Noname manuscript No.
(will be inserted by the editor)

Voronoi Conditional Gradient Method for Constrained Nonconvex Optimization

Abbas Khademi

Abstract The Conditional Gradient method offers a computationally efficient, projection-free framework for constrained problems; however, in nonconvex settings it may converge to stationary points of low quality. We propose the Voronoi Conditional Gradient (VCG) method, a geometric heuristic that systematically explores the feasible region by constructing adaptive Voronoi partitions from previously discovered stationary points. VCG incrementally refines a Voronoi decomposition of the feasible region and initiates new conditional gradient runs from interior points of underexplored cells, thereby promoting systematic coverage of the search space. We evaluate VCG on two classes of NP-hard problems and demonstrate that it consistently finds high-quality candidate solutions.

Keywords Conditional Gradient Method · Frank–Wolfe · Voronoi Decomposition · Nonconvex Optimization · Heuristic Search

Mathematics Subject Classification 65K05 · 52C22 · 90C26 · 90C59

1 Introduction

Constrained nonconvex optimization problems arise across a wide range of disciplines, including machine learning, signal processing, engineering design, and operations research [14,29]. A canonical formulation is:

$$\min_{x \in \mathcal{X}} f(x), \tag{P}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable (possibly nonconvex) function and $\mathcal{X} \subset \mathbb{R}^n$ is a compact convex set.

A. Khademi
School of Mathematics and Computer Science, Iran University of Science and Technology, Tehran, Iran.
E-mail: abbaskhademi92@gmail.com
https://orcid.org/0000-0002-8276-6821

The Conditional Gradient (CG) method—also known as the Frank–Wolfe algorithm—is a projection-free optimization framework [8,15,25] that is often the method of choice for such problems, especially when projections onto $\mathcal{X}$ are computationally prohibitive. Its primary appeal lies in requiring only two ingredients per iteration: access to the gradient $\nabla f(x)$ and a linear minimization oracle over $\mathcal{X}$, i.e., the ability to solve $\min_{s\in\mathcal{X}}\langle\nabla f(x), s\rangle$. For many structured constraint sets (e.g., simplices, nuclear-norm balls, or polytopes), this linear subproblem admits highly efficient solutions [7,11,17].

When f is convex, numerous Conditional Gradient (CG) variants efficiently solve problem (P) in terms of the functional value gap [9]. However, for nonconvex f, the method's theoretical guarantees are limited to convergence to first-order stationary points [21,24], offering no assurance regarding solution quality. This is a critical limitation: such stationary points may be of poor quality and far from meaningful local minimizers. Although recent CG variants address structured nonconvexity [28], extend to nonsmooth objectives [12], or handle quasar-convex settings [22,27], they all rely on specific structural assumptions about the objective or constraints. In contrast, we propose a fundamentally different paradigm—based on geometric partitioning—that requires no such assumptions.

While restart-based or perturbation strategies have been explored to escape poor stationary points [19,32], they often lack principled guidance for selecting restart locations. In contrast, space-partitioning methods from global optimization [18,26] offer systematic exploration but typically require function evaluations or operations—such as projections or domain-specific sampling—that fall outside the projection-free oracle model of CG.

In certain special cases (e.g., when f is concave), the authors of [3] propose a tailored initialization strategy that yields high-quality solutions with CG. However, this approach relies on strong structural assumptions and does not extend to general nonconvex problems.

We bridge this gap by integrating adaptive Voronoi partitioning directly into the projection-free CG paradigm, enabling systematic exploration without requiring problem-specific structure. In this paper, we introduce a geometry-driven heuristic that dynamically partitions the search space to escape poor local minima. The method operates iteratively: each time a stationary point is found via a standard Conditional Gradient run, it is added to a growing set of known solutions. This set then serves as the set of generating points for a Voronoi decomposition of the feasible region $\mathcal{X}$.

Within the intersection of each Voronoi cell with $\mathcal{X}$, we select a new initialization point by solving a simple auxiliary optimization problem that maximizes its distance to the cell's boundaries. These interior points form a diverse and well-distributed set of starting locations for subsequent CG runs, effectively steering the search toward underexplored regions of the feasible set.

The main contributions of this work are threefold. First, we propose a novel algorithm that integrates the classical CG method with adaptive Voronoi partitioning to enable systematic exploration in nonconvex optimization. Second, the method remains entirely projection-free, preserving the computational ef-

ficiency of CG. Third, we provide numerical evidence on indefinite quadratic programming and sigmoid regression problems, demonstrating that the proposed approach consistently yields higher-quality solutions.

The remainder of this paper is organized as follows. Section 2 briefly reviews the CG algorithm and Voronoi partitioning preliminaries. Section 3 details the proposed VCG framework. Section 4 presents numerical experiments and results. Finally, Section 5 summarizes our findings and concludes the paper.

2 Preliminaries

This section outlines the problem setting and introduces the core concepts our method builds upon: the Conditional Gradient method and Voronoi partitioning.

2.1 Optimization Problem

We address the nonconvex optimization problem:

$$\min_{x \in \mathcal{X}} f(x) \tag{P}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable and potentially nonconvex function, and $\mathcal{X} \subset \mathbb{R}^n$ is a compact convex set. We assume the objective function f is L-smooth, meaning its gradient ∇f is Lipschitz continuous:

$$\|\nabla f(x) - \nabla f(y)\|_* \leq L\|x - y\| \quad \forall x, y \in \mathcal{X}, \tag{1}$$

where $\|\cdot\|_*$ is the dual norm of $\|\cdot\|$.

2.2 Conditional Gradient Method

The Conditional Gradient (CG) method is an iterative, projection-free method for constrained optimization [15,17]. At iteration k, given the current point $x^k \in \mathcal{X}$, it solves the Linear Minimization Oracle (LMO)

$$s^k \in \operatorname*{argmin}_{s \in \mathcal{X}} \langle \nabla f(x^k), s \rangle, \tag{2}$$

and updates $x^{k+1} = (1 - t_k)x^k + t_k s^k$, where the step size $t_k \in [0, 1]$. For the nonconvex problem (P), CG converges to a stationary point [8,22,23,31]. The CG method with a given step-size rule is summarized in Algorithm 1.

The choice of the step size $t_k \in [0, 1]$ critically affects CG convergence and practical performance. Adaptive step-size approaches, such as those introduced in [22,31], dynamically tune t_k according to the problem's local geometry, significantly improving convergence speed in practice; moreover, they do not require knowledge of the exact gradient Lipschitz constant L.

Algorithm 1 Conditional Gradient Method

Input: Initial point $x^0 \in \mathcal{X}$, tolerance $\varepsilon_1 > 0$, max iterations $K_{\max}$
for $k = 0$ to $K_{\max} - 1$ **do**
  $s^k \leftarrow \operatorname{argmin}_{s \in \mathcal{X}} \langle \nabla f(x^k), s \rangle$
  $g_k \leftarrow \langle \nabla f(x^k), x^k - s^k \rangle$
  if $g_k < \varepsilon_1$ **then**
    return x^k
  end if
  Select step size $t_k \in [0, 1]$
  $x^{k+1} \leftarrow (1 - t_k)x^k + t_k s^k$
end for
return $x^{K_{\max}}$

2.3 Voronoi Partitioning

To prevent our algorithm from repeatedly converging to the same stationary point, we introduce a systematic adaptive exploration strategy based on Voronoi partitioning. The concept of partitioning a space based on proximity to a set of points, known as a Voronoi diagram or tessellation, is a fundamental structure in computational geometry [2,4,30].

To explore diverse regions of $\mathcal{X}$, we partition the feasible set into Voronoi cells based on previously found stationary points $A = \{p^1, \cdots, p^K\} \subset \mathcal{X}$. The Voronoi cell $\mathcal{V}_i$ for point p^i is

$$\mathcal{V}_i = \left\{ x \in \mathcal{X} : \|x - p^i\|_2 \leq \|x - p^j\|_2 \quad \forall j \neq i \right\}, \tag{3}$$

which is convex as it is the intersection of $\mathcal{X}$ with half-spaces defined by

$$(p^j - p^i)^\top x \leq \frac{1}{2}(p^j - p^i)^\top (p^j + p^i).$$

A visual example of a Voronoi diagram with points is shown in Figure 1. This figure illustrates a Voronoi partition of a two-dimensional box $\mathcal{X}$ induced by a finite set of points (marked as dots). Each Voronoi cell $\mathcal{V}_i$ corresponds to the region of space that is closer to its associated generator p^i than to any other.

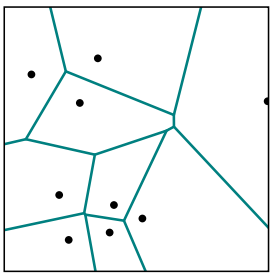

Fig. 1 Voronoi partitioning of a box set $\mathcal{X}$ with generator points p^i. Each cell $\mathcal{V}_i$ contains points closer to p^i than to any other p^j.

For each cell $\mathcal{V}_i$, defined by the affine constraints

$$g_j(x) := (p^j - p^i)^\top x - \frac{1}{2}(p^j - p^i)^\top (p^j + p^i) \leq 0, \quad j \neq i, \tag{4}$$

We compute an interior point by solving the following auxiliary problem

$$\begin{aligned} \min_{x \in \mathcal{X}, \tau \in \mathbb{R}} \quad & \tau \\ \text{s.t.} \quad & g_j(x) \leq \tau, \quad \forall j \neq i. \end{aligned} \tag{5}$$

Proposition 1 *Assume $\mathcal{X} \subset \mathbb{R}^n$ is a full-dimensional compact convex set. Let (x^*, τ^*) be an optimal solution to problem (5). If $\tau^* < 0$, then x^* lies in the interior of $\mathcal{V}_i$ relative to its affine hull.*

Proof Since $\mathcal{X}$ is compact and convex and the constraints in (5) are affine, the feasible set is compact and convex, guaranteeing existence of an optimal solution. If $\tau^* < 0$, then $g_j(x^*) \leq \tau^* < 0$ for all $j \neq i$, meaning x^* strictly satisfies all inequalities defining $\mathcal{V}_i$. By full-dimensional assumption, $\mathcal{X}$ has non-empty interior, and strict inequality constraints are preserved under small perturbations. Therefore, there exists $\delta > 0$ such that the ball $B(x^*, \delta) \cap \mathcal{X} \subset \mathcal{V}_i$, implying x^* is in the relative interior of $\mathcal{V}_i$. □

3 Voronoi Conditional Gradient Method

The Voronoi Conditional Gradient (VCG) method iteratively combines local search with global exploration. Starting from an initial point, it applies the CG method to find a stationary point. The feasible set $\mathcal{X}$ is then partitioned into Voronoi cells based on all discovered stationary points. For each cell, an interior point is computed to initialize a new CG run over $\mathcal{X}$, ensuring diverse exploration. The process repeats until no new distinct stationary points are found. Algorithm 2 formalizes the procedure.

We include x^0 in the initial center set to discourage re-initialization near the starting point, which may lie in a poorly explored region. However, in practice, this has minimal impact once multiple stationary points are discovered.

VCG leverages CG's convergence to stationary points [22,24,31]. Voronoi-based initialization ensures new runs start in regions distant from known solutions, empirically covering more basins than random multi-starts, as shown in Section 4. While global optimality is not guaranteed, the geometric diversity enhances solution quality.

We now establish some properties of the VCG algorithm.

Proposition 2 *Let $A^{(\ell)}$ denote the set of centers at the beginning of level ℓ. For any $x \in \mathcal{X}$, define the coverage radius as:*

$$r^{(\ell)}(x) := \min_{p \in A^{(\ell)}} \|x - p\|_2. \tag{6}$$

Then either $A^{(\ell+1)} = A^{(\ell)}$ (termination), or there exists a point $x \in \mathcal{X}$ such that $r^{(\ell+1)}(x) < r^{(\ell)}(x)$.

Algorithm 2 Voronoi Conditional Gradient (VCG) Method

1: **Input:** Objective f, feasible set $\mathcal{X}$, max levels T, tolerance ε
2: **Initialize:** $A \leftarrow \emptyset$, $\mathcal{S} \leftarrow \emptyset$, pick initial $p^0 \in \mathcal{X}$ ▷ Initial feasible point
3: Run CG from p^0 to obtain stationary point p^1
4: $\mathcal{S} \leftarrow \mathcal{S} \cup \{(p^1, f(p^1))\}$, $A \leftarrow A \cup \{p^0, p^1\}$
5: **for** $\ell = 1$ to T **do**
6: Construct Voronoi cells $\{\mathcal{V}_i\}_{i=1}^{|A|}$ using centers A via (3)
7: $A^{\text{new}} \leftarrow \emptyset$ ▷ Collect new stationary points
8: **for** each Voronoi cell $\mathcal{V}_i$, $i = 1, \ldots, |A|$ **do**
9: Solve slack problem (5) to get (x^*, τ^*)
10: **if** $\tau^* < 0$ **then** ▷ Cell has non-empty interior
11: Run CG from x^* to obtain x^{stat}
12: $\mathcal{S} \leftarrow \mathcal{S} \cup \{(x^{\text{stat}}, f(x^{\text{stat}}))\}$
13: $A^{\text{new}} \leftarrow A^{\text{new}} \cup \{x^{\text{stat}}\}$
14: **end if**
15: **end for**
16: **if** $A^{\text{new}} = \emptyset$ **then**
17: **break** ▷ No new distinct points found
18: **end if**
19: $A \leftarrow A \cup \{x \in A^{\text{new}} : \min_{p \in A} \|x - p\|_2 > \varepsilon\}$ ▷ Add ε-distinct points
20: **end for**
21: **Output:** $x^* = \text{argmin}\{v : (x, v) \in \mathcal{S}\}$ ▷ Best solution found

Proof If $A^{(\ell+1)} = A^{(\ell)}$, the algorithm terminates at line 16. Otherwise, $A^{(\ell+1)} = A^{(\ell)} \cup A^{\text{new}}$ with $A^{\text{new}} \neq \emptyset$. Let $p^{\text{new}} \in A^{\text{new}}$. By the distinctness criterion (line 15), $\min_{p \in A^{(\ell)}} \|p^{\text{new}} - p\|_2 > \varepsilon$. Therefore, $r^{(\ell+1)}(p^{\text{new}}) = 0 < \varepsilon < r^{(\ell)}(p^{\text{new}})$. □

Corollary 1 *The cardinality of the center set is monotonically non-decreasing:* $|A^{(\ell+1)}| \geq |A^{(\ell)}|$ *with strict inequality unless termination occurs.*

Under the L-smoothness assumption, if CG uses appropriate step sizes (e.g., backtracking line search or adaptive rules), then either the algorithm terminates at a stationary point or $\lim_{k \to \infty} g_k = 0$. Any accumulation point x^* satisfies $g(x^*) = 0$, i.e., it is a first-order stationary point [22,23]. Thus, the output of VCG is a stationary point.

4 Numerical Experiments

This section evaluates the empirical performance of the proposed VCG method on two nonconvex optimization problems, comparing its performance with the standard CG method and a multi-start CG baseline. All algorithms were implemented in the Julia programming language [5] and executed on Google Colaboratory [6]. For both the CG and VCG methods, we employed an adaptive step-size strategy with the parameters $\varepsilon_1 = 10^{-6}$, $\varepsilon = 10^{-3}$, $K_{\max} = 1000$, and $T = 8$. The auxiliary problem (5) within the VCG algorithm was solved using the HiGHS solver [16] via the JuMP modeling language [13].

4.1 Standard Quadratic Programming

We consider the nonconvex standard quadratic program:

$$\min_{x \in \Delta^n} x^\top Q x, \tag{StQP}$$

where $\Delta^n = \{x \in \mathbb{R}^n : \sum_{i=1}^n x_i = 1, x \geq 0\}$ is the standard n-dimensional simplex, and $Q \in \mathbb{R}^{n \times n}$ is an indefinite symmetric matrix. Problem (StQP) is known to be NP-hard [1]. The feasible set is compact and convex, and the linear minimization oracle required by CG is highly efficient, simply finding the index of the minimum component of the gradient vector.

For our numerical evaluation, we use five instances of dimension $n = 900$ from the dataset in [20]. We compare the performance of VCG against two baseline approaches: the standard CG method, initialized at a single random feasible point, and a multi-start CG variant that performs repeated CG runs from independently sampled initial points. To ensure a fair comparison, the total runtime of multi-start CG is capped at the full execution time of VCG across all restarts.

The results, summarized in Table 1, report objective values, computational runtimes, and the number of points evaluated, alongside the best-known bounds from [21], which were obtained using advanced local and global optimization techniques that contain off-the-shelf solvers and state-of-the-art approaches. The best objective value for each instance is highlighted in bold. In this table, for each method, we report the final objective value. Values in parentheses denote (total runtime in seconds, number of local optimizations). For Standard CG, this is a single run. For Multi-start CG, this is the number of independent random starts. For VCG, this is the total count of CG procedures launched, including the initial run and subsequent runs within each explored Voronoi cell.

VCG achieves the best objective value across all instances, consistently outperforming both Standard CG and the multi-start CG baseline in terms of final objective value. Moreover, VCG matches or improves the best upper bounds reported in [21] on these instances, providing strong empirical evidence that the proposed Voronoi-based exploration effectively escapes poor stationary points and identifies higher-quality solutions.

Table 1 Performance Comparison on Standard Quadratic Programming Instances.

Instance	VCG	Multi-start CG	Standard CG	Best UB [21]
`900(0.75)-1`	**-7.4088** (591.0145s, 346)	-7.3955 (591.2040s, 655)	-6.9918 (0.6160s, 1)	**-7.4088**
`900(0.75)-2`	**-7.3517** (700.2640s, 417)	-7.3390 (700.5695s, 770)	-7.0135 (0.6256s, 1)	**-7.3517**
`900(0.75)-3`	**-7.3603** (779.7218s, 432)	-7.2238 (780.3137s, 871)	-6.7512 (0.6535s, 1)	-7.2946
`900(0.75)-4`	**-7.3870** (637.3222s, 392)	-7.3743 (638.0418s, 727)	-7.3444 (0.5406s, 1)	**-7.3870**
`900(0.75)-5`	**-7.3925** (723.0110s, 425)	-7.2688 (723.1018s, 791)	-6.8711 (0.5273s, 1)	-7.3325

4.2 Least Squares Sigmoid Regression

Next, we address the nonconvex machine learning problem of least-squares sigmoid regression, formulated as:

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \ & \frac{1}{m}\sum_{i=1}^{m}\left(y_i - \frac{1}{1+\exp\left(-x^\top a_i\right)}\right)^2 \\ \text{s.t.} \ & \|x\|_1 \le s, \end{aligned} \tag{7}$$

where $\{(a_i, y_i)\}_{i=1}^m$ is a given dataset. Despite the objective's nonconvexity, the feasible set is convex and compact. This structure is well-suited for CG methods, as the required LMO admits an efficient, closed-form solution.

We evaluate performance on nine datasets from LIBSVM [10] (`a1a`–`a9a`). All instances share a feature dimension of $n = 123$, but the sample size m varies from 1,605 to 32,561. We set the regularization parameter $s = 50$ for all experiments.

As shown in Table 2, VCG consistently achieves the lowest objective value across all datasets. Notably, VCG reaches these superior solutions while performing significantly fewer local optimizations than the multi-start baseline, demonstrating superior exploration efficiency. These results underscore VCG's ability to effectively navigate complex, nonconvex landscapes, particularly in high-data regimes where optimization stability is critical.

Table 2 Performance Comparison on Sigmoid Regression Datasets.

Instance	VCG	Multi-start CG	Standard CG
`a1a` ($n = 123,\ m = 1,605$)	**0.9302** (86.0436s, 134)	0.9330 (86.5214s, 244)	0.9503 (0.2348s, 1)
`a2a` ($n = 123,\ m = 2,265$)	**0.9424** (95.8796s, 114)	0.9440 (95.8910s, 196)	0.9562 (0.3798s, 1)
`a3a` ($n = 123,\ m = 3,185$)	**0.9429** (139.5299s, 116)	0.9433 (139.6777s, 214)	0.9567 (0.4965s, 1)
`a4a` ($n = 123,\ m = 4,781$)	**0.9405** (226.3476s, 132)	0.9428 (226.8613s, 222)	0.9558 (0.7924s, 1)
`a5a` ($n = 123,\ m = 6,414$)	**0.9436** (196.0998s, 105)	0.9454 (196.2849s, 161)	0.9587 (1.0944s, 1)
`a6a` ($n = 123,\ m = 11,220$)	**0.9483** (432.1264s, 114)	0.9496 (433.6842s, 204)	0.9622 (1.5427s, 1)
`a7a` ($n = 123,\ m = 16,100$)	**0.9476** (546.1385s, 123)	0.9482 (549.9109s, 183)	0.9622 (2.2073s, 1)
`a8a` ($n = 123,\ m = 22,696$)	**0.9481** (904.5734s, 122)	0.9497 (905.1183s, 215)	0.9631 (3.1900s, 1)
`a9a` ($n = 123,\ m = 32,561$)	**0.7275** (1039.1539s, 122)	0.7283 (1043.7113s, 207)	0.7376 (4.5644s, 1)

5 Conclusion

We introduce the Voronoi Conditional Gradient (VCG) algorithm, a novel enhancement of the Conditional Gradient method tailored for nonconvex optimization. By integrating Voronoi partitioning, VCG systematically explores the feasible set, mitigating the risk of converging to suboptimal local minima inherent in standard Conditional Gradient approaches. Numerical experiments on standard quadratic programming and sigmoid regression demonstrate that VCG consistently achieves superior solution quality. These results highlight the efficacy of geometry-driven exploration in enhancing first-order methods

for nonconvex optimization, offering a scalable and robust framework for applications in machine learning and operations research.

Data Availability

The source code for all experiments is available at this [link].

References

1. Ahmadi, A.A., Zhang, J.: On the complexity of finding a local minimizer of a quadratic function over a polytope. Mathematical Programming pp. 1–10 (2022)
2. Aurenhammer, F.: Voronoi diagrams—a survey of a fundamental geometric data structure. ACM Computing Surveys (CSUR) **23**(3), 345–405 (1991)
3. Ben-Tal, A., Roos, E.: An algorithm for maximizing a convex function based on its minimum. INFORMS Journal on Computing **34**(6), 3200–3214 (2022)
4. de Berg, M., van Kreveld, M., Overmars, M., Schwarzkopf, O.: Computational Geometry: Algorithms and Applications, 3rd edn. Springer-Verlag (2008)
5. Bezanson, J., Edelman, A., Karpinski, S., Shah, V.B.: Julia: A fresh approach to numerical computing. SIAM Review **59**(1), 65–98 (2017)
6. Bisong, E.: Google colaboratory. In: Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners, pp. 59–64. Springer (2019)
7. Bomze, I.M., Rinaldi, F., Zeffiro, D.: Frank–Wolfe and friends: a journey into projection-free first-order optimization methods. Annals of Operations Research **343**(2), 607–638 (2024)
8. Braun, G., Carderera, A., Combettes, C.W., Hassani, H., Karbasi, A., Mokhtari, A., Pokutta, S.: Conditional gradient methods. : 2211.14103 (2025)
9. Braun, G., Carderera, A., Combettes, C.W., Hassani, H., Karbasi, A., Mokhtari, A., Pokutta, S.: Conditional Gradient Methods: From Core Principles to AI Applications. MOS-SIAM Series on Optimization (2025)
10. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST) **2**(3), 1–27 (2011)
11. Combettes, C.W., Pokutta, S.: Complexity of linear minimization and projection on some sets. Operations Research Letters **49**(4), 565–571 (2021)
12. De Oliveira, W.: Short paper-a note on the Frank–Wolfe algorithm for a class of nonconvex and nonsmooth optimization problems. Open Journal of Mathematical Optimization **4**, 1–10 (2023)
13. Dunning, I., Huchette, J., Lubin, M.: JuMP: A modeling language for mathematical optimization. SIAM Review **59**(2), 295–320 (2017)
14. Floudas, C.A.: Encyclopedia of Optimization. Springer-Verlag, Berlin, Heidelberg (2005)
15. Frank, M., Wolfe, P.: An algorithm for quadratic programming. Naval Research Logistics Quarterly **3**(1-2), 95–110 (1956)
16. Huangfu, Q., Hall, J.A.J.: Parallelizing the dual revised simplex method. Mathematical Programming Computation **10**(1), 119–142 (2018)
17. Jaggi, M.: Revisiting Frank-Wolfe: Projection-free subgradient method. Proceedings of the 30th International Conference on Machine Learning (ICML) pp. 427–435 (2013)
18. Jones, D.R., Perttunen, C.D., Stuckman, B.E.: Lipschitzian optimization without the lipschitz constant. Journal of Optimization Theory and Applications **79**(1), 157–181 (1993)
19. Kerdreux, T., d'Aspremont, A., Pokutta, S.: Restarting Frank-Wolfe: faster rates under hölderian error bounds. Journal of Optimization Theory and Applications **192**(3), 799–829 (2022)

20. Khademi, A., Marandi, A.: Code and data repository for quadratic optimization through the lens of adjustable robust optimization (2025). DOI 10.1287/ijoc.2024.0577.cd. Available for download at `https://github.com/INFORMSJoC/2024.0577`
21. Khademi, A., Marandi, A.: Quadratic optimization through the lens of adjustable robust optimization. INFORMS Journal on Computing pp. 1–25 (2025)
22. Khademi, A., Silveti-Falls, A.: Adaptive conditional gradient descent (2025). ArXiv:2510.11440
23. Lacoste-Julien, S.: Convergence rate of Frank-Wolfe for non-convex objectives (2016). :1607.00345
24. Lacoste-Julien, S., Jaggi, M.: On the global linear convergence of Frank-Wolfe optimization variants. Advances in Neural Information Processing Systems 28 (NIPS 2015) pp. 496–504 (2015)
25. Levitin, E.S., Polyak, B.T.: Constrained minimization methods. USSR Computational Mathematics and Mathematical Physics **6**(5), 1–50 (1966)
26. Locatelli, M., Schoen, F.: Global optimization: theory, algorithms, and applications. SIAM (2013)
27. Martínez-Rubio, D.: Smooth quasar-convex optimization with constraints. arXiv:2510.01943 (2025)
28. Maskan, H., Hou, Y., Sra, S., Yurtsever, A.: Revisiting Frank-Wolfe for structured nonconvex optimization (2025). ArXiv:2503.08921
29. Nocedal, J., Wright, S.J.: Numerical Optimization, 2nd edn. Springer Series in Operations Research and Financial Engineering. Springer (2006)
30. Okabe, A., Boots, B., Sugihara, K., Chiu, S.N.: Spatial Tessellations: Concepts and Applications of Voronoi Diagrams, 2nd edn. John Wiley & Sons (2000)
31. Pedregosa, F., Negiar, G., Askari, A., Jaggi, M.: Linearly convergent Frank–Wolfe with backtracking line-search. In: International Conference on Artificial Intelligence and Statistics, pp. 1–10. PMLR (2020)
32. Reddi, S.J., Sra, S., Póczos, B., Smola, A.: Stochastic Frank-Wolfe methods for nonconvex optimization. In: 2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton), pp. 1244–1251. IEEE (2016)