solution set of finitely many second-order cone constraints.
2. A function $f: \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ is SOC-representable if and only if it is convex and its epigraph is SOC-representable.

Remark 2.2.16. The practical consequence is that if f and C are SOC-representable, then the convex optimization problem

$$\begin{aligned} & \min f(x) \\ & \text{s.t.} \quad x \in C \end{aligned}$$

can be cast as an SOCP and efficiently solved via interior-point methods¹².

Proposition 2.2.17. *A nonempty set X is SOC-representable if and only if its indicator function $\delta_X(x)$ is SOC-representable¹³.*

2.2.1 Examples of SOC-Representable Sets and Functions

A key strength of second-order cone programming (SOCP) is its capacity to model a wide variety of convex structures. We highlight below several representative sets and functions that can be expressed in second-order cone form.

1. **Affine and Constant Functions:** Any affine function $g(x) = a^\top x + b$ and constant functions are SOC-representable.
2. **Norms and Quadratic Forms:**
 - The Euclidean norm $g(x) = \|x\|$.
 - The squared norm $g(x) = \|x\|^2$.
 - Convex quadratic forms $g(x) = x^\top Qx + q^\top x + r$ with $Q \succcurlyeq 0$.
3. **Convex Sets:**
 - Polyhedral sets defined by finitely many linear inequalities.
 - The hypograph of the geometric mean constraint $t \leq \sqrt{x_1 x_2}$, for $x_1, x_2 \geq 0$.

A more comprehensive list of SOC-representable structures and their derivations is provided in²

2.2.2 Non-SOC-representable Cases

Despite the broad coverage, not all convex functions are SOC-representable. Notably:

- **Nonconvex functions** are excluded, since epigraph convexity is necessary.
- Certain **transcendental convex functions**, such as e^x and $-\ln(x)$, are not exactly SOC-representable due to the algebraic structure of SOC constraints.

Definition 2.2.18 (Root Mean Square Error (RMSE)). Let $\{x^{(k)}\}_{k=1}^N \subset \mathbb{R}^n$ be a sequence of approximate solutions generated by an algorithm, and let $x^* \in \mathbb{R}^n$ denote a reference solution (analytical or SOCP-based). The *root mean square error (RMSE)* is defined as

$$\text{RMSE} := \left(\frac{1}{N} \sum_{k=1}^N \|x^{(k)} - x^*\|^2 \right)^{1/2}.$$

Remark 2.2.19. RMSE provides a quantitative measure of the average deviation of the computed iterates from the reference solution and serves as a standard metric to evaluate convergence behavior.

In our experiments, we plot $\log_{10}(\text{RMSE})$ against the number of iterations, which enhances readability and captures convergence over multiple orders of magnitude. This allows us to compare the accuracy and efficiency of the Subgradient, MM, and PSM-ELS methods on a common scale and to track how rapidly their iterates approach the benchmark solution.

Notation: The optimal value of a program will be denoted by $Val(\cdot)$.

3. Existing Computational Approaches

Two principal computational strategies for solving the Generalized Heron Problem (1) have been proposed in the literature: (i) a projected subgradient method grounded in convex analysis, introduced by Mordukhovich, Nam, and Salinas², and (ii) a majorization–minimization (MM) algorithm developed by Chi and Lange¹. We summarize both methods below, emphasizing their iterative schemes and key theoretical properties.

3.1 Projected Subgradient Method

Mordukhovich, Nam, and Salinas² proposed an algorithm to solve (1) by means of the classical projected subgradient method. The scheme applies directly to the nonsmooth convex objective (1) and requires only the ability to compute Euclidean projections onto the sets C_i and S .

At iteration k , one computes

$$y_i^k := \text{proj}_{C_i}(x_k), \quad i = 1, \dots, m,$$

and forms a subgradient of F at x_k as

$$g_k := \sum_{i=1}^m u_{ik}, \quad u_{ik} := \begin{cases} \frac{x_k - y_i^k}{\|x_k - y_i^k\|}, & x_k \notin C_i, \\ N_{C_i}(x_k) \cap \mathbb{B}, & x_k \in C_i, \end{cases}$$

where $N_{C_i}(x_k)$ denotes the normal cone and $\mathbb{B}$ is the closed unit ball. In practice, if $x_k \in C_i$, one may take $u_{ik} = 0$.

The update is then given by

$$x_{k+1} := \text{proj}_S(x_k - \alpha_k g_k),$$

where the step size $\alpha_k > 0$ is chosen according to a diminishing rule, e.g.,

$$\sum_{k=1}^{\infty} \alpha_k = \infty, \quad \sum_{k=1}^{\infty} \alpha_k^2 < \infty.$$

Under these standard assumptions, iterative sequence of $\{x_k\}$ converges to an optimal solution of (1). The method is simple and robust, requiring only basic projection computations. Its main drawback is slow convergence, stemming from the nonsmoothness of F and the reliance on predetermined diminishing step sizes.

3.2 Majorization–Minimization Algorithm

Chi and Lange¹ reformulated (1) within the MM framework, which iteratively minimizes a surrogate function that majorizes F and is easier to optimize. Given the current iterate $x_m \in S$, the method constructs the quadratic surrogate

$$g(x \mid x_m) = 1/2 \sum_{i=1}^m w_i^{(m)} \|x - \text{proj}_{C_i}(x_m)\|^2,$$

where the weights

$$w_i^{(m)} = \frac{1}{\|x_m - \text{proj}_{C_i}(x_m)\|}$$

are inversely proportional to the current distances from x_m to the sets C_i .

Minimizing this surrogate g over S yields the update

$$(6) \quad x_{m+1} = \text{proj}_S \left(\sum_{i=1}^m \alpha_i^{(m)} \text{proj}_{C_i}(x_m) \right),$$

where the normalized weights are

$$\alpha_i^{(m)} = \frac{w_i^{(m)}}{\sum_{j=1}^m w_j^{(m)}}.$$

The MM algorithm ensures monotone decrease of F and, in Euclidean settings, typically converges much faster than the basic subgradient method. However, it is specialized to the Euclidean norm and depends on explicit formulas for the projections proj_{C_i} , which may limit its applicability in more general metric or constraint settings.

These two algorithms will serve as the baseline methods against which we compare our proposed strategies in Section 4.

4. Proposed Methods

In this section, we introduce two novel computational approaches for solving the Generalized Heron Problem (1). The first method enhances the projected subgradient algorithm of¹³ by incorporating an *exact line search* strategy, enabling adaptive step sizes and significantly faster convergence compared to the standard diminishing step rule. The second method reformulates (1) as a *Second–Order Cone Program* (SOCP), thereby enabling the use of high-performance conic solvers to obtain solutions to high accuracy.

4.1 Projected Subgradient Method with Exact Line Search

The projected subgradient method of² employs the classical diminishing step-size rule $\alpha_k = 1/k$, which guarantees convergence under mild

assumptions. However, this choice often leads to excessively small steps in later iterations, causing very slow progress. To overcome this drawback, we propose to select α_k adaptively by means of an *exact line search* along the feasible ray. This allows the algorithm to take larger steps when advantageous and smaller steps only when necessary, thereby accelerating convergence in practice.

Theorem 4.1.20 (Exact Line Search for the GHP). *Let $x_k \in S$ be the current iterate in the projected subgradient method for (1), let $g_k \in \partial F(x_k)$ be a subgradient, and set $p_k = -g_k$. Define $\varphi(\alpha) := F(x_k + \alpha p_k) = \sum_{i=1}^m d_{C_i}(x_k + \alpha p_k)$, $\alpha \geq 0$. Then φ admits a minimizer α_k , and if φ is differentiable at α_k in the interior of the feasible ray, it satisfies the first-order condition $\varphi'(\alpha_k) = \sum_{i=1}^m \left\langle \frac{x_k + \alpha_k p_k - \text{proj}_{C_i}(x_k + \alpha_k p_k)}{\|x_k + \alpha_k p_k - \text{proj}_{C_i}(x_k + \alpha_k p_k)\|}, p_k \right\rangle = 0$*

Proof. Each distance function d_{C_i} is convex and 1-Lipschitz. The function $\varphi(\alpha)$ is a sum of compositions of these convex functions with an affine map ($\alpha \mapsto x_k + \alpha p_k$), and is therefore convex in α . Since the feasible set for α is a closed interval, the convex function φ must attain its minimum.

Now we compute the derivative. A standard result in convex analysis states that the distance function $d_{C_i}(z)$ is differentiable everywhere except for points $z \in C_i$. For any $z \notin C_i$, its gradient is given by the unit vector pointing from its projection:

$$\nabla d_{C_i}(z) = \frac{z - \text{proj}_{C_i}(z)}{\|z - \text{proj}_{C_i}(z)\|}.$$

Let $z(\alpha) = x_k + \alpha p_k$. The derivative of the i -th term in $\varphi(\alpha)$ can be found using the chain rule:

$$\frac{d}{d\alpha} d_{C_i}(z(\alpha)) = \langle \nabla d_{C_i}(z(\alpha)), z'(\alpha) \rangle.$$

Since $z'(\alpha) = p_k$, we can substitute the gradient formula to get:

$$\frac{d}{d\alpha} d_{C_i}(x_k + \alpha p_k) = \left\langle \frac{x_k + \alpha p_k - \text{proj}_{C_i}(x_k + \alpha p_k)}{\|x_k + \alpha p_k - \text{proj}_{C_i}(x_k + \alpha p_k)\|}, p_k \right\rangle.$$

Summing over all $i = 1, \dots, m$ gives the expression for the derivative of $\varphi(\alpha)$:

$$\varphi'(\alpha) = \sum_{i=1}^m \left\langle \frac{x_k + \alpha p_k - \text{proj}_{C_i}(x_k + \alpha p_k)}{\|x_k + \alpha p_k - \text{proj}_{C_i}(x_k + \alpha p_k)\|}, p_k \right\rangle.$$

By Fermat's rule, a minimizer α_k in the interior of the feasible domain where φ is differentiable must satisfy the stationarity condition $\varphi'(\alpha_k) = 0$. This gives the root-finding equation for the optimal step size.

Algorithm 1: Projected Subgradient Method with Exact Line Search (PSM-ELS)**Input:** $C_1, \dots, C_m \subset \mathbb{R}^n$, feasible set S , $x^0 \in S$, $\varepsilon > 0$, $K_{\max}$ **Output:** Approximate solution x^* **for** $k = 0, 1, 2, \dots, K_{\max}$ **do**

$$y_i^k \leftarrow \text{proj}_{C_i}(x^k), \quad i = 1, \dots, m$$

$$u_i^k \leftarrow \begin{cases} \frac{x^k - y_i^k}{\|x^k - y_i^k\|}, & x^k \notin C_i \\ 0, & x^k \in C_i \end{cases}$$

$$g^k \leftarrow \sum_{i=1}^m u_i^k, \quad p^k \leftarrow -g^k$$

if $\|p^k\| = 0$ **then return** x^k **end if**

$$\alpha_k \leftarrow \arg\min_{\alpha \geq 0} \sum_{i=1}^m d_{C_i}(x^k + \alpha p^k)$$

$$x^{k+1} \leftarrow \text{proj}_S(x^k + \alpha_k p^k)$$

if $\|F(x^{k+1}) - F(x^k)\| < \varepsilon$ **then return** x^{k+1} **end if****end for****return** $x^{k_{\max}+1}$ **4.2 SOCP Formulation**

Before presenting the SOCP reformulation of the generalized Heron problem, we briefly motivate this approach. As discussed in the preliminaries, second-order cone programming (SOCP) offers a powerful framework for handling convex optimization models whenever the underlying sets are SOC-representable. This subsection establishes that the generalized Heron problem (GHP) can be exactly reformulated as an SOCP, thereby enabling the use of efficient interior-point solvers to obtain globally optimal solutions with high accuracy¹³.

Theorem 4.2.21 (SOCP reformulation of the GHP). *Let $S, C_1, \dots, C_m \subset \mathbb{R}^n$ be nonempty closed convex sets that admit second-order cone representations. Then the generalized Heron problem*

$$(7) \quad \min_{x \in S} F(x) := \sum_{i=1}^m d_{C_i}(x)$$

is equivalent to the SOCP

$$(8) \quad \begin{aligned} \min_{x, y_i, t_i} \quad & \sum_{i=1}^m t_i \\ \text{s.t.} \quad & x \in S, \quad y_i \in C_i, \\ & \|x - y_i\| \leq t_i, \quad t_i \geq 0, \quad i = 1, \dots, m. \end{aligned}$$

In particular:

1. The optimal values coincide:

$$\text{val}(7) = \text{val}(8)$$

2. If (x^*, y_i^*, t_i^*) solves (8), then x^* solves (7).

3. If x^* solves (7), then setting $y_i^* := \text{proj}_{C_i}(x^*)$, $t_i^* := \|x^* - y_i^*\|$, yields an optimal solution (x^*, y_i^*, t_i^*) of (8).

Proof. (SOCP $\Rightarrow$ GHP) If (x, y_i, t_i) is feasible for (8), then $y_i \in C_i$ and $t_i \geq \|x - y_i\| \geq d_{C_i}(x)$ for all i . Summing gives

$$F(x) = \sum_{i=1}^m d_{C_i}(x) \leq \sum_{i=1}^m t_i,$$

so $\text{val}(1) \leq \text{val}(8)$.

(GHP $\Rightarrow$ SOCP) If x^* is optimal for (7), define $y_i^* = \text{proj}_{C_i}(x^*)$ and $t_i^* = \|x^* - y_i^*\| = d_{C_i}(x^*)$. Then (x^*, y_i^*, t_i^*) is feasible for (8) and attains objective $\sum_i d_{C_i}(x^*) = F(x^*)$. Thus $\text{val}(8) \leq \text{val}(1)$.

Combining the two directions yields equality of values and the optimal solution correspondence.

4.3 Practical Solution via Conic Solvers

Once the generalized Heron problem is expressed in the SOCP form (8), it can be solved directly using standard modeling frameworks (e.g., **CVX**, **CVXPY**, or **YALMIP**) and high-performance conic solvers such as **MOSEK**, **ECOS**, or **SCS**. These solvers exploit interior-point methods to compute globally optimal solutions with polynomial-time complexity and high numerical precision.

Algorithm 2: Solving the Generalized Heron Problem via SOCP

Input: SOC-representable convex sets $S, C_1, \dots, C_m$

Output: Optimal solution x^* and value $F(x^*)$

Procedure SolveGHP($S, C_1, \dots, C_m$)

 Define variables $x \in \mathbb{R}^n, y_i \in \mathbb{R}^n, t_i \in \mathbb{R}, i = 1, \dots, m$

 Objective: minimize $\sum_{i=1}^m t_i$

 Constraints: $x \in S, y_i \in C_i, \|x - y_i\| \leq t_i, t_i \geq 0$

 Solve SOCP with a conic solver (e.g., MOSEK, ECOS)

 Return $x^*, \{y_i^*\}, F(x^*) = \sum_i t_i^*$

end procedure

5. Numerical Illustrations

We now present four numerical examples that test the proposed methods under varied settings. The first two involve simple two-dimensional problems, while the third and fourth increase complexity and scale. Together, these cases illustrate the accuracy, efficiency, and robustness of PSM-ELS, MM, the basic subgradient method, and the SOCP reformulation.

Experimental Setup.

All experiments were conducted in Python 3.11 on Google Colab, utilizing an Intel(R) Xeon(R) CPU @ 2.20 GHz with 12.7 GB of RAM. For the iterative methods (PSM, MM, and PSM-ELS), we adopted the stopping criterion $F(x^{(k+1)}) - F(x^{(k)}) < 10^{-15}$, where $F(x)$ is the objective function of (1).

Example 5.1 (Four Squares with a Disk Constraint [11, Example 4.2]). We study problem (1) in $\mathbb{R}^2$, where the target sets C_i are axis-aligned squares of half-side $r_i = 1$, centered at $(-7,1)$, $(-5,-8)$, $(4,7)$, and $(5,1)$. The feasible region is the closed disk $S = B_{1.5}[(-3,4)]$, and the starting point is $x^{(0)} = (-3,5.5) \in S$.

The generalized Heron problem in $\mathbb{R}^2$ takes the form

$$\min_{x \in S} \sum_{i=1}^4 d_{C_i}(x).$$

Since, by Remark 16, whenever both the objective f and the feasible sets C_i are SOC-representable the problem can be cast as a conic program, we proceed as follows. We introduce auxiliary points $y_i \in C_i$ and nonnegative scalars t_i such that

$$\|x - y_i\| \leq t_i.$$

With these substitutions, the problem is equivalent to the conic formulation

$$(9) \min_{x, \{y_i, t_i\}} \sum_{i=1}^4 t_i \text{ s.t. } x \in S, y_i \in C_i, \|x - y_i\| \leq t_i, t_i \geq 0 \ (i = 1, \dots, 4)$$

The equivalence is standard: for any feasible (x, y_i, t_i) , $t_i \geq \|x - y_i\| \geq d_{C_i}(x)$, hence $\sum_i t_i \geq \sum_i d_{C_i}(x)$. At an optimizer x^* , setting $y_i^* = \text{proj}_{C_i}(x^*)$ and $t_i^* = \|x^* - y_i^*\|$ yields equality.

We now verify that each constraint set in (9) is indeed SOC-representable. Similar verifications are carried out for all subsequent examples.

1. **Disk S :** Let $x = (x_1, x_2)$, $x_S = (-3,4)$, $R_S = 1.5$. Then $x \in S$ iff $\|x - x_S\| \leq R_S$, i.e.,

$$\|x + (3, -4)\| \leq 1.5.$$

This is a standard Lorentz cone constraint in $\mathbb{R}^3$ with no auxiliary variables.

2. **Squares C_i :** Each C_i is the axis-aligned square of half-side $r_i = 1$ centered at $c_i = (c_{i,1}, c_{i,2})$. Hence

$$c_{i,1} - 1 \leq y_{i,1} \leq c_{i,1} + 1, \quad c_{i,2} - 1 \leq y_{i,2} \leq c_{i,2} + 1.$$

These correspond to four linear inequalities per i . Any inequality $a^\top z \leq b$ admits the degenerate SOC form

$$\|0\| \leq b - a^\top z,$$

which is the canonical SOC-representable (5), with $A = 0$, $b_0 = 0$, $c = -a$, $d = b$. Thus each C_i is SOC-representable (indeed polyhedral).

3. **Distance constraints:** For each i , the condition $\|x - y_i\| \leq t_i$ is directly a Lorentz cone constraint in $\mathbb{R}^3$ with right-hand side t_i .
4. **Nonnegativity:** Each $t_i \geq 0$ is a degenerate SOC constraint of the form $\|0\| \leq t_i$.

By solving the SOCP reformulation with the ECOS solver, we obtain the reference solution $x^* \approx (-2.040135, 2.847325)$ and objective value $F^* \approx 26.1341859$.

Table 1: Performance metrics for Example 5.1

Method	Objective Value	CPU Time (s)	Iterations
Subgradient (Basic)	26.1341859065	14.254	90,626
MM Algorithm	26.1341859059	0.004	17
PSM-ELS (Subgrad + Exact LS)	26.1341859059	0.018	6
SOCP (ECOS)	26.1341858958	0.022	N/A

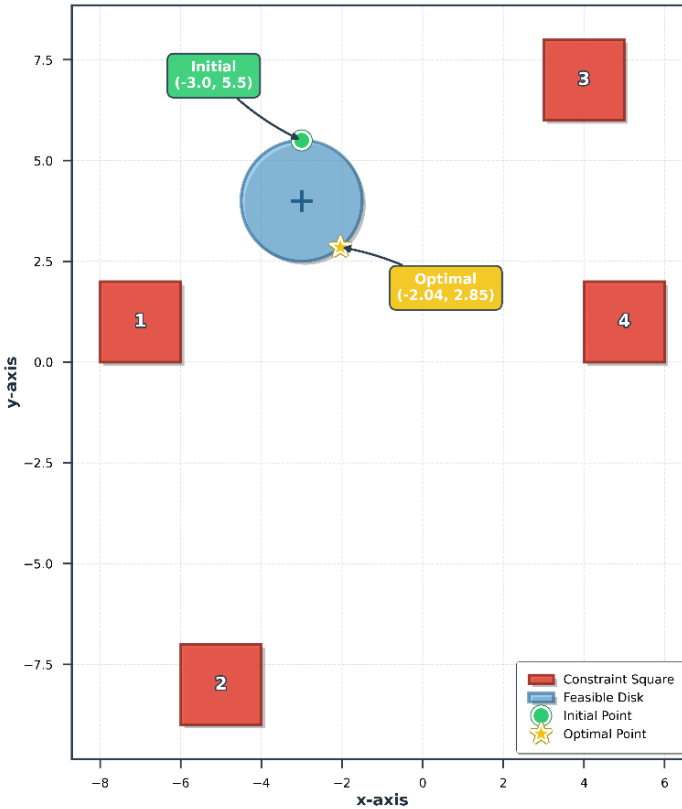


Figure 1: Geometry of Example 5.1

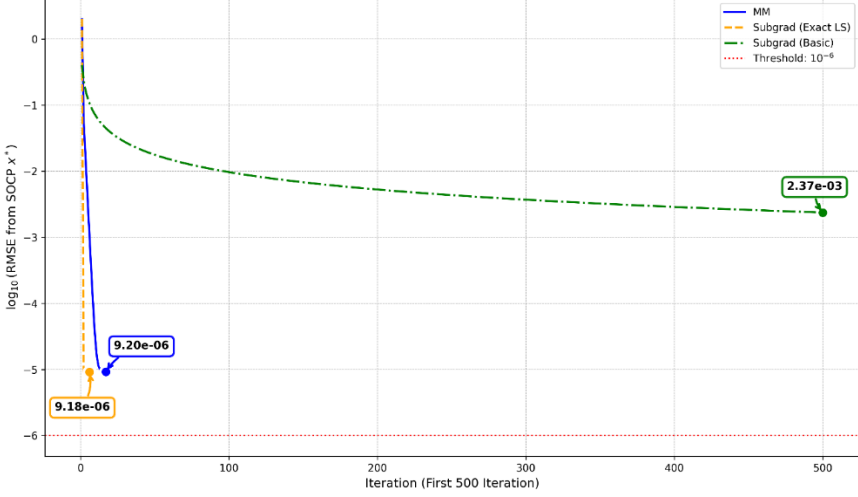


Figure 2: RMSE ($\log_{10}$ scale) vs. iteration for Example 5.1.

Among iterative methods, **PSM-ELS** converged to the same accuracy in only **6 iterations** (0.018 s), substantially faster than MM (17 iterations, 0.004 s) and vastly more efficient than the basic subgradient method ($> 90,000$ iterations, 14 s, with final RMSE $\approx 2.37 \times 10^{-3}$). The RMSE curve (Fig. 2) shows rapid monotone decay for PSM-ELS and MM, reaching the 10^{-6} threshold consistent with the SOCP benchmark. By contrast, the basic subgradient method converges very slowly near its characteristic sublinear rate.

Example 5.2 (Five Cubes with a Ball Constraint²). We next consider (1) in $\mathbb{R}^3$, where the target sets C_i are axis-aligned cubes with half-side $r_i = 1$, centered at $(0, -4, 0)$, $(-4, 2, -3)$, $(-3, -4, 2)$, $(-5, 4, 4)$, and $(-1, 8, 1)$. The feasible region is the closed unit ball centered at $(0, 2, 0)$. The starting point is $x^{(0)} = (0, 2, 0) \in S$. The benchmark solution is

$$x^* \approx (-0.92531, 1.62907, 0.07883), \quad F^* \approx 22.23480.$$

Table 2 and Fig. 4 summarize the results.

The SOCP solver computed F^* in ≈ 0.076 s. PSM-ELS achieved the same accuracy in just **9 iterations** (0.019 s), outperforming MM (13 iterations, 0.003 s) and far surpassing the basic subgradient method (over 1,100 iterations, 0.55 s). Both PSM-ELS and MM show fast, monotone error decay to 10^{-6} , while the basic subgradient follows its $O(k^{-1/2})$ trajectory and converges much more slowly.

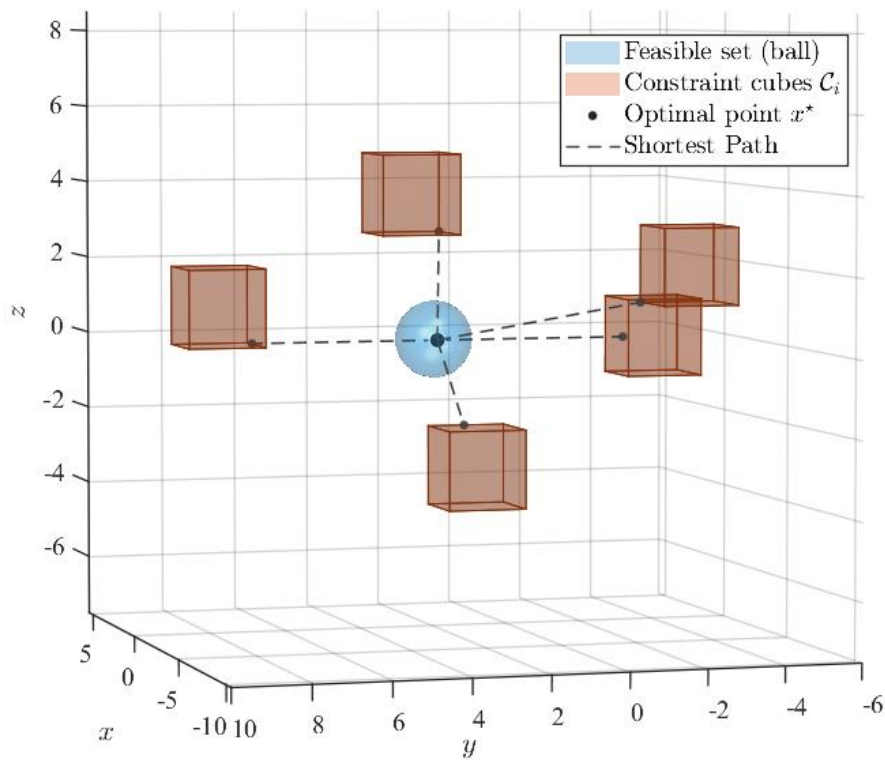


Figure 3: Geometry of Example 5.2.

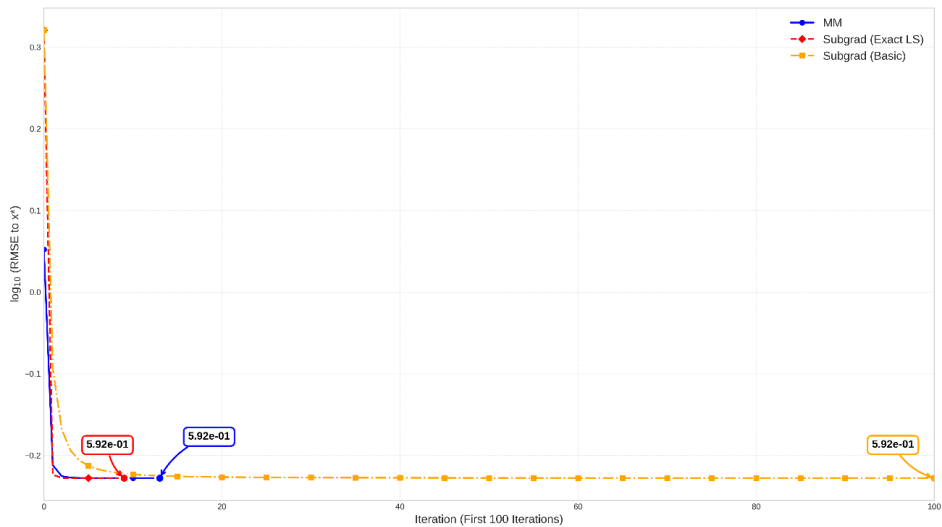


Figure 4: RMSE ($\log_{10}$ scale) vs. iteration for Example 5.2.

Example 5.3 (Generalized Heron Problem in $\mathbb{R}^2$). We solve the generalized Heron problem in $\mathbb{R}^2$ with eight closed unit balls $C_k = B_1[c_k]$ centered at $c_1 = (4,0)$, $c_2 = (-4,0)$, $c_3 = (0,4)$, $c_4 = (0,-4)$, $c_5 = (2,2)$, $c_6 = (-2,2)$, $c_7 = (-2,-2)$, $c_8 = (2,-2)$, and feasible region the unit disk $S = B_1[(0,0)]$. The initial point is $x^{(0)} = (-1,0) \in S$. The analytical solution is $x^* = (0,0)$ with benchmark objective value $F^* = \sum_{i=1}^8 d_{C_i}(x^*) \approx 19.3137085$ (Case 1¹⁴).

All algorithms from Sections 3–4 were run with the stopping rule $|F(x_{k+1}) - F(x_k)| < 10^{-15}$. Table 3 and Fig. 6 summarize performance. The SOCP formulation (solved by ECOS) provides a high-accuracy *reference* solution and objective value; it reached F^* in 0.041 s. Among iterative methods, the proposed **PSM-ELS** (projected subgradient with exact line search) converged to the same accuracy in only **2 iterations** and 0.0045 s, substantially faster than both the MM scheme and the basic projected subgradient method. The MM algorithm was also reliable, taking 40 iterations and 0.016 s. In contrast, the basic subgradient method exhibited its characteristic $O(k^{-1/2})$ behavior, requiring **5,513** iterations and 0.85 s to approach the target accuracy.

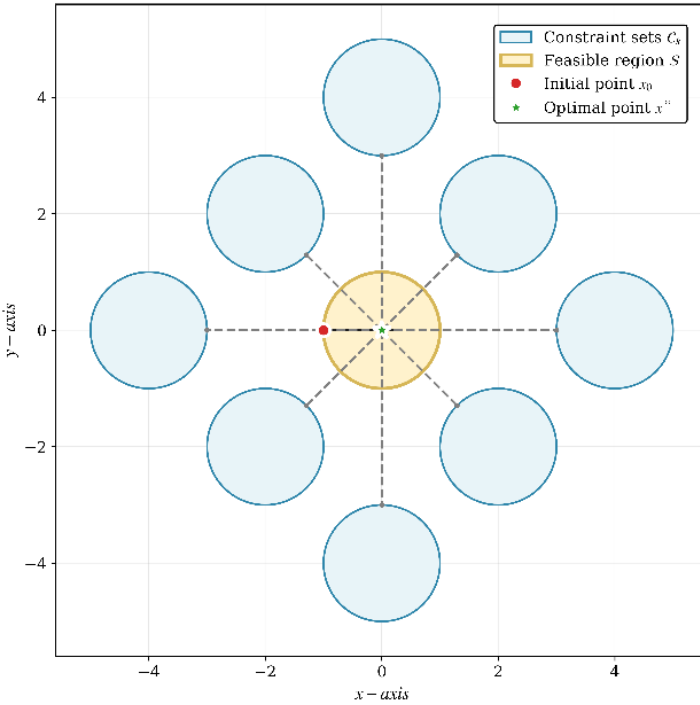


Figure 5: Geometry of Example 5.3.

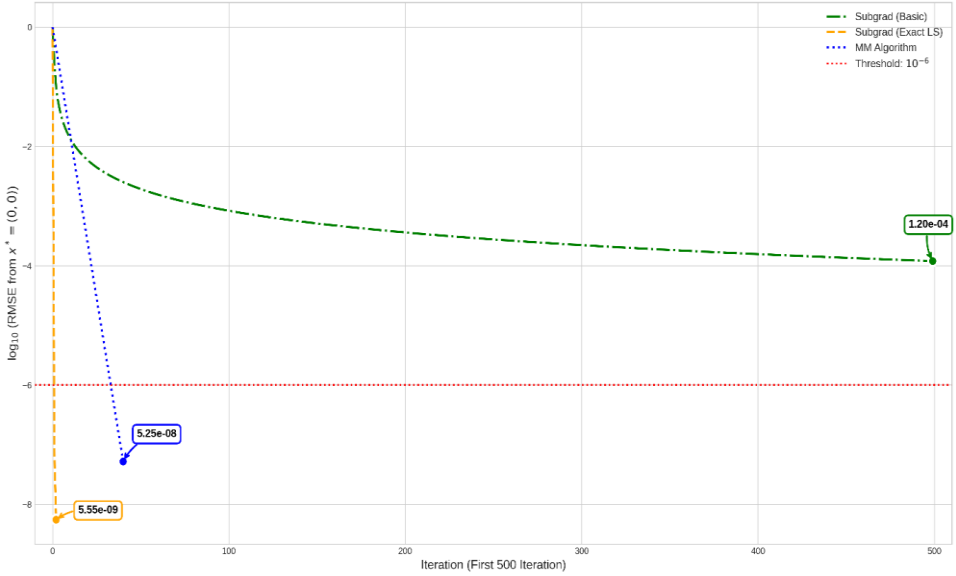


Figure 6: RMSE ($\log_{10}$ scale) vs. iteration for Example 5.3.

Figure 6 plots $\log_{10}(\|x^{(k)} - x^*\|)$ versus iteration. Both PSM-ELS and MM display rapid, monotone decay, stabilizing well below the 10^{-6} threshold (dashed red line) in agreement with the SOCP benchmark; the annotations show terminal RMSEs $\approx 5.6 \times 10^{-9}$ (PSM-ELS) and $\approx 5.3 \times 10^{-8}$ (MM). The basic subgradient method decreases much more slowly and remains around $10^{-3.8}$ after 500 iterations.

Table 3: Performance metrics for Example 5.3.

Method	Objective Value	CPU Time (s)	Iterations
Subgradient (Basic)	19.3137084990	0.854435	5,513
MM Algorithm	19.3137084990	0.015533	40
PSM-ELS (Subgrad + Exact LS)	19.3137084990	0.004521	2
SOCP (ECOS Solver)	19.3137084977	0.041053	N/A

Example 5.4 (Generalized Heron Problem in $\mathbb{R}^{25}$ — Case 2¹⁴). We consider the generalized Heron problem in $\mathbb{R}^{25}$, where the feasible set is the closed ball $C = B_1[c]$ of radius 1 centered at $c = (2.5, 4.5, 3.5, 6.5, 4.5, 12.5, 8.5, 10.5, 5.5, 8.5, 6.5, 11.5, 4.5, 8.5, 3.5, 6.5, 2.5, 12.5, 6.5, 8.5, 4.5, 10.5, 5.5, 2.5, 6.5)$.

There are $n = 10$ target sets $C_k = B_1[c_k]$, each a closed ball of radius 1 with center $c_k \in \mathbb{R}^{25}$ given by

$$\begin{aligned}
c_1 &= (6, 4, 3, 6, 4, 12, 8, 10, 5, 8, 6, 11, 4, 8, 3, 6, 2, 12, 6, 8, 4, 10, 5, 2, 6), \\
c_2 &= (-2, 4, 3, 6, 4, 12, 8, 10, 5, 8, 6, 11, 4, 8, 3, 6, 2, 12, 6, 8, 4, 10, 5, 2, 6), \\
c_3 &= (2, -4, 3, 6, 4, 12, 8, 10, 5, 8, 6, 11, 4, 8, 3, 6, 2, 12, 6, 8, 4, 10, 5, 2, 6), \\
c_4 &= (2, 4, -3, 6, 4, 12, 8, 10, 5, 8, 6, 11, 4, 8, 3, 6, 2, 12, 6, 8, 4, 10, 5, 2, 6), \\
c_5 &= (2, 4, 3, -6, 4, 12, 8, 10, 5, 8, 6, 11, 4, 8, 3, 6, 2, 12, 6, 8, 4, 10, 5, 2, 6), \\
c_6 &= (2, 4, 3, 6, -4, 12, 8, 10, 5, 8, 6, 11, 4, 8, 3, 6, 2, 12, 6, 8, 4, 10, 5, 2, 6), \\
c_7 &= (2, 4, 3, 6, 4, -12, 8, 10, 5, 8, 6, 11, 4, 8, 3, 6, 2, 12, 6, 8, 4, 10, 5, 2, 6), \\
c_8 &= (2, 4, 3, 6, 4, 12, -8, 10, 5, 8, 6, 11, 4, 8, 3, 6, 2, 12, 6, 8, 4, 10, 5, 2, 6), \\
c_9 &= (2, 4, 3, 6, 4, 12, 8, -10, 5, 8, 6, 11, 4, 8, 3, 6, 2, 12, 6, 8, 4, 10, 5, 2, 6), \\
c_{10} &= (2, 4, 3, 6, 4, 12, 8, 10, -5, 8, 6, 11, 4, 8, 3, 6, 2, 12, 6, 8, 4, 10, 5, 2, 6).
\end{aligned}$$

Table 4 and Fig. 7 show that the SOCP solver (ECOS) provides the benchmark $F^* \approx 105.16415296$ in 0.045 s. All iterative schemes matched this value to within 10^{-12} . Among them, **PSM-ELS** was most iteration-efficient, reaching the benchmark in only 5 iterations (0.014 s). MM also converged reliably in 12 iterations with the shortest runtime (0.0069 s), while the basic subgradient method required 40 iterations (0.0072 s), consistent with its slower sublinear rate. The RMSE curves confirm these trends: PSM-ELS attains the residual floor fastest, MM follows, and the basic subgradient needs substantially more updates. These results highlight both the scalability of PSM-ELS in high dimensions and the value of SOCP formulations as accuracy certificates and performance baselines.

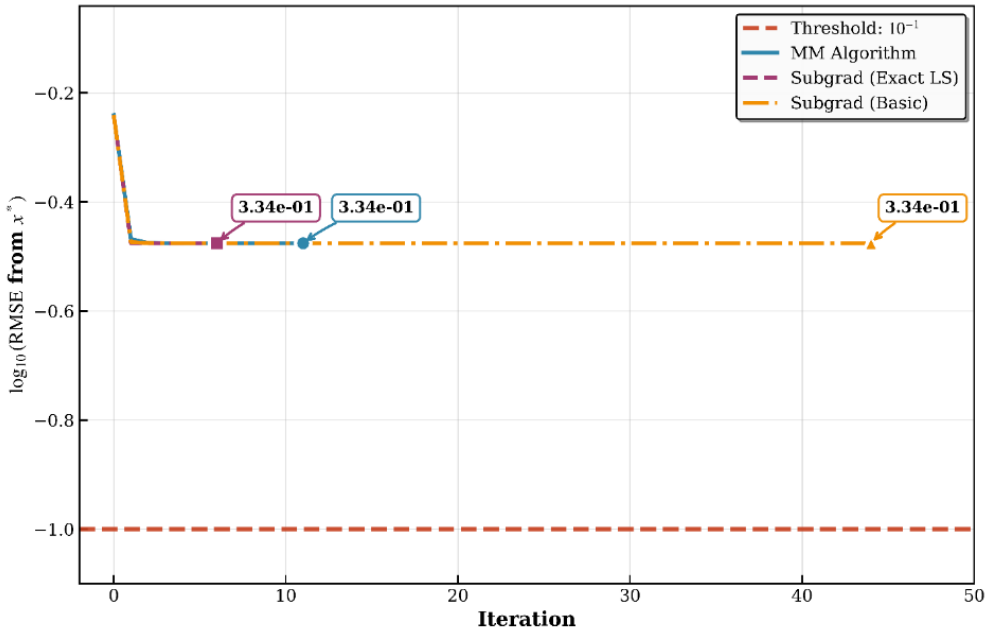


Figure 7: RMSE versus iteration for Example 4.

Performance metrics for 25

Method	Objective Value	CPU Time (s)	Iterations
Subgradient (Basic)	105.164154921925160	0.007231	40
MM Algorithm	105.164154921925103	0.006884	12
PSM-ELS (Subgrad + Exact LS)	105.164154921925089	0.014116	5
SOCP (ECOS Solver)	105.164152962032361	0.045189	N/A

6. Advantages of the Proposed Method

The proposed Projected Subgradient Method with Exact Line Search (PSM-ELS) offers several advantages over traditional approaches such as the classical projected subgradient method and the majorization–minimization (MM) algorithm. These advantages can be classified into two categories: theoretical and practical.

Theoretical Advantages

- **Faster convergence.** The adaptive exact line search selects step sizes optimally along feasible rays, enabling much larger and more effective steps in early iterations compared with the conservative diminishing step-size rules used in classical subgradient methods.
- **Guaranteed descent property.** By solving a one-dimensional convex optimization problem at each step, the method ensures that the objective function decreases monotonically. This eliminates the risk of stagnation due to overly small step sizes.
- **Robustness to nonsmoothness.** Unlike MM, which relies on Euclidean surrogate models, PSM-ELS is firmly grounded in convex analysis. It retains its convergence guarantees even when the distance-sum objective is nonsmooth or the constraint set has complex geometry.

Practical Advantages

- **Reduced computational cost.** Because convergence is reached in only a handful of iterations, the overall CPU time is significantly lower than for the classical subgradient method.
- **Stability.** Exact line search automatically avoids oscillations that often occur with poorly tuned or overly aggressive step sizes, ensuring smooth and consistent convergence.
- **Wide applicability.** The method is not restricted to Euclidean surrogate structures, making it applicable to a much broader class of generalized Heron problems and related convex optimization models.

- **Simplicity of implementation.** Each iteration requires only projection computations and the solution of a simple one-dimensional root-finding problem, making the algorithm straightforward to implement and integrate with existing optimization frameworks.
- **Scalability.** Numerical experiments demonstrate that the method maintains its efficiency and accuracy across dimensions ranging from low-dimensional settings in $\mathbb{R}^2$ and $\mathbb{R}^3$ to higher-dimensional problems in $\mathbb{R}^{25}$.

7. Conclusion

In this paper we proposed two novel computational approaches for the Generalized Heron Problem: (i) a projected subgradient method with exact line search (PSM-ELS), and (ii) a second-order cone programming (SOCP) reformulation.

Our theoretical analysis established that exact line search transforms the classical subgradient method into a monotone descent algorithm, eliminating the inefficiency of diminishing step rules. Numerical experiments across diverse examples in $\mathbb{R}^2$, $\mathbb{R}^3$, and $\mathbb{R}^{25}$ consistently confirmed that PSM-ELS achieves benchmark accuracy in the fewest iterations, with rapid RMSE decay and competitive runtimes. Compared with the classical subgradient method, PSM-ELS delivers orders-of-magnitude acceleration; compared with MM, it enjoys broader applicability; and compared with SOCP, it offers a lightweight, scalable alternative suitable for larger problems.

These findings highlight the practical and theoretical significance of PSM-ELS as a versatile tool for nonsmooth convex optimization of distance-sum type. Promising avenues for future research include: (i) accelerated or stochastic variants for very large-scale data, (ii) extensions to non-Euclidean norms or manifold-constrained problems, and (iii) applications to facility location, imaging, and signal processing where high-dimensional generalized Heron models naturally arise.

Disclosure statement: The authors declare that there are no relevant financial or non-financial competing interests to disclose.

Funding: This research did not receive any funding or grant support.

References

1. E. C. Chi and K. Lange; A look at the generalized Heron problem through the lens of majorization–minimization, *The American Mathematical Monthly*, 121(2) (2014), 95-108.

- doi:[10.4169/amer.math.monthly.121.02.095](https://doi.org/10.4169/amer.math.monthly.121.02.095)
2. B. S. Mordukhovich, N. M. Nam and J. S. Jr; Solving a generalized Heron problem by means of convex analysis, *The American Mathematical Monthly*, 119(2) (2012), 87-99.
doi:[10.4169/amer.math.monthly.119.02.087](https://doi.org/10.4169/amer.math.monthly.119.02.087)
 3. B. S. Mordukhovich, N. M. Nam and J. Salinas; Applications of variational analysis to a generalized Heron problem, *Applicable Analysis*, 91(10) (2011), 1915-1942. doi:[10.1080/00036811.2011.604849](https://doi.org/10.1080/00036811.2011.604849)
 4. B. Mordukhovich and N. M. Nam; Applications of variational analysis to a generalized Fermat–Torricelli problem, *Journal of Optimization Theory and Applications*, 148(3) (2011), 431-454. doi:[10.1007/s10957-010-9761-7](https://doi.org/10.1007/s10957-010-9761-7)
 5. A. Beck; *Introduction to Nonlinear Optimization: Theory, Algorithms, and Applications with MATLAB*, Society for Industrial and Applied Mathematics, 2014. doi:[10.1137/1.9781611973655](https://doi.org/10.1137/1.9781611973655)
 6. D. P. Bertsekas, A. Nedic and A. E. Ozdaglar; *Convex Analysis and Optimization*, Athena Scientific, 2003
 7. A. Dhara and J. Dutta; *Optimality Conditions in Convex Optimization: A Finite-Dimensional View*, CRC Press, 2011. doi:[10.1201/b11156](https://doi.org/10.1201/b11156)
 8. J. B. Hiriart-Urruty and C. Lemaréchal; *Convex Analysis and Minimization Algorithms I: Fundamentals*, Springer Science & Business Media, 1996
 9. R. T. Rockafellar; *Convex Analysis*, Princeton University Press, 1970.
 10. B. Mordukhovich and N. M. Nam; *An Easy Path to Convex Analysis and Applications*, Springer, 2023. doi:[10.1007/978-3-031-26458-0](https://doi.org/10.1007/978-3-031-26458-0)
 11. S. Boyd and L. Vandenberghe; *Convex Optimization*, Cambridge University Press, 2004.
 12. M. S. Lobo, L. Vandenberghe, S. Boyd and H. Lebrete; Applications of second-order cone programming, *Linear Algebra and its Applications*, 284(1) (1998), 193–228. doi:[https://doi.org/10.1016/S0024-3795\(98\)10032-0](https://doi.org/10.1016/S0024-3795(98)10032-0)
 13. A. Ben-Tal and A. S. Nemirovski; *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*, Society for Industrial and Applied Mathematics, 2001.
 14. D. R. Sahu, S. Sharma, J. C. Yao and X. Zhao; Accelerated iterative splitting methods on Hadamard manifolds, *Optimization*, (2024), 4169-4206. doi:[10.1080/02331934.2024.2399807](https://doi.org/10.1080/02331934.2024.2399807)