Convergence to Second-Order Stationarity for Constrained Non-Convex Optimization

Maher Nouiehed∗ Jason Lee† Meisam Razaviyayn‡

October 5, 2018

Abstract

We consider the problem of finding an approximate second-order stationary point of a constrained non-convex optimization problem. We first show that, unlike the unconstrained scenario, the vanilla projected gradient descent algorithm may converge to a strict saddle point even when there is only a single linear constraint. We then provide a hardness result by showing that checking $(\epsilon_g, \epsilon_H)$-second order stationarity is NP-hard even in the presence of linear constraints. Despite our hardness result, we identify instances of the problem for which checking second order stationarity can be done efficiently. For such instances, we propose a dynamic second order Frank–Wolfe algorithm which converges to $(\epsilon_g, \epsilon_H)$-second order stationary points in $O(\max\{\epsilon_g^{-2}, \epsilon_H^{-3}\})$ iterations. The proposed algorithm can be used in general constrained non-convex optimization as long as the constrained quadratic sub-problem can be solved efficiently.

1 Introduction

Designing efficient algorithms for non-convex optimization has been an active area of research in recent years, see [1, 6, 7, 8, 9, 10, 11, 14, 15, 30, 34]. For a general non-convex problem, even finding a local optimum of the objective function is NP-Hard in the worst-case scenario [32]. Therefore, in practice, most existing algorithms converge to first or second order stationary points of the objective function. The latter provides stronger guarantees as it constitutes a smaller subset of the critical points of the objective function that includes local and global optima. Moreover, when applied to functions with “nice” geometrical properties, the set of second order stationary points could even be the same as the set of global optima; see [2, 3, 5, 35, 38, 39, 40] for examples of such objective functions.

Most existing algorithms for finding second order stationary points focus on unconstrained optimization problems. As a first order algorithm, [18] shows that noisy-stochastic gradient descent converges to a local optimum of the objective function under strict saddle property. For the vanilla gradient descent algorithm, [28] uses stable manifold theorem to show that gradient descent with random initialization and sufficiently small constant step size converges to the set of second-order stationary points of the objective function, almost surely. More specifically, they show that the set of initial points that converge to a strict saddle point of the objective function is a measure zero set. As a negative result, [17] constructs an example where the simple gradient descent can take exponential number of iterations to escape a strict saddle point. Motivated by this example, [24] proposes a perturbed form of gradient descent that can efficiently escape saddle points under strict saddle property.

∗Department of Industrial and Systems Engineering, University of Southern California.
†Data Science and Operations Department, Marshall School of Business, University of Southern California.
‡Department of Industrial and Systems Engineering, University of Southern California.
Using higher order derivative information of the objective function, [7, 8, 15, 34, 36] propose trust region or cubic regularization methods for finding a second order stationary point in unconstrained optimization problems. More specifically, the traditional trust region method [12, Algorithm 6.1.1] and cubic regularization methods, which are based on the work of [21, 34], are able to converge to second order stationary points. Moreover, [7, 8] proposed the Adaptive Regularization framework using Cubics, also known as (ARC), and established convergence to a second order stationary point. This method computes at each iteration an (approximate) global optimum for a local cubic model which resembles the behavior of the original objective function. They show that ARC requires $O(\epsilon^{-3/2})$ iterations to converge to an $\epsilon$-first-order stationary point, and $O(\epsilon^{-3})$ iterations to reach an $\epsilon$-second-order stationary point. Motivated by these rates, [15] designed a trust region method, entitled TRACE, which has the same iteration complexity bound as ARC. TRACE alters the acceptance criteria adopted in the traditional trust region method, and introduces a new mechanism for updating the radius of the trust region. In a more recent work, [13] developed an algorithm with a dynamic choice for direction and step-size. In particular, the dynamic algorithm decides on the step that offers a more significant reduction in the objective value. A more general framework that contains as special cases the dynamic algorithm, ARC and TRACE was proposed in [14]. This framework uses a set of generic conditions that need to be satisfied at each trial step, and converges to second order stationarity with optimal iteration complexity bound.

For constrained optimization problems, many recent papers propose algorithms that converge to first-order and second-order stationary points. For example, [26] proposed a Frank–Wolfe algorithm that converges to an $\epsilon$-first order stationary point with complexity $O(\epsilon^{-2})$. Another work by [20] shows that projected gradient descent converges to an $\epsilon$-first order stationary point with the same complexity bound. Convergence to second-order stationary points can be achieved by extending some of the aforementioned second or third order methods. For instance, [6] adapted the ARC method and showed that the worst-case function evaluation complexity for converging to an $\epsilon$-first order stationary point is $O(\epsilon^{-3/2})$. Moreover, [10] showed that the same rate of convergence can be achieved for solving general smooth problems involving both non-convex equality and inequality constraints, using a cubic regularization method. In addition, a conceptual trust region method was proposed in [11] to compute an $\epsilon$-approximate $q$-th order stationary point in at most $O(\epsilon^{-q-1})$ iterations. The iteration complexity bounds computed for these methods hide per-iteration complexity of solving the sub-problem. These sub-problems are either quadratic or cubic constrained optimization problems, which are in general NP-complete; see section 4 in this paper.

Concurrent to this work, [31] proposed a general framework that yields convergence to an approximate $(\epsilon_g, \epsilon_H)$-second order stationary point in at most $O(\max\{\epsilon_g^{-2}, \epsilon_H^{-3}\})$ iterations. This is achieved when the feasible set is convex and compact. In particular, the framework uses a first order method to converge to an approximate first order stationary point, and then computes a second order descent direction if it exists. Since solving the quadratic sub-problem to optimality is NP-Hard, they suggest to approximately solve these sub-problems. In this paper, we show that, even for linear constraints, finding an approximate solution for these sub-problems is NP-Hard.

In addition to these second order methods, first order methods have also been used for finding second order stationary points of optimization problems with manifold constraints. The recent work [27] shows that manifold gradient descent converges to local minima under the strict saddle property. More recently, [22] established similar result for a primal-dual optimization procedure for solving linear equality constrained optimization problems. Unlike linear equality constrained scenario, the behavior of first order algorithms is poorly understood in the presence of linear inequality constraints.

In this paper, we first provide an example that shows that projected gradient descent algorithm may con-
verge to strict saddle points with positive probability even in the presence of a single linear constraint. We then discuss an NP-hardness result about solving the sub-problem of current second-order methods applied to constrained optimization problems. Then, inspired by algorithms proposed in [13] and [26], we propose a simple second-order Frank–Wolfe algorithm that uses a dynamic choice for direction and step-size method. Moreover, we show its convergence to \((\epsilon_g, \epsilon_H)\)-second order stationary points in \(O(\max\{\epsilon_g^{-2}, \epsilon_H^{-3}\})\) iterations. Unlike the algorithms proposed in [26, 31], our algorithm does not require any boundedness assumption on the feasible set.

### 2 First and Second Order Stationarity

In order to better understand the first and second order stationary definition in constrained optimization, let us first start by considering the unconstrained optimization problem

\[
\min_{x \in \mathbb{R}^n} f(x),
\]

where \(f : \mathbb{R}^n \mapsto \mathbb{R}\) is a twice continuously differentiable function. We say a point \(\bar{x}\) is a first order stationary point of (1) if \(\nabla f(\bar{x}) = 0\). Similarly, a point \(\bar{x}\) is said to be a second-order stationary point if \(\nabla f(\bar{x}) = 0\) and \(\nabla^2 f(\bar{x}) \succeq 0\). Moreover, if all second order stationary points of the objective function are local optima, we say the function satisfies the \textit{strict saddle} property. This property is satisfied in many practical objective functions; see [18, 25, 38, 39, 40]. In addition, if every local optima of the objective function is globally optimal, then finding the global optimum of the objective boils down to finding a second order stationary point; see [19, 29, 35, 37] for examples of such functions.

Keeping the unconstrained case in mind, let us consider the constrained optimization problem

\[
\min_{x \in \mathcal{F}} f(x),
\]

where \(\mathcal{F} \subseteq \mathbb{R}^n\) is a closed convex set. We say a point \(\bar{x}\) is a first order stationary point of (2) if \(\bar{x} \in \mathcal{F}\) and \(\langle \nabla f(\bar{x}), x - \bar{x} \rangle \geq 0\) for all \(x \in \mathcal{F}\), or equivalently if

\[
0 = \min_s \langle \nabla f(\bar{x}), s \rangle \quad \text{s.t.} \quad \bar{x} + s \in \mathcal{F}, \|s\| \leq 1.
\]

(3)

Similarly, as defined in [4], we say a point \(\bar{x}\) is a second order stationary point of the optimization problem (2) if \(\bar{x} \in \mathcal{F}\) is a first order stationary point and

\[
0 \leq d^T \nabla^2 f(\bar{x}) d, \quad \forall d \text{ s.t. } \langle d, \nabla f(\bar{x}) \rangle = 0 \text{ and } \bar{x} + d \in \mathcal{F}.
\]

(4)

Moreover, we say that (2) satisfies the strict saddle property if every saddle point of the objective is not a second order stationary point. Notice that these definitions simplify to the corresponding unconstrained definitions when \(\mathcal{F} = \mathbb{R}^n\).

Motivated by (3) and (4), given a feasible point \(x\), we define the following first and second order stationarity measures

\[
\mathcal{X}(x) \triangleq \left| \min_s \langle \nabla f(x), s \rangle \right| \quad \text{s.t.} \quad x + s \in \mathcal{F}, \|s\| \leq 1.
\]

(5)

and

\[
\psi(x) \triangleq \left| \min_d d^T \nabla^2 f(x) d \right| \quad \text{s.t.} \quad x + d \in \mathcal{F}, \|d\| \leq 1, \langle \nabla f(x), d \rangle \leq 0.
\]

(6)
The optimality measure (5) has been used before in the literature [12, 6, 10]. However, to the best of our knowledge, the second order stationarity measure (6) has not been utilized before. The next lemma motivates the use of these stationarity measures.

**Lemma 1.** The first and second order stationarity measures $\mathcal{X}(\cdot)$ and $\psi(\cdot)$ are continuous in $x$. Moreover, if $\bar{x} \in \mathcal{F}$ then

- $\mathcal{X}(\bar{x}) = 0$ if and only if $\bar{x}$ is a first order stationary point.
- $\mathcal{X}(\bar{x}) = \psi(\bar{x}) = 0$ if and only if $\bar{x}$ is a second order stationary point.

The above lemma motivates the following definition.

**Definition 2.** Approximate first and second order stationary points:

- Given a positive scalar $\epsilon_g$, a point $\bar{x}$ is said to be an $\epsilon_g$-first order stationary point of the optimization problem (2) if $\bar{x} \in \mathcal{F}$ and $\mathcal{X}(\bar{x}) \leq \epsilon_g$.

- Given positive scalars $\epsilon_g$ and $\epsilon_H$, a point $\bar{x}$ is said to be an $(\epsilon_g, \epsilon_H)$-second order stationary point of the optimization problem (2) if $\bar{x} \in \mathcal{F}$, $\mathcal{X}(\bar{x}) \leq \epsilon_g$ and $\psi(\bar{x}) \leq \epsilon_H$.

Notice that in Definition 2 when the optimization problem (2) is unconstrained, $\epsilon_g$-first order stationarity condition is equivalent to $\|\nabla f(\bar{x})\| \leq \epsilon_g$. Similarly, $(\epsilon_g, \epsilon_H)$-second order stationarity condition is equivalent to $\|\nabla f(\bar{x})\| \leq \epsilon_g$ and $\lambda_{\min}(\nabla^2 f(\bar{x})) \geq -\epsilon_H$. These are the standard definitions of the approximate first and second order stationarity in unconstrained optimization [13, 15, 14, 34, 8].

In the unconstrained scenario, it is well-known that gradient descent with random initialization converges to second order stationary points with probability one [28]. Moreover, there exist various efficient algorithms for finding an $(\epsilon_g, \epsilon_H)$-second order stationary point of the objective function [34, 15, 13, 14, 7, 8].

In what follows, we study whether these results can be directly extended to the constraint scenario by answering the following questions:

**Question 1:** Does projected gradient descent with random initialization converge to second order stationary points with probability one?

**Question 2:** Does there exist an efficient algorithm for finding an $(\epsilon_g, \epsilon_H)$-second order stationary point of the general constrained optimization problem (2)?

### 3 Projected Gradient Descent with Random Initialization May Converge to Strict Saddle Points with Positive Probability

It is known that gradient descent with fixed step size can converge to an $\epsilon$-first order stationary point in $O(\epsilon^{-2})$ iterations for unconstrained smooth optimization problems [33]. Moreover, it escapes strict saddle points of a general smooth unconstrained optimization with probability one when randomly initialized [28]. In the general constrained optimization problem (2), projected gradient descent algorithm is a
natural replacement for gradient descent. The iterates of the projected gradient descent algorithm are obtained by

\[ x_{k+1} \leftarrow \mathcal{P}_F (x_k - \alpha_k \nabla f(x_k)) , \]

where \( \alpha_k \) is the step-size, \( k \) is the iteration number, and \( \mathcal{P}_F \) is the projection operator onto the feasible set \( F \). A natural question about projected gradient descent is whether it has the same behavior as gradient descent algorithm. More specifically, can projected gradient descent escape saddle points under strict saddle property? To answer this question, we provide an example where projected gradient descent fails to converge to second order stationary points even in the presence of a single linear constraint.

Consider the following optimization problem

\[
\min_{x, y \in \mathbb{R}} \ f(x, y) \triangleq -xye^{-x^2-y^2} + \frac{1}{2}y^2 \quad \text{s.t.} \quad x + y \leq 0. \tag{7}
\]

The landscape of the function \( f \) and its corresponding negative gradient mapping are plotted in Figures 1a and 1b. Notice that the function \( f(\cdot) \) has the following first and second order derivatives:

\[
\nabla f(x, y) = \begin{pmatrix}
-(1 - 2x^2)ye^{-x^2-y^2} \\
-(1 - 2y^2)xe^{-x^2-y^2} + y
\end{pmatrix}
\]

\[
\nabla^2 f(x, y) = \begin{pmatrix}
2xy(3 - 2x^2)e^{-x^2-y^2} & -(1 - 2x^2)(1 - 2y^2)e^{-x^2-y^2} \\
-(1 - 2x^2)(1 - 2y^2)e^{-x^2-y^2} & 2xy(3 - 2y^2)e^{-x^2-y^2} + 1
\end{pmatrix}
\]

Figure 1: The landscape and gradient mapping of \( f \). The red box in 1b shows a non-zero measure set that converges to the origin when projected gradient descent is used.

First of all, it is not hard to check that \( \nabla f(0, 0) = 0 \), \( \nabla^2 f(0, 0) = \begin{pmatrix} 0 & -1 \\ -1 & 1 \end{pmatrix} \), and for the feasible direction \( \mathbf{v} = (-1, -1) \), we have \( \mathbf{v}^T \nabla^2 f(0, 0) \mathbf{v} = -1 \). Hence, the point \((0, 0)\) is a saddle point of the objective, while it is not second order stationary. Therefore, the origin is a strict saddle point. However, as one can see in Figure 1b projected gradient descent algorithm may converge to the origin if initialized around the lower right corner of the figure. This observation is true for various step-size selection rules.

To formalize this observation, in what follows, we show that projected gradient descent converges to the strict saddle point \((0, 0)\) if initialized inside the red box in Figure 1b.

First, we show that if the sequence generated by projected gradient descent method intersects a subset of the boundary of the constraint in (7), then the algorithm will eventually converge to the origin.
Lemma 3. If for any \( k \in \mathbb{N} \), the iterate \((x_k, y_k)\) of the sequence generated by projected gradient descent method with constant step-size \( 0 < \bar{\alpha} < 2/3 \) applied to (7) satisfies
\[
x_k \geq 0, \quad y_k = -x_k,
\]
then \( \{x_k, y_k\} \) converges to the origin.

Proof. Proof of this lemma is relegated to Appendix A.

It remains to show that there exists a non-zero measure region so that if we initialize the projected gradient descent algorithm in this region, the iterates converge to a point on the boundary satisfying the conditions in Lemma 3.

Theorem 4. For any given constant step-size \( \alpha_k = \bar{\alpha} \) with \( 0 < \bar{\alpha} < 2/3 \), there exists \( \epsilon > 0 \) so that if we initialize in the set
\[
B_\epsilon \triangleq \{(x, y) \mid 0.5 - \epsilon \leq x \leq 0.5, -0.5 - \epsilon \leq y \leq -0.5\},
\]
then the projected gradient descent method with fixed step-size \( \bar{\alpha} \) converges to the origin when applied to (7).

Proof. Proof of this Theorem is relegated to Appendix B.

This result shows that there is a positive probability that projected gradient descent with random initialization converges to a strict saddle point of the objective. Based on our example, we conjecture that even perturbed/stochastic projected gradient descent algorithm cannot help in escaping strict saddle points. Therefore, a natural question to ask is whether there exists an efficient algorithm for finding an (approximate) second order stationary point. This question is the focus of the next section.

4 Finding or Checking (Approximate) Second Order Stationarity is NP-Hard Even in the Presence of Linear Constraints

Consider the quadratic co-positivity problem
\[
\min_{x \in \mathbb{R}^n} \frac{1}{2} x^T Q x \quad \text{s.t.} \quad x \geq 0.
\] (9)
The classical result of [32] shows that checking whether \( \bar{x} = 0 \) is a local minimum (or equivalently a second order stationary point) of (9) is an NP-hard problem. In particular, [32, Lemma 2] shows that, by adding a ball constraint \( \|x\| \leq 1 \), the optimal objective value of (9) is either 0 or \(-2^{-n}\). Thus, checking whether \( \bar{x} = 0 \) is an \((\epsilon_g, \epsilon_H)\)-second stationary point is not polynomial time solvable if \( \epsilon_H \) is small. In this section, we show that even a less ambitious goal is NP-hard. More precisely, we show that checking \((\epsilon_g, \epsilon_H)\)-second stationarity is NP-hard in \((n, 1/\epsilon_H)\).

Before proceeding to the result, let us define some notations. Let \( G(V, E) \) be a graph with the set of vertices \( V \) and the set of edges \( E \). Also let \( |V| \) be the cardinality of \( V \) and \( A_G \) be the adjacency matrix of graph \( G \). We define \( C_n \triangleq \{Q \in \mathbb{R}^{n \times n} \mid x^T Q x \geq 0, \forall x \geq 0\} \) to be the set of co-positive matrices. We denote the identity matrix and the all-one matrix of size \( n \) by \( I_n \) and \( 1_n \) respectively. We say graph \( G \) has a stable set of size \( t \) if it contains a subset of \( t \) vertices, from which no two vertices in the subset are connected by an edge.

Lemma 5. Let \( G = (V, E) \) be a graph with \( |V| = n \). Given a scalar \( t \) with \( t \leq n \), define
\[
Q = (I_n + A_G)(t - \frac{1}{2}) - 1_n, \quad \text{and} \quad \delta = \frac{1}{2n+1}.
\]
Then the following are equivalent:
i. \( \min_{x \geq 0, \|x\| \leq 1} x^T Q x \leq -\frac{\delta}{\sqrt{n}} \).

ii. \( G \) contains a stable set of size \( t \)

**Proof.** We first show that i implies ii. By the definition of the set \( C_n \), the condition

\[
\min_{\|x\| \leq 1, x \geq 0} x^T Q x \leq -\frac{\delta}{\sqrt{n}}
\]

implies that \( Q \notin C_n \). Therefore, by [16, Lemma 4.1], \( G \) contains a stable set of size \( t \).

To show the converse, we use [16, lemma 4.5]. Suppose that \( G \) contains a stable set of size \( t \). By [16, lemma 4.1], \( Q \notin C_n \). Moreover, by [16, lemma 4.5] it is \( \delta \) far away from \( C_n \). In other words,

\[
\|Y - Q\|_F > \delta, \quad \forall Y \in C_n.
\]  

Let \( \bar{x} \in \arg \min_{x \geq 0, \|x\| \leq 1} x^T Q x \). Define \( \tilde{Q} = Q + \frac{\delta}{\sqrt{n}} I_n \). Clearly, \( \tilde{Q} \notin C_n \) due to (10) and \( \bar{x} \in \arg \min_{x \geq 0, \|x\| \leq 1} x^T \tilde{Q} x \). Then,

\[
\min_{x \geq 0, \|x\| \leq 1} x^T Q x = \bar{x}^T \tilde{Q} \bar{x} = \bar{x}^T Q \bar{x} - \frac{\delta}{\sqrt{n}} \|\bar{x}\|^2
\]

\[
= \min_{x \geq 0, \|x\| \leq 1} x^T \tilde{Q} x - \frac{\delta}{\sqrt{n}} \|x\|^2
\]

\[
\leq -\frac{\delta}{\sqrt{n}}
\]

where the last inequality is due to the fact that \( \tilde{Q} \notin C_n \).

The result of the above lemma directly implies the following theorem about the hardness of checking second order stationarity.

**Theorem 6.** For the co-positivity problem

\[
\min_{x \in \mathbb{R}^n} \frac{1}{2} x^T Q x \quad \text{s.t.} \quad x \geq 0,
\]  

there is no algorithm which can check whether \( x = 0 \) is an \((\epsilon_g, \epsilon_H)\)-second order stationary point in polynomial time in \((n, \frac{1}{\epsilon_H})\), unless \( P = NP \).

**Proof.** The result is an immediate consequence of Lemma [5]
5 Easy Instances of Finding Second Order Stationarity in Constrained Optimization: A Second Order Frank–Wolfe Algorithm

As discussed in previous sections, although designing polynomial time algorithms for finding second order stationary points is easy when the optimization problem is unconstrained, the same problem becomes very hard in the general convex constrained case. In particular, even for checking second order stationarity, one needs to (approximately) solve a quadratic constrained optimization problem (6), which is NP-hard as shown in Section 4. However, for some special types of constraint set \( \mathcal{F} \), the quadratic constrained optimization problem (6) can be solved efficiently. For example, when \( \mathcal{F} \) is formed by small number of linear constraints, [23] presents a backtracking approach which can find the solution of (6) efficiently. More precisely, by doing an exhaustive backtracking search on the set of constraints, one can find the solution of the problem

\[
\begin{align*}
\min_d & \quad d^T Qd \\
\text{s.t.} & \quad x + d \in \mathcal{F}, \|d\| \leq 1 \\
& \quad \langle \nabla f(x), d \rangle \leq 0.
\end{align*}
\]

(12)

efficiently when \( \mathcal{F} = \{x \mid a_i^T x \leq b_i, \text{ for } i = 1, \ldots, m\} \) assuming that \( m \) is small and one can afford a search which is exponentially large in \( m \).

Assuming that (12) can be solved efficiently for a given \( \mathcal{F} \), a natural question to ask is as follows:

Assume that the constraint set \( \mathcal{F} \) is such that the quadratic optimization problem (12) can be solved efficiently. For such a constraint set \( \mathcal{F} \), can we find an \((\epsilon_g, \epsilon_H)\)-second order stationary point of the general smooth optimization problem (2) efficiently?

In this section, we answer this question affirmatively by proposing a polynomial time algorithm for finding \((\epsilon_g, \epsilon_H)\) second order stationary point of problem (2) assuming that a quadratic optimization problem of the form (12) can be solved efficiently at each iteration. The proposed algorithm can be viewed as a simple second order generalization of the Frank–Wolfe algorithm proposed in [26]. In particular, in addition to the first order Frank–Wolfe direction computed by solving (5) at \( x_k \), we also compute a second-order descent direction by solving (6) at each iteration. Then we dynamically choose the direction that potentially offers more reduction in the objective value. This dynamic method was used in [13] to design an algorithm for unconstrained optimization problems. They show convergence to an \((\epsilon_g, \epsilon_H)\)-second-order stationary points with complexity \( O(\max\{\epsilon_g^{-2}, \epsilon_H^{-3}\}) \). Our proposed algorithm adapts this method to the constrained scenario while maintaining the same convergence guarantees and complexity bounds.

Notations. Given a sequence of iterates \( \{x_k\} \) computed by an algorithm for solving (2), we define \( \mathcal{X}_k \triangleq \mathcal{X}(x_k) \) and \( \psi_k \triangleq \psi(x_k) \), where \( \mathcal{X}(\cdot) \) and \( \psi(\cdot) \) functions are defined in (5) and (6).

Throughout this section, we make the following assumption.

Assumption 7. The objective function \( f \) is twice continuously differentiable and bounded below by a scalar \( f_{\min} \) on \( \mathcal{F} \). The constraint set \( \mathcal{F} \) is closed and convex. We assume that functions \( \nabla f(\cdot) \) and \( \nabla^2 f(\cdot) \) are Lipschitz continuous on the path defined by the iterates computed in algorithm 1 with Lipschitz constants \( L \) and \( \rho \), respectively. Furthermore, the gradient sequence \( \{\nabla f(x_k)\} \) is bounded such that there
exists a scalar constant \( g_{\max} \in \mathbb{R}_{++} \) such that \( \| \nabla f(x_k) \|_2 \leq g_{\max} \) for all \( k \in \mathbb{N} \). Moreover, we assume that the Hessian sequence \( \{ \nabla^2 f(x_k) \} \) is bounded in norm, that is, there exist a scalar constant \( H_{\max} \in \mathbb{R}_{++} \) such that \( \| \nabla^2 f(x_k) \|_2 \leq H_{\max} \) for all \( k \in \mathbb{N} \).

### 5.1 Algorithm Description

Let \( x_k \) be the iterate in our algorithm at iteration \( k \). Given point \( x_k \), we define the following first order and second order descent directions

\[
\hat{s}_k \triangleq \arg \min_s \langle \nabla f(x_k), s \rangle \\
\text{s.t. } x_k + s \in \mathcal{F}, \|s\| \leq 1.
\]

and

\[
\hat{d}_k \triangleq \arg \min_d \quad d^T \nabla^2 f(x_k) d \\
\text{s.t. } x_k + d \in \mathcal{F}, \|d\| \leq 1 \\
\langle \nabla f(x_k), d \rangle \leq 0.
\]

Notice that in the unconstrained scenario, \( \hat{s}_k = -\nabla f(x_k) \) and \( \hat{d}_k \) is the eigenvector corresponding to the smallest eigenvalue of the Hessian matrix \( \nabla^2 f(x_k) \), which lead to the simple directions proposed in [13] for unconstrained scenario.

The algorithm described below follows a dynamic strategy of choosing between \( \hat{s}_k \) and \( \hat{d}_k \) for all \( k \in \mathbb{N} \). The choice is done based on which direction predicts a larger reduction in the objective. If \( \hat{s}_k \) is always chosen, then the algorithm resembles Frank–Wolfe algorithm [26]. Hence, our algorithm can be seen as a second order extension of Frank–Wolfe algorithm.

---

**Algorithm 1 Second Order Frank–Wolfe with Fixed Step-size**

**Require:** The constants \( \tilde{L} \triangleq \max \{ L, g_{\max} \} \), \( \tilde{\rho} \triangleq \max \{ \rho, H_{\max} \} \).

1: procedure
2: Choose \( x_0 \in \mathcal{F} \).
3: Compute \( X_0 \) and \( \psi_0 \) by solving (5) and (6), respectively.
4: for \( k = 0, 1, 2, \ldots \) do
5: if \( X_k = 0 \) then set \( \hat{s}_k = 0 \), else compute \( \hat{s}_k \) and \( X_k \) by solving (13).
6: if \( \psi_k = 0 \) then set \( \hat{d}_k = 0 \), else compute \( \hat{d}_k \) and \( \psi_k \) by solving (14).
7: if \( \psi_k = X_k = 0 \) then
8: terminate and return \( x_k \).
9: end if
10: if \( \frac{X_k^2}{2L} \geq \frac{2\psi_k^3}{3\tilde{\rho}^2} \) then
11: set \( x_{k+1} \leftarrow x_k + \frac{X_k}{L} \hat{s}_k \)
12: else
13: set \( x_{k+1} \leftarrow x_k + \frac{2\psi_k}{\tilde{\rho}} \hat{d}_k \)
14: end if
15: end for
16: end procedure
5.2 Convergence Results

We first note that regardless of the direction we choose, the step size is either $X_k \frac{\tilde{L}}{L}$ or $2\psi_k \tilde{\rho}$, which are both less than or equal to 1. Thus, the iterates generated by the algorithm are always feasible. Also notice that, unlike the algorithms proposed in \cite{26, 31}, our algorithm does not require any boundedness assumption on the feasible set $F$. Another advantage of the proposed algorithm is that it does not require the knowledge of the desired accuracy level ($\epsilon_g, \epsilon_H$). This allows us to modify our termination rule when running the algorithm if needed.

Next, we show that Algorithm 1 asymptotically converges to a second order stationary point.

**Theorem 8.** Under Assumption 7, $$\lim_{k \to \infty} X_k = \lim_{k \to \infty} \psi_k = 0.$$ In other words, any limit point of the iterates is a second order stationary point.

**Proof.** We first show the following reduction bound in the objective value

$$f(x_k) - f(x_{k+1}) \geq \max \left\{ \frac{X_k^2}{2}, \frac{2\psi_k^3}{3\tilde{\rho}^2} \right\}.$$  \hspace{1cm} (15)

First of all, notice that if Step 8 is reached, then clearly the reduction bound is satisfied. Otherwise, $x_{k+1}$ is set in Step 11 or Step 13.

If Step 11 is reached, then using descent lemma \cite{31 Appendix A.24}, we obtain

$$f(x_{k+1}) \leq f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k \rangle + \frac{1}{2} \|x_{k+1} - x_k\|^2$$

$$\leq f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k \rangle + \frac{L}{2} \|x_{k+1} - x_k\|^2$$

$$\leq f(x_k) + \frac{X_k^2}{L} \langle \nabla f(x_k), \hat{s}_k \rangle + \frac{X_k^2}{2L} \|\hat{s}_k\|^2$$

$$\leq f(x_k) - \frac{X_k^2}{2L} + \frac{X_k^2}{2L}$$

$$= f(x_k) - \frac{X_k^2}{2L}. \hspace{1cm} (15)$$

Otherwise, if Step 13 is reached, then using second-order descent lemma, we obtain

$$f(x_{k+1}) \leq f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k \rangle$$

$$+ \frac{1}{2} \langle x_{k+1} - x_k \rangle^T \nabla^2 f(x_k)(x_{k+1} - x_k) + \frac{\tilde{\rho}}{6} \|x_{k+1} - x_k\|^3$$

$$\leq f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k \rangle$$

$$+ \frac{1}{2} \langle x_{k+1} - x_k \rangle^T \nabla^2 f(x_k)(x_{k+1} - x_k) + \frac{\tilde{\rho}}{6} \|x_{k+1} - x_k\|^3$$

$$\leq f(x_k) + \frac{2\psi_k}{\tilde{\rho}} \langle \nabla f(x_k), \hat{d}_k \rangle + \frac{2\psi_k^2}{\tilde{\rho}^2} \langle \hat{d}_k \rangle^T \nabla^2 f(x_k)(\hat{d}_k) + \frac{4\psi_k^3}{3\tilde{\rho}^2} \|\hat{d}_k\|^2$$

$$\leq f(x_k) - \frac{2\psi_k^3}{\tilde{\rho}^2} + \frac{4\psi_k^3}{3\tilde{\rho}^2}$$

$$= f(x_k) - \frac{2\psi_k^3}{3\tilde{\rho}^2}. \hspace{1cm} (16)$$
where the fourth inequality holds since \( \langle \nabla f(x_k), \hat{d}_k \rangle \leq 0 \), \( \| \hat{d}_k \| \leq 1 \), and \( \psi_k = - (\hat{d}_k)^T \nabla^2 f(x_k) \hat{d}_k \).

Combining (15) and (16) with Step 10, we obtain the following reduction in the objective value

\[
f(x_{k+1}) \leq f(x_k) - \max \left\{ \frac{\mathcal{X}_k^2}{2L}, \frac{2\psi_k^3}{3\rho^2} \right\}. \tag{17}\]

By summing over the iterations, we get

\[
f(x_{\ell+1}) - f(x_0) = \sum_{k=0}^{\ell} (f(x_{k+1}) - f(x_k)) \leq - \sum_{k=0}^{\ell} \max \left\{ \frac{\mathcal{X}_k^2}{2L}, \frac{2\psi_k^3}{3\rho^2} \right\}. \tag{18}\]

Hence, since \( f \) is bounded below by \( f_{\text{min}} \), we have

\[
0 \leq \sum_{k=0}^{\ell} \max \left\{ \frac{\mathcal{X}_k^2}{2L}, \frac{2\psi_k^3}{3\rho^2} \right\} \leq f(x_0) - f_{\text{min}}. \]

Thus,

\[
\lim_{k \to \infty} \mathcal{X}_k = \lim_{k \to \infty} \psi_k = 0.
\]

Moreover, the continuity of the functions \( \mathcal{X}(\cdot) \) and \( \psi(\cdot) \) implies that every limit point of the iterates is a second order stationary point.

The next result computes the worst-case complexity required to reach an \( \epsilon_g \)-first order stationary point and to reach an \( (\epsilon_g, \epsilon_H) \)-second order stationary point.

**Theorem 9.** Let \( \epsilon_g, \epsilon_H > 0 \). The number of iterations required for Algorithm 1 to find an \( \epsilon_g \)-first order stationary point is at most

\[
\frac{2\tilde{L}(f(x_0) - f_{\text{min}})}{\epsilon_g^2}.
\]

Moreover, the number of iterations required to find an \( (\epsilon_g, \epsilon_H) \)-second order stationary point is at most

\[
\frac{f(x_0) - f_{\text{min}}}{\min \left\{ \frac{\epsilon_g^2}{2L}, \frac{2\epsilon_H^3}{3\rho^2} \right\}}.
\]

**Proof.** First notice that the sufficient decrease bound \((18)\) implies that

\[
\sum_{k=0}^{\ell} \max \left\{ \frac{\mathcal{X}_k^2}{2L}, \frac{2\psi_k^3}{3\rho^2} \right\} \leq f(x_0) - f_{\text{min}}, \tag{19}\]

for every iteration \( \ell \). Define the index sets

\[
\mathcal{G}(\epsilon_g) \triangleq \{ k \mid \mathcal{X}_k > \epsilon_g \} \quad \text{and} \quad \mathcal{H}(\epsilon_H) \triangleq \{ k \mid \psi_k > \epsilon_H \}.
\]

According to the bound \((19)\), it is easy to show that the cardinality of the above two sets is bounded by

\[
| \mathcal{G}(\epsilon_g) | \leq \frac{2\tilde{L}(f(x_0) - f_{\text{min}})}{\epsilon_g^2},
\]

\[
| \mathcal{G}(\epsilon_g) \cup \mathcal{H}(\epsilon_H) | \leq \frac{f(x_0) - f_{\text{min}}}{\min \left\{ \frac{\epsilon_g^2}{2L}, \frac{2\epsilon_H^3}{3\rho^2} \right\}}. \tag{20}\]

\( \square \)
The complexity order of the proposed algorithm is the same as the algorithm proposed in [13] [31]. In particular, the complexity of finding an $\epsilon_g$-first-order stationary point is $O(\epsilon_g^{-2})$, which is not order optimal under our assumptions (at least for unconstrained scenario). Second order information in conjunction with smoothness of the Hessian has been used in unconstrained scenario before to design algorithms with better complexity orders for reaching first order stationarity [6] [10] [11].
References

[1] A. Anandkumar and R. Ge. Efficient approaches for escaping higher order saddle points in non-convex optimization. In Conference on Learning Theory, pages 81–102, 2016.

[2] A. S. Bandeira, N. Boumal, and V. Voroninski. On the low-rank approach for semidefinite programs arising in synchronization and community detection. In Conference on Learning Theory, pages 361–382, 2016.

[3] B. Barazandeh and M. Razaviyayn. On the behavior of the expectation-maximization algorithm for mixture models. arXiv preprint arXiv:1809.08705, 2018.

[4] D. P. Bertsekas. Nonlinear programming. Athena scientific Belmont, 1999.

[5] N. Boumal. Nonconvex phase synchronization. SIAM Journal on Optimization, 26(4):2355–2377, 2016.

[6] C. Cartis, N. Gould, and P. L. Toint. An adaptive cubic regularization algorithm for nonconvex optimization with convex constraints and its function-evaluation complexity. IMA Journal of Numerical Analysis, 32(4):1662–1695, 2012.

[7] C. Cartis, N. I. Gould, and P. L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. part i: motivation, convergence and numerical results. Mathematical Programming, 127(2):245–295, 2011.

[8] C. Cartis, N. I. Gould, and P. L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. part ii: worst-case function-and derivative-evaluation complexity. Mathematical programming, 130(2):295–319, 2011.

[9] C. Cartis, N. I. Gould, and P. L. Toint. On the evaluation complexity of cubic regularization methods for potentially rank-deficient nonlinear least-squares problems and its relevance to constrained nonlinear optimization. SIAM Journal on Optimization, 23(3):1553–1574, 2013.

[10] C. Cartis, N. I. Gould, and P. L. Toint. On the evaluation complexity of constrained nonlinear least-squares and general constrained nonlinear optimization using second-order methods. SIAM Journal on Numerical Analysis, 53(2):836–851, 2015.

[11] C. Cartis, N. I. Gould, and P. L. Toint. Second-order optimality and beyond: Characterization and evaluation complexity in convexly constrained nonlinear optimization. Foundations of Computational Mathematics, pages 1–35, 2017.

[12] A. R. Conn, N. I. Gould, and P. L. Toint. Trust region methods, volume 1. Siam, 2000.

[13] F. E. Curtis and D. P. Robinson. Exploiting negative curvature in deterministic and stochastic optimization. arXiv preprint arXiv:1703.00412, 2017.

[14] F. E. Curtis, D. P. Robinson, and M. Samadi. An inexact regularized newton framework with a worst-case iteration complexity of $o(\epsilon^{-3/2})$ for nonconvex optimization. arXiv preprint arXiv:1708.00475, 2017.

[15] F. E. Curtis, D. P. Robinson, and M. Samadi. A trust region algorithm with a worst-case iteration complexity of $o(\epsilon^{-5/2})$ for nonconvex optimization. Mathematical Programming, 162(1-2):1–32, 2017.

[16] P. J. Dickinson and L. Gijben. On the computational complexity of membership problems for the completely positive cone and its dual. Computational optimization and applications, 57(2):403–415, 2014.
S. S. Du, C. Jin, J. D. Lee, M. I. Jordan, A. Singh, and B. Poczos. Gradient descent can take exponential time to escape saddle points. In Advances in Neural Information Processing Systems, pages 1067–1077, 2017.

R. Ge, F. Huang, C. Jin, and Y. Yuan. Escaping from saddle points: online stochastic gradient for tensor decomposition. In Conference on Learning Theory, pages 797–842, 2015.

R. Ge, J. D. Lee, and T. Ma. Matrix completion has no spurious local minimum. In Advances in Neural Information Processing Systems, pages 2973–2981, 2016.

S. Ghadimi, G. Lan, and H. Zhang. Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization. Mathematical Programming, 155(1-2):267–305, 2016.

A. Griewank. The modification of newtons method for unconstrained optimization by bounding cubic terms. Technical report, Technical report NA/12, 1981.

M. Hong, J. D. Lee, and M. Razaviyayn. Gradient primal-dual algorithm converges to second-order stationary solutions for nonconvex distributed optimization. arXiv preprint arXiv:1802.08941, 2018.

Y. Hsia and R.-L. Sheu. Trust region subproblem with a fixed number of additional linear inequality constraints has polynomial complexity. arXiv preprint arXiv:1312.1398, 2013.

C. Jin, R. Ge, P. Netrapalli, S. M. Kakade, and M. I. Jordan. How to escape saddle points efficiently. arXiv preprint arXiv:1703.00887, 2017.

K. Kawaguchi. Deep learning without poor local minima. In Advances in Neural Information Processing Systems, pages 586–594, 2016.

S. Lacoste-Julien. Convergence rate of frank-wolfe for non-convex objectives. arXiv preprint arXiv:1607.00345, 2016.

J. D. Lee, I. Panageas, G. Piliouras, M. Simchowitz, M. I. Jordan, and B. Recht. First-order methods almost always avoid saddle points. arXiv preprint arXiv:1710.07406, 2017.

J. D. Lee, M. Simchowitz, M. I. Jordan, and B. Recht. Gradient descent converges to minimizers. arXiv preprint arXiv:1602.04915, 2016.

H. Lu and K. Kawaguchi. Depth creates no bad local minima. arXiv preprint arXiv:1702.08580, 2017.

S. Lu, Z. Wei, and L. Li. A trust region algorithm with adaptive cubic regularization methods for nonsmooth convex minimization. Computational Optimization and Applications, 51(2):551–573, 2012.

A. Mokhtari, A. Ozdaglar, and A. Jadbabaie. Escaping saddle points in constrained optimization. arXiv preprint arXiv:1809.02162, 2018.

K. G. Murty and S. N. Kabadi. Some np-complete problems in quadratic and nonlinear programming. Mathematical programming, 39(2):117–129, 1987.

Y. Nesterov. Introductory lectures on convex optimization: A basic course, volume 87. Springer Science & Business Media, 2013.

Y. Nesterov and B. T. Polyak. Cubic regularization of newton method and its global performance. Mathematical Programming, 108(1):177–205, 2006.
For a given $k \in \mathbb{N}_+$, let $(x_k, y_k) = (x_k, -x_k)$ be the current iterate with $x_k \geq 0$. We first show that iterate $k + 1$ generated by projected gradient descent satisfies

$$x_{k+1} + y_{k+1} = 0.$$  \hfill (21)

Then we show that

$$x_{k+1} \geq 0, \quad x_k - x_{k+1} \geq 0.$$  \hfill (22)

Combining (21) and (22), we will complete our proof.

First note that if $x_k = 0$, then the result trivially holds. Assume that $x_k > 0$, we define

$$\bar{x}_{k+1} \triangleq x_k - \bar{\alpha} \nabla_x f(x_k, -x_k) = x_k - \bar{\alpha}(1 - 2x_k^2)x_k e^{-2x_k^2} \quad \text{and}$$

$$\bar{y}_{k+1} \triangleq y_k - \bar{\alpha} \nabla_y f(x_k, -x_k) = -x_k + \bar{\alpha}[(1 - 2x_k^2)x_k e^{-2x_k^2} + x_k].$$

Since $\bar{x}_{k+1} + \bar{y}_{k+1} = \bar{\alpha}x_k > 0$, the point $(\bar{x}_{k+1}, \bar{y}_{k+1})$ is not feasible. By projecting $(\bar{x}_{k+1}, \bar{y}_{k+1})$ to the feasible set $\{(x, y) \mid y + x \leq 0\}$, we obtain

$$x_{k+1} = x_k - \bar{\alpha}(1 - 2x_k^2)x_k e^{-2x_k^2} - \frac{\bar{\alpha}}{2} x_k \quad \text{and}$$

$$y_{k+1} = -x_k + \bar{\alpha}[(1 - 2x_k^2)x_k e^{-2x_k^2} + \frac{1}{2} x_k].$$  \hfill (23)

Obviously (21) holds. We now show that

$$x_{k+1} = x_k \left[1 - \frac{\bar{\alpha}}{2} - \bar{\alpha}(1 - 2x_k^2)e^{-2x_k^2}\right] \geq 0 \quad \text{and}$$

$$x_k - x_{k+1} = x_k \left[\bar{\alpha}(1 - 2x_k^2)e^{-2x_k^2} + \frac{\bar{\alpha}}{2}\right] \geq 0.$$  \hfill (24)

Let $g(x) \triangleq (1 - 2x^2)e^{-2x^2}$. This function has two global minima $x = \pm 1$, and one global maximum $x = 0$. Hence,

$$-e^{-2} \leq g(x) \leq 1, \quad \forall \ x.$$
Using (24), we get
\[ x_{k+1} = x_k \left[1 - \frac{\bar{\alpha}}{2} - \bar{\alpha}g(x_k)\right] \geq x_k \left[1 - \frac{\alpha}{2} - \bar{\alpha}\right] \geq 0, \]
where the second inequality holds since \( \bar{\alpha} < 2/3 \) and \( x_k \geq 0 \). Also,
\[ x_k - x_{k+1} = x_k \bar{\alpha} \left[g(x_k) + \frac{1}{2}\right] \geq x_k \bar{\alpha} \left[\frac{1}{2} - e^{-2}\right] \geq 0. \]
Combining (21) and (22), we conclude that for any \( \bar{x} \)
\[ \sup_{x} \bar{\alpha} < 2 \]
\[ \bar{x} \]
\[ \lim_{k \to \infty} x_k = \bar{x} \]
which implies
\[ \bar{x} = 0, \quad \text{or} \quad g(\bar{x}) = -\frac{1}{2}. \]
Since \( \max_{x} g(x) > -e^{-2} \), we get that \( \bar{x} = 0 \) which completes the proof.

\section*{B Proof of Theorem 4}
Consider the initial point \((x_0, y_0)\). If we can show that \( y_1 = -x_1 \) and \( x_1 \geq 0 \), then using Lemma 3, we conclude that the sequence of iterates \( \{x_k, y_k\} \) eventually converges to the origin. Thus it suffices to show that there exist an \( \epsilon > 0 \) such that if
\[ 0.5 - \epsilon \leq x_0 \leq 0.5, \quad \text{and} \quad -0.5 - \epsilon \leq y_0 \leq -0.5, \]
then the next iterate \((x_1, y_1)\) satisfies
\[ x_1 = x_0 + \bar{\alpha}y_0(1 - 2x_0^2)e^{-x_0^2 - y_0^2} \geq 0, \]
\[ y_1 + x_1 = y_0 + \bar{\alpha}[x_0(1 - 2y_0^2)e^{-x_0^2 - y_0^2} - y_0] + x_0 + \bar{\alpha}y_0(1 - 2x_0^2)e^{-x_0^2 - y_0^2} \geq 0. \]
The first condition (26a) is satisfied when the step-size \( \bar{\alpha} \) is small enough. To prove (26b) we utilize the conditions in (25) to obtain the following inequalities
\[ -2\epsilon \leq x_0 + y_0 \leq 0, \]
\[ 0.25 + (0.5 - \epsilon)^2 \leq x_0^2 + y_0^2 \leq (0.5 + \epsilon)^2 + 0.25, \]
\[ 0.5 \leq 1 - 2x_0^2 \leq 1 - 2(0.5 - \epsilon)^2, \]
\[ 1 - 2(0.5 + \epsilon)^2 \leq 1 - 2y_0^2 \leq 0.5, \]
which implies that
\[ x_0(1 - 2y_0^2) + y_0(1 - 2x_0^2) \geq (0.5 - \epsilon)(1 - 2(0.5 + \epsilon)^2) - (0.5 + \epsilon)(1 - 2(0.5 - \epsilon)^2) \]
\[ = -3\epsilon + 4\epsilon^3. \]
Then using (27), we get
\[ x_0 + y_0 - \bar{\alpha}y_0 + \bar{\alpha} \left[x_0(1 - 2y_0^2) + y_0(1 - 2x_0^2)\right] e^{-x_0^2 - y_0^2} \geq -2\epsilon + 0.5\bar{\alpha} + \bar{\alpha}(-3\epsilon + 4\epsilon^3)e^{-0.25 + (0.5+\epsilon)^2}. \]
Note that the right hand side is \( 0.5\bar{\alpha} + O(\epsilon) \) which is greater than or equal to zero for sufficiently small \( \epsilon \). This shows that condition (26b) holds, and the proof follows by Lemma 3.