# Finding Second-Order Stationary Points in Constrained Minimization: A Feasible Direction Approach

Nadav Hallak* Marc Teboulle†

**Abstract**

This paper introduces a method for computing points satisfying the second-order necessary optimality conditions in constrained nonconvex minimization. The method comprises two independent steps corresponding to the first and second order conditions. The first-order step is a generic closed map algorithm which can be chosen from a variety of first-order algorithms, making it adjustable to the given problem. The second-order step can be viewed as a second-order feasible direction step for constrained nonconvex minimization. We prove that any limit point of the resulting scheme satisfies the second-order necessary optimality condition, and establish the scheme's convergence rate and complexity, under standard and mild assumptions. Numerical tests validate our theoretical results, and illustrate how and when the proposed method can be efficiently implemented.

**Keywords:** feasible direction methods, second-order necessary optimality conditions, constrained optimization, convergence analysis.

## 1 Introduction

This paper presents a method to obtain solutions that satisfy the classical second-order necessary optimality conditions for the nonconvex constrained problem:

$$\inf\{f(\mathbf{x}) \ : \ \mathbf{x} \in C\}, \tag{P}$$

where:

- $f : \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable.
- The set $C \subseteq \mathbb{R}^n$ is nonempty, closed, and convex.

*School of Mathematical Sciences, Tel-Aviv University, Ramat-Aviv 69978, Israel. E-mail: nadav_hallak@outlook.com. This research is supported by a postdoctoral fellowship under ISF grant 1844-16

†School of Mathematical Sciences, Tel-Aviv University, Ramat-Aviv 69978, Israel. E-mail: teboulle@tauex.tau.ac.il. This research was partially supported by the Israel Science Foundation, under ISF Grant 1844-16

- $f_* := \inf\{f(\mathbf{x}) \ : \ \mathbf{x} \in C\} > -\infty$.

The literature on second-order methods, which we regard as methods that use the Hessian of the objective function, can roughly be categorized into three prominent frameworks: curvilinear, trust-region, and cubic regularization. For background and references on unconstrained second-order methods, in particular the trust-region approach, we refer the reader to the comprehensive book [10], and the more recent review [26, Section 2]; to [1, Section 1] where the unconstrained curvilinear approach is well-summarized, and to the literature review in [7, Section 1] given as an introductory survey for a cubic regularization method aimed to solved unconstrained problems. Computing second-order stationarity points in nonconvex problems is not an easy task even in the unconstrained setting, as second-order methods usually require a combination of several assumptions, see for example the curvilinear method in [1], the assumptions on the trust-region method given in [10, Chapter II, Sections 6.2 and 6.3], or the requirements for the cubic regularization method [22].

While the literature on methods converging to second-order stationary points in the unconstrained setting has been well-studied in the above alluded references, the research activities and convergence results for the *constrained* setting, more specifically for minimization of nonconvex smooth functions over general convex constraints, are much more limited and significantly more complicated. Curvilinear methods [11, 13, 15, 14, 21, 24], trust-region methods [10, Part III] and [9], and cubic regularization [8], become more difficult to analyze when compared to their unconstrained versions. These methods (see for instance the recent works [8, 9]) require much more elaborate, often restrictive or difficult to verify, assumptions, to ensure some viable convergence properties, as well as significant computational demands involving hard problems that must be solved at every iteration.

The main goal of this paper is to provide a novel alternative for finding second-order stationary points in constrained minimization. We develop a simple and versatile method with convergence guarantees, under a minimal set of assumptions. The method we propose bares some resemblance to the three prominent frameworks alluded above, but the core idea is conceptually much more similar to the very classical feasible directions framework (with an exact line-search procedure), which goes back to [27]. We call it the *Two-Directions (TD)* method, as it updates the current iterate according to the best of two major steps corresponding to the first and second-order necessary optimality conditions.

The first step computes a first-order update direction using a general *closed map algorithm* [27, Chapter 4] that guarantees that any accumulation point satisfies the first-order necessary optimality condition. This generic choice allows for making the method versatile, as it enables it to use any classical first-order algorithm as its first-order step. Resultantly, first-order algorithms tailored for specific instances of the problem can be invoked in order to hasten or improve the method's performance. The other and second step computes the second-order update direction by minimizing the second-order directional derivative, which is a quadratic function associated to the Hessian, over a compact set of feasible directions; this step can be regarded as a second-order feasible direction procedure.

In the special unconstrained case, computing the second-order update direction in the

proposed TD method essentially reduces to finding the eigenvector corresponding to the smallest eigenvalue of the Hessian, followed by a simple line search.In the constrained scenario, the second order direction is the most computationally demanding requirement in the TD method, as the problem to be solved may not be convex, and in general results in a hard optimization problem. Yet, recent advances in the field of quadratic optimization suggest that it is computationally tractable in various interesting cases, depending on the geometry of the set of constraints involved, see for instance [19], and the more recent work [4], which also contains a concise review on the topic. This together with the implementation issues will be discussed in more depth in the numerical illustrations section.

**Paper Outline.** In Section 2 we introduce the terminology and basic elements to be used throughout the study. The *Two-Directions (TD)* method is presented in Section 3, and its convergence properties are analyzed in Section 4. We conclude with a discussion on the implementation aspects of the TD method, as well as numerical illustrations, in Section 5.

# 2 Preliminaries

This section introduces the terminology and fundamental definitions used throughout the paper, starting with the notion of feasible directions. Recall that for an arbitrary nonempty subset $C$ of $\mathbb{R}^n$, a nonzero $d \in \mathbb{R}^n$ is a feasible direction at $\mathbf{x} \in C$ if there is an $\varepsilon > 0$ such that $\mathbf{x} + t\mathbf{d} \in C$ for all $t \in [0, \varepsilon]$, see e.g., [2, 27]. Here, we are working with $C$ being closed and convex, and a direction $\mathbf{d} \in \mathbb{R}^n$ is called a *feasible direction* at $\mathbf{x} \in C$ if $\mathbf{x} + \mathbf{d} \in C$. The set of such feasible directions at a point $\mathbf{x} \in C$ will be denoted by

$$\mathcal{D}_{\mathbf{x}} = \{\mathbf{d} \in \mathbb{R}^n : \mathbf{x} + \mathbf{d} \in C\}.$$

For convenience, we use the standard notation for the directional derivatives of $f$ with respect to $\mathbf{d} \in \mathbb{R}^n$: $f'(\mathbf{x}; \mathbf{d}) = \langle \mathbf{d}, \nabla f(\mathbf{x}) \rangle$ and $f''(\mathbf{x}; \mathbf{d}) = \langle \mathbf{d}, \nabla^2 f(\mathbf{x})\mathbf{d} \rangle$. The classical necessary optimality conditions associated with the above directional derivatives and feasible directions, are as follows.

**Lemma 2.1** (First order optimality condition [6, Prop. 2.1.2]). *Let $\mathbf{x}^* \in C$ be a local minimizer of $(P)$. Then:*

$$f'(\mathbf{x}^*; \mathbf{d}) \geq 0 \qquad \forall \mathbf{d} \in \mathcal{D}_{\mathbf{x}^*}.$$

**Lemma 2.2** (Second order optimality condition [6, Ex. 2.1.10]). *Let $\mathbf{x}^* \in C$ be a local minimizer of $(P)$. Then:*

*(i) $\mathbf{x}^*$ satisfies the first-order optimality condition, and*

*(ii) for any $\mathbf{d} \in \mathcal{D}_{\mathbf{x}^*}$ the following implication holds true*

$$f'(\mathbf{x}^*; \mathbf{d}) = 0 \;\Rightarrow\; f''(\mathbf{x}^*; \mathbf{d}) \geq 0.$$

We will call points that satisfy the first/second-order optimality conditions (FO)/(SO) points. Clearly, we have that (SO) implies (FO), but not vice versa.

A straightforward observation is that the above optimality conditions are defined by the *signs* of the first and second derivatives. Denote by $\mathcal{B}_{[\mathbf{x}^*,\varepsilon]} = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x} - \mathbf{x}^*\|_2 \leq \varepsilon\}$ the closed ball with center $\mathbf{x}^* \in \mathbb{R}^n$ and radius $\varepsilon > 0$. Fix any $r > 0$, and consider a subset of $\mathcal{D}_{\mathbf{x}}$ in which all directions are bounded,

$$D_{\mathbf{x},r} \equiv \mathcal{D}_{\mathbf{x}} \cap \mathcal{B}_{[\mathbf{0},r]}.$$

Then, an equivalent statement, given in terms of the set $D_{\mathbf{x},r}$ instead of $\mathcal{D}_{\mathbf{x}^*}$, can be used to characterize (FO)/(SO) points.

**Lemma 2.3** (Necessary optimality conditions)**.** *Let $\mathbf{x}^* \in C, r > 0$. Then*

*(i) $\mathbf{x}^*$ is an (FO) point if and only if $f'(\mathbf{x}^*; \mathbf{d}) \geq 0$ for any $\mathbf{d} \in D_{\mathbf{x}^*,r}$.*

*(ii) $\mathbf{x}^*$ is an (SO) point if and only if it is an (FO) point and the following implication holds true for any $\mathbf{d} \in D_{\mathbf{x}^*,r}$*

$$f'(\mathbf{x}^*; \mathbf{d}) = 0 \;\Rightarrow\; f''(\mathbf{x}^*; \mathbf{d}) \geq 0.$$

# 3 A Two Direction Method for Second-Order Optimality

## 3.1 First-Order Based Oracle

Our goal in this paper is to develop a method to obtain points satisfying the (SO) condition. There are numerous algorithms that produce points satisfying the (FO) condition under various assumptions on the model, some of which are tailored for specific instances of (P). Since obtaining (FO) points is not our focus, the Two Directions (TD) method we propose will employ a generic step using a *First-Order Based Oracle (FOBO)* to ensure that any accumulation point of the TD method satisfies the (FO) condition. This generic step is based on the general concept of algorithms, as described in [27, Section 4]. In particular, FOBOs are point-to-set *closed maps*, a notion we recall from [27].

**Definition 3.1** (closed map [27, Section 4.4])**.** Let $\mathcal{M} : C \to C$ be a point-to-set map. The map $\mathcal{M}$ is said to be closed at $\mathbf{x} \in C$ if for any sequences $\{\mathbf{x}^k\}_{k\geq 0}$ and $\{\mathbf{y}^k\}_{k\geq 0}$ satisfying

$$\begin{aligned} \mathbf{x}^k \in C, \qquad & \mathbf{x}^k \to \mathbf{x}, \\ \mathbf{y}^k \in \mathcal{M}(\mathbf{x}^k), \qquad & \mathbf{y}^k \to \mathbf{y}, \end{aligned}$$

we have that $\mathbf{y} \in \mathcal{M}(\mathbf{x})$. The map $\mathcal{M}$ is said to be closed on $C$ if it is closed at each point in $C$.

The definition of FOBO is given next.

**Definition 3.2** (FOBO)**.** A closed mapping $\mathcal{M} : C \to C$ is called a FOBO of $f$ on $C$ if for any $\mathbf{x} \in C$ and $\mathbf{y} \in \mathcal{M}(\mathbf{x})$ exactly one of the following is satisfied: either $\mathbf{y} = \mathbf{x}$ and $\mathbf{x}$ is an (FO) point, or $f(\mathbf{y}) < f(\mathbf{x})$.

The conditions a FOBO is required to satisfy are elementary, and as such, are shared by many algorithms; see for example [18] or the already mentioned [2, 27], and references therein. In particular, the classical Projected Gradient Descent (PGD) method (e.g., [3, Section 9.4]) is a FOBO as the gradient mapping is a continuous mapping and any sufficiently small step-size will result in a closed map.

By themselves, the conditions a FOBO should satisfy provide little information on the performance of the FOBO. Hence, when a FOBO is analyzed in the literature, additional assumptions on the model are usually made in order to establish its convergence properties. These usually lead to a sufficient decrease-like property of the sequence generated by the FOBO. We will do the same, and make the following assumption in the forthcoming convergence analysis.

**Assumption 1** (FOBO sufficient decrease)**.** *Let $\mathcal{M} : C \to C$ be a FOBO. There exists $\gamma_{\mathcal{M}} > 0$ such that for any $\mathbf{x} \in C$ and $\mathbf{y} \in \mathcal{M}(\mathbf{x})$,*

$$f(\mathbf{x}) - f(\mathbf{y}) \geq \gamma_{\mathcal{M}} \|\mathbf{x} - \mathbf{y}\|^2. \tag{3.1}$$

Note that Assumption 1 holds true for classical first-order methods under the usual Lipschitz continuity property of the gradient of the objective function, see e.g. [3, 6].

## 3.2 The Two-Directions Method

The Two-Directions (TD) method, described by Algorithm 1, employs the following steps: (i) execute the selected FOBO on the current iterate to compute the first-order update $\mathbf{y}^k$; (ii) compute the second-order update by calculating the (feasible) direction $\mathbf{u}^k$, and the best step-size $q_k$ by exact line-search; (iii) update the iterate according to the best update among the first and second-order steps.

**Algorithm 1:** Two directions method (TD)

**Input.** $\mathbf{x}^0 \in C, r \in (0,\infty)$, a FOBO $\mathcal{M}(\cdot)$.

**General step.**

Step 1. First-order step:

$$\mathbf{y}^k \leftarrow \mathcal{M}(\mathbf{x}^k).$$

Step 2. Second-order step:

$$\begin{aligned} \mathbf{u}^k &\in \operatorname{argmin}\left\{ f''(\mathbf{x}^k;\mathbf{d}) \ : \ f'(\mathbf{x}^k;\mathbf{d}) \leq 0, \mathbf{d} \in D_{\mathbf{x}^k,r} \right\}; \\ q_k &\in \operatorname*{argmin}_{q\in[0,1]} f(\mathbf{x}^k + q\mathbf{u}^k). \end{aligned}$$

Step 3. Update

$$\mathbf{x}^{k+1} = \begin{cases} \mathbf{x}^k + q_k\mathbf{u}^k & \text{if } f(\mathbf{x}+q_k\mathbf{u}^k) \leq f(\mathbf{y}^k); \\ \mathbf{y}^k & \text{otherwise.} \end{cases}$$

Several remarks on the TD method are in order.

**Remark 3.1.** (i) The value of $r$ can be chosen arbitrarily, but must be finite in order to obtain the convergence properties given in Section 4. We emphasize that the feasible set of problem (P) *does not* have to be bounded.

(ii) We make the convention that if $\mathbf{d} = \mathbf{0}$ is a minimizer in the direction problem in Step 2, then $\mathbf{u}^k \equiv \mathbf{0}$, and if $q = 0$ is a minimizer in the step problem in Step 2, then $q_k \equiv 0$. We also make the convention that if there exist more than one solution to any of the optimization problems solved throughout the TD method's procedure, then there exists a predetermined rule to choose exactly one solution.

(iii) Step 2 of the TD algorithm requires two optimization procedures. One consists of finding $q_k$, which requires solving a univariate optimization problem over a compact interval. This is, in general, possible without any special requirements; see for example [23, Section 8.7] regarding search methods for univariate minimization, or specifically the commonly-used method [17]. The remaining one consists of computing $\mathbf{u}^k$, which is in general NP-hard. Nonetheless, the literature on quadratic constrained optimization provides various reasonable conditions under which the problem of computing $\mathbf{u}^k$ admits an exact SDP-relaxation, and thus can be solved in polynomial time. Moreover, in some circumstances, when the number of constraints is small, it is possible to solve this problem quite fast and efficiently; see Section 5.1 for a discussion on the issue.

Although our focus is on constrained minimization, it is interesting to examine the resulting TD method when the problem is unconstrained. Under the assumption that $C = \mathbb{R}^n$, the TD method is reduced to the simple procedure whose general step is described next.

**Unconstrained Two directions method (UTD).**
**Input.** $\mathbf{x}^0 \in \mathbb{R}^n, r \in (0, \infty)$, a FOBO $\mathcal{M}(\cdot)$. For any $k \geq 0$ do:

1. First-order step: $\mathbf{y}^k \leftarrow \mathcal{M}(\mathbf{x}^k)$.
2. Second-order step:
   - compute the direction $\mathbf{u}^k$ which is the eigenvector corresponding to the smallest negative eigenvalue of $\nabla^2 f(\mathbf{x}^k)$; if $\nabla^2 f(\mathbf{x}^k)$ is a PSD matrix, set $\mathbf{u}^k = \mathbf{0}$.
   - compute the step-size $q_k \in \underset{q \in [0,r]}{\operatorname{argmin}} f(\mathbf{x}^k + q\mathbf{u}^k)$.
3. Update
$$\mathbf{x}^{k+1} = \begin{cases} \mathbf{x}^k + q_k \mathbf{u}^k & \text{if } f(\mathbf{x} + q_k \mathbf{u}^k) \leq f(\mathbf{y}^k), \\ \mathbf{y}^k & \text{otherwise.} \end{cases}$$

We now turn to the analyze the convergence properties of the TD method, which are valid for the UTD method as well.

# 4 Analysis of the TD Method

## 4.1 Building Blocks

This subsection presents the building blocks used in the analysis. To avoid the clutter resulting from repeated input statements, we will make the convention that any sequence generated by the TD method was generated with the inputs $\mathbf{x}^0 \in C$, $r > 0$, and $\mathcal{M}(\cdot)$, without restating it. When additional assumptions are made on the inputs, we will explicitly write them.

The analysis of the TD method requires a Lipschitz continuity assumption on the Hessian of $f$, which we now state explicitly.

**Assumption 2.** *The Hessian of $f$ is Lipschitz continuous on $C$,*

$$\|\nabla^2 f(\mathbf{x}) - \nabla^2 f(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\| \quad \textit{for all} \quad \mathbf{x}, \mathbf{y} \in C,$$

*for some $L > 0$.*

**Remark 4.1** (The Lipschitz constant)**.** It is important to note that Assumption 2 is needed only to analyze the TD method. Indeed, we actually *do not* use the Lipschitz constant $L$, and *do not* require any information about it, to perform the steps of the algorithm.

Assumption 2 implies the *cubic descent lemma* introduced in [16], and commonly used when analyzing second-order methods, see e.g., [7, 22].

**Lemma 4.1** (Cubic descent lemma [16, Eq. (1.3)]). *Suppose that Assumption 2 holds true. Then for any* $\mathbf{x}, \mathbf{y} \in C$,

$$f(\mathbf{y}) \leq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{1}{2}\langle \mathbf{y} - \mathbf{x}, \nabla^2 f(\mathbf{x})(\mathbf{y} - \mathbf{x}) \rangle + \frac{L}{6}\|\mathbf{y} - \mathbf{x}\|_2^3.$$

To avoid ambiguity, we now summarize the blanket assumptions made throughout this section.

**Blanket Assumptions.**

(i) Assumption 1: The chosen FOBO satisfies a sufficient decrease property;

(ii) Assumption 2: The Hessian of $f$ is Lipschitz continuous on $C$.

Problem (P) is not necessarily convex, and so finding the optimal solution, or using any information that relates to it, is not possible. Therefore, instead of assessing the quality of a solution based on the optimal solution, we must define some other measures for optimality that quantify the violation of the necessary optimality conditions. Since here we do not specify the first-order oracle, but rather use a generic FOBO quantified by the map $\mathcal{M}$, we suggest to adopt the following (generic) first-order optimality measure.

**Definition 4.1** (First-order optimality measure). Let $\mathbf{x} \in C$. The first-order optimality measure at $\mathbf{x}$ is

$$S(\mathbf{x}) \equiv \inf\{\|\mathbf{z} - \mathbf{x}\| : \mathbf{z} \in \mathcal{M}(\mathbf{x})\}.$$

Indeed, by Definition 3.2 of the FOBO, $S(\mathbf{x}) = 0$ if and only if $\mathbf{x}$ satisfies the first-order necessary optimality condition. The following example illustrates the adequacy of $S(\cdot)$ in the context of classical first order methods.

**Example 4.1** (PGD first-order optimality measure). A classic result (e.g. [3, Thm. 9.10]) is that a point $\mathbf{x}^*$ satisfies the (FO) condition if and only if $\mathbf{x}^* = \mathrm{P}_C(\mathbf{x}^* - s\nabla f(\mathbf{x}^*))$ for some $s > 0$. When the FOBO is the PGD method with a constant step-size, i.e. $\mathcal{M}(\mathbf{x}) = \mathrm{P}_C(\mathbf{x} - s\nabla f(\mathbf{x}))$ (for $s$ sufficiently small), the first-order optimality measure is exactly the norm of the gradient mapping (see [3, Sec. 9.4.1]).

Next we define the second-order optimality measure.

**Definition 4.2** (Second -order optimality measure). Let $\mathbf{x} \in C$. The second-order optimality measure at $\mathbf{x}$ is

$$Q(\mathbf{x}) = -f''(\mathbf{x}; \mathbf{u}_\mathbf{x}),$$

where

$$\mathbf{u}_\mathbf{x} \in \operatorname{argmin}\{f''(\mathbf{x}; \mathbf{d}) : f'(\mathbf{x}; \mathbf{d}) \leq 0, \mathbf{d} \in D_{\mathbf{x},r}\}.$$

A simple comparison between Definition 4.2 and Step 2 in the TD method leads to the conclusion that

$$Q(\mathbf{x}^k) = -f''(\mathbf{x}^k; \mathbf{u}_{\mathbf{x}^k}) = -f''(\mathbf{x}^k; \mathbf{u}^k), \tag{4.1}$$

thus providing an intuitive reason for the choice of the second-order update direction in the TD method – choose a direction that violates the second-order optimality measure the most.

The next example illustrates the definition of the second-order optimality measure by examining it when the problem is unconstrained.

**Example 4.2.** Suppose that $C \equiv \mathbb{R}^n$, and let $\mathbf{x} \in \mathbb{R}^n$ be an (FO) point. Then $\nabla f(\mathbf{x}) = 0$, and subsequently

$$Q(\mathbf{x}) = -\min\{f''(\mathbf{x}; \mathbf{d}) : \|\mathbf{d}\|_2 \leq r\} = -r \min\{\lambda_n(\nabla^2 f(\mathbf{x})), 0\},$$

where $\lambda_n(\nabla^2 f(\mathbf{x}))$ is the smallest eigenvalue of $\nabla^2 f(\mathbf{x})$.

Measures for optimality should be nonnegative scalars that equal zero whenever the point does satisfy the optimality condition. The next lemma verifies that the defined measures indeed satisfy the latter.

**Lemma 4.2** (Optimality measures)**.** *Let $\mathbf{x} \in C$. Then,*

*(i)* $S(\mathbf{x}), Q(\mathbf{x}) \geq 0$*;*

*(ii)* $\mathbf{x}$ *is and (FO) point if and only if* $S(\mathbf{x}) = 0$*.*

*(iii) Suppose that* $\mathbf{x}$ *is an (FO) point. Then* $Q(\mathbf{x}) = 0$ *if and only if* $\mathbf{x}$ *satisfies the (SO) condition.*

*Proof.* (i) Follows from the definitions of $S(\cdot)$ and $Q(\cdot)$ respectively. Likewise, (ii) follows from the definition of $S(\cdot)$.

(iii) It holds that $\mathbf{x}$ is an (SO) point if and only if any (recall that $\mathbf{x}$ is an (FO) point)

$$\mathbf{w} \in \{\mathbf{d} \in D_{\mathbf{x},r} : f'(\mathbf{x}; \mathbf{d}) \leq 0\},$$

satisfies $f''(\mathbf{x}; \mathbf{w}) \geq 0$. By the definition of $\mathbf{u}_{\mathbf{x}}$, the latter holds if and only if $f''(\mathbf{x}; \mathbf{u}_{\mathbf{x}}) = 0$, which in turn holds if and only if $Q(\mathbf{x}) = 0$.

□

Lemma 4.2 suggests that a natural measure to quantify the violation of the (SO) condition is the maximum between the first-order optimality measure $S(\cdot)$ and the second-order optimality measure $Q(\cdot)$:

$$\mathcal{E}(\mathbf{x}) := \max\{S(\mathbf{x}), Q(\mathbf{x})\}.$$

Obviously, a point $\mathbf{x} \in C$ is an (SO) point if and only if $\mathcal{E}(\mathbf{x}) = 0$.

Finally, we record an important continuity property for $Q(\cdot)$ which follows from a sensitivity and stability result of nonlinear programs given in [12, Theorem 2.2.2]; note that the latter was derived from combining Theorem 1 and Theorem 2 in [5, Section 3].

**Lemma 4.3** (Continuity of the second-order optimality measure)**.** *The second-order optimality measure $Q : C \to [0, \infty)$, which is defined by*

$$Q(\mathbf{x}) = -\min\left\{f''(\mathbf{x};\mathbf{d}) : f'(\mathbf{x};\mathbf{d}) \le 0, \mathbf{d} \in D_{\mathbf{x},r}\right\},$$

*is a continuous function.*

## 4.2 Convergence Properties of the TD Method

We are now ready to analyze the convergence properties of the TD method. The main highlights are, as always, the subsequence convergence and the convergence rate results, both of which are derived using a sufficient decrease property (Lemma 4.5). To prove the sufficient decrease of the TD method, we first require a simple property of one-dimensional quadratic functions.

**Lemma 4.4.** *Let $\alpha > 0$, $\beta \le 0$, and $\gamma \in \mathbb{R}$. Define $g(t) = \alpha t^2 + 2\beta t + \gamma$ and $t^* \in \operatorname{argmin}\{g(t) : t \in [0,1]\}$. Then*

$$-t^* g(t^*) \ge \min\left\{-\gamma - \beta, \frac{\beta}{\alpha}\gamma - \frac{\beta^3}{\alpha^2}\right\}.$$

*Proof.* Since $\alpha > 0$ and $\beta \le 0$, it is easy to see that the minimizer $t^*$ of the convex function $g(t)$ over $t \in [0,1]$ is:

$$t^* = \begin{cases} -\frac{\beta}{\alpha}, & 1 \ge -\frac{\beta}{\alpha}, \\ 1, & 1 < -\frac{\beta}{\alpha}, \end{cases} \quad \text{and hence} \quad g(t^*) = \begin{cases} -\frac{\beta^2}{\alpha} + \gamma, & 1 \ge -\frac{\beta}{\alpha}, \\ \alpha + 2\beta + \gamma, & 1 < -\frac{\beta}{\alpha}. \end{cases}$$

Suppose that $1 < -\frac{\beta}{\alpha}$. Then $t^* = 1$ and $t^* g(t^*) = g(1) = \alpha + 2\beta + \gamma \le \beta + \gamma$. Otherwise, $t^* = -\frac{\beta}{\alpha}$ and $t^* g(t^*) = -\frac{\beta}{\alpha}\gamma + \frac{\beta^3}{\alpha^2}$. To conclude, $t^* g(t^*) \le \max\left\{\gamma + \beta, -\frac{\beta}{\alpha}\gamma + \frac{\beta^3}{\alpha^2}\right\}$, and the required immediately follows. □

**Lemma 4.5** (Sufficient decrease property)**.** *Let $\{(\mathbf{x}^k, \mathbf{u}^k, q_k)\}_{k\ge 0}$ be a sequence generated by the TD method. Then for any $k \ge 0$ it holds that*

$$f(\mathbf{x}^k) - f(\mathbf{x}^{k+1}) \ge \max\left\{\min\left\{\frac{1}{4}Q(\mathbf{x}^k), \frac{9}{16L^2}Q(\mathbf{x}^k)^3\right\}, \gamma_{\mathcal{M}} S(\mathbf{x}^k)^2\right\}. \tag{4.2}$$

*Proof.* Suppose that the TD method activates Step 3 with $\mathbf{y}^k \in \mathcal{M}(\mathbf{x}^k)$ such that $f(\mathbf{x}^k + q_k\mathbf{u}^k) > f(\mathbf{y}^k)$. Then $\mathbf{x}^{k+1} = \mathbf{y}^k$, and thanks to Assumption 1 we obtain

$$f(\mathbf{x}^k) - f(\mathbf{x}^{k+1}) = f(\mathbf{x}^k) - f(\mathbf{y}^k) \ge \gamma_{\mathcal{M}} \|\mathbf{x}^k - \mathbf{y}^k\|^2 \ge \gamma_{\mathcal{M}} S(\mathbf{x}^k)^2, \tag{4.3}$$

where the last inequality follows from the definition of the first optimality measure, which warrants $S(\mathbf{x}^k) \le \|\mathbf{x}^k - \mathbf{y}^k\|$ (see Definition 4.1).

Otherwise, $f(\mathbf{y}^k) \geq f(\mathbf{x}^k + q_k\mathbf{u}^k)$ and the TD method activates Step 3 with $\mathbf{x}^{k+1} = \mathbf{x}^k + q_k\mathbf{u}^k$, where $q_k \in \underset{q\in[0,1]}{\operatorname{argmin}} f(\mathbf{x}^k + q\mathbf{u}^k)$. Therefore, for any $t \in [0,1]$:

$$\begin{aligned} f(\mathbf{x}^k) - f(\mathbf{x}^{k+1}) &= f(\mathbf{x}^k) - f(\mathbf{x}^k + q_k\mathbf{u}^k) \\ &\geq f(\mathbf{x}^k) - f(\mathbf{x}^k + t\mathbf{u}^k) \\ &\geq -t\left(f'(\mathbf{x}^k;\mathbf{u}^k) + \frac{1}{2}tf''(\mathbf{x}^k;\mathbf{u}^k) + \frac{L}{6}t^2\right), \end{aligned} \tag{4.4}$$

where the first inequality follows from the definition of $q_k$ and the second from the cubic descent lemma (see Lemma 4.1). Let $g(t) := f'(\mathbf{x}^k;\mathbf{u}^k) + \frac{1}{2}f''(\mathbf{x}^k;\mathbf{u}^k)t + \frac{L}{6}t^2$. Then the inequality (4.4) reads

$$f(\mathbf{x}^k) - f(\mathbf{x}^{k+1}) \geq -tg(t), \qquad \forall t \in [0,1],$$

which in particular holds true for $t^* \in \operatorname{argmin}\{g(t) : t \in [0,1]\}$, i.e., $f(\mathbf{x}^k) - f(\mathbf{x}^{k+1}) \geq -t^*g(t^*)$. Therefore, invoking Lemma 4.4 with $\alpha = \frac{L}{6}, \beta = \frac{1}{4}f''(\mathbf{x}^k;\mathbf{u}^k)$ $(\leq 0)$, and $\gamma = f'(\mathbf{x}^k;\mathbf{u}^k)$, it follows that

$$f(\mathbf{x}^k) - f(\mathbf{x}^{k+1}) \geq \min\{c_1^k, c_2^k\}, \tag{4.5}$$

where

$$\begin{aligned} c_1^k &= -f'(\mathbf{x}^k;\mathbf{u}^k) - \frac{1}{4}f''(\mathbf{x}^k;\mathbf{u}^k), \\ c_2^k &= \frac{3}{2L}f'(\mathbf{x}^k;\mathbf{u}^k)f''(\mathbf{x}^k;\mathbf{u}^k) - \frac{9}{16L^2}f''(\mathbf{x}^k;\mathbf{u}^k)^3. \end{aligned}$$

By combining (4.3) and (4.5) we deduce that

$$f(\mathbf{x}^k) - f(\mathbf{x}^{k+1}) \geq \max\left\{\min\left\{c_1^k, c_2^k\right\}, \gamma_{\mathcal{M}}S(\mathbf{x}^k)^2\right\}. \tag{4.6}$$

We will now bound $c_1^k$ and $c_2^k$ from below. For that purpose, recall that by definition of the TD method (Step 2) we have that

$$f'(\mathbf{x}^k;\mathbf{u}^k) \leq 0 \tag{4.7}$$

and

$$f''(\mathbf{x}^k;\mathbf{u}^k) = f''(\mathbf{x}^k;\mathbf{u}_{\mathbf{x}^k}) = -Q(\mathbf{x}^k), \tag{4.8}$$

where the equality folllows from (4.1). Consequently, using (4.7), (4.8), and the fact that $f''(\mathbf{x}^k;\mathbf{u}^k) \leq 0$, we obtain

$$\begin{aligned} c_1^k &= -f'(\mathbf{x}^k;\mathbf{u}^k) - \frac{1}{4}f''(\mathbf{x}^k;\mathbf{u}^k) \geq \frac{1}{4}Q(\mathbf{x}^k), \\ c_2^k &= \frac{3}{2L}f'(\mathbf{x}^k;\mathbf{u}^k)f''(\mathbf{x}^k;\mathbf{u}^k) + \frac{9}{16L^2}\left(-f''(\mathbf{x}^k;\mathbf{u}^k)\right)^3 \geq \frac{9}{16L^2}Q(\mathbf{x}^k)^3. \end{aligned}$$

Finally, plugging these bounds in (4.6) yields the required (4.2). □

The next theorem establishes that any limit point of the TD method is an (SO) point.

**Theorem 4.1** (Convergence properties)**.** *Let $\{(\mathbf{x}^k, \mathbf{y}^k)\}_{k\geq 0}$ be a sequence generated by the TD method. Then:*

*(i) the sequence $\{f(\mathbf{x}^k)\}_{k\geq 0}$ is non-increasing.*

*(ii)* $\lim_{k\to\infty} \mathcal{E}(\mathbf{x}^k) \equiv \lim_{k\to\infty} \max\{S(\mathbf{x}^k), Q(\mathbf{x}^k)\} = 0.$

*(iii) Any accumulation point of $\{(\mathbf{x}^k, \mathbf{y}^k)\}_{k\geq 0}$ is an (SO) point.*

*Proof.* (i) Follows immediately from the update step in the TD method (Step 3).

(ii) From summing the sufficient decrease relation (4.2) in Lemma 4.5 over $k = 0, 1, \ldots, K$, (for any $K \geq 0$) we obtain

$$f(\mathbf{x}^0) - f_* \geq f(\mathbf{x}^0) - f(\mathbf{x}^{K+1}) \geq \sum_{k=0}^{K} \max\left\{\min\left\{\frac{1}{4}Q(\mathbf{x}^k), \frac{9}{16L^2}Q(\mathbf{x}^k)^3\right\}, \gamma_{\mathcal{M}} S(\mathbf{x}^k)^2\right\}. \tag{4.9}$$

Then, taking the limit $K \to \infty$ in (4.9) results with

$$f(\mathbf{x}^0) - f_* \geq \sum_{k=0}^{\infty} \max\left\{\min\left\{\frac{1}{4}Q(\mathbf{x}^k), \frac{9}{16L^2}Q(\mathbf{x}^k)^3\right\}, \gamma_{\mathcal{M}} S(\mathbf{x}^k)^2\right\},$$

which in turn implies that $\lim_{k\to\infty} \mathcal{E}(\mathbf{x}^k) \equiv \lim_{k\to\infty} \max\{S(\mathbf{x}^k), Q(\mathbf{x}^k)\} = 0.$

(iii) Let $(\mathbf{x}^*, \mathbf{y}^*) \in C \times C$ be an accumulation point of $\{(\mathbf{x}^k, \mathbf{y}^k)\}_{k\geq 0}$ , and let $\{(\mathbf{x}^{k_j}, \mathbf{y}^{k_j})\}_{j\geq 1}$ be a subsequence that converges to $(\mathbf{x}^*, \mathbf{y}^*)$. We will first prove that $\mathbf{x}^*$ is an (FO) point, and then that $\mathbf{x}^*$ is an (SO) point.

Recall that by Step 1 of the TD method, the sequence $\{\mathbf{y}^{k_j}\}_{j\geq 1}$ is generated by $\mathbf{y}^{k_j} \in \mathcal{M}(\mathbf{x}^{k_j})$. Thus, by the definition of FOBO (cf. Definition 3.2), one has $f(\mathbf{x}^{k_j}) \geq f(\mathbf{y}^{k_j})$, and $\mathbf{y}^* \in \mathcal{M}(\mathbf{x}^*)$, (thanks to the closeness of $\mathcal{M}$). Now, since the sequence $\{f(\mathbf{x}^k)\}_{k\geq 0}$ is non-increasing and bounded below, it is convergent, and hence

$$f(\mathbf{x}^*) = \lim_{j\to\infty} f(\mathbf{x}^{k_j}) \geq \lim_{j\to\infty} f(\mathbf{y}^{k_j}) \geq \lim_{j\to\infty} f(\mathbf{x}^{k_j+1}) = f(\mathbf{x}^*),$$

where the inequality $f(\mathbf{y}^{k_j}) \geq f(\mathbf{x}^{k_j+1})$ follows from Step 3 of the TD method. Consequently, from the above relations it follows that $f(\mathbf{x}^*) = f(\mathbf{y}^*)$, and by the fact that $\mathbf{y}^* \in \mathcal{M}(\mathbf{x}^*)$, one must have (again by the definition of FOBO) that $\mathbf{x}^* = \mathbf{y}^*$ and that $\mathbf{x}^*$ satisfies the (FO) condition (i.e., $S(\mathbf{x}^*) = 0$).

Since $Q(\cdot)$ is continuous (cf. Lemma 4.3) and nonnegative (cf. Lemma 4.2), Part (ii) implies that

$$Q(\mathbf{x}^*) = \lim_{j\to\infty} Q(\mathbf{x}^{k_j}) = 0.$$

Subsequently, $\mathcal{E}(\mathbf{x}^*) = \max\{S(\mathbf{x}^*), Q(\mathbf{x}^*)\} = 0$, meaning that $\mathbf{x}^*$ is an (SO) point.

□

We conclude this section with a convergence rate and complexity result for the TD method.

**Theorem 4.2** (Convergence rate and complexity)**.** *Let $\{\mathbf{x}^k\}_{k\geq 0}$ be a sequence generated by the TD method, and denote $\Delta := f(\mathbf{x}^0) - f_*$. Then:*

*(i) For any $K \geq k \geq 0$ the following holds true:*

$$\min_{k=0,1,\ldots,K} \mathcal{E}(\mathbf{x}^k) \leq \max\left\{\frac{4\Delta}{K+1}, \sqrt{\frac{\Delta}{\gamma_{\mathcal{M}}(K+1)}}, \sqrt[3]{\frac{16L^2\Delta}{9(K+1)}}\right\}. \tag{4.10}$$

*(ii) Let $\epsilon > 0$. Then for any*

$$K \geq \max\left\{\frac{4\Delta}{\epsilon}, \frac{\Delta}{\gamma_{\mathcal{M}}\epsilon^2}, \frac{16L^2\Delta}{9\epsilon^3}\right\} - 1$$

*it holds that $\min_{k=0,1,\ldots,K} \mathcal{E}(\mathbf{x}^k) \leq \epsilon$.*

*Proof.* (i) As already proven in Theorem 4.1 (cf. relation (4.9)), summing the sufficient decrease relation (4.2) in Lemma 4.5 over $k = 0, 1, \ldots, K$ (for any $K \geq 0$) results in

$$f(\mathbf{x}^0) - f(\mathbf{x}^{K+1}) \geq \sum_{k=0}^{K} \max\left\{\min\left\{\frac{1}{4}Q(\mathbf{x}^k), \frac{9}{16L^2}Q(\mathbf{x}^k)^3\right\}, \gamma_{\mathcal{M}} S(\mathbf{x}^k)^2\right\}. \tag{4.11}$$

Thus,

$$f(\mathbf{x}^0) - f(\mathbf{x}^{K+1}) \geq (K+1) \min_{k=0,1,\ldots,K} \max\left\{\min\left\{\frac{1}{4}Q(\mathbf{x}^k), \frac{9}{16L^2}Q(\mathbf{x}^k)^3\right\}, \gamma_{\mathcal{M}} S(\mathbf{x}^k)^2\right\},$$

and since $f(\mathbf{x}^{K+1}) \geq f_* > -\infty$, we have that

$$\min_{k=0,1,\ldots,K} \max\left\{\min\left\{\frac{1}{4}Q(\mathbf{x}^k), \frac{9}{16L^2}Q(\mathbf{x}^k)^3\right\}, \gamma_{\mathcal{M}} S(\mathbf{x}^k)^2\right\} \leq \frac{f(\mathbf{x}^0) - f_*}{K+1} = \frac{\Delta}{K+1}.$$

The last inequality implies that there exists $k \in \{0, 1, \ldots, K\}$ such that

$$\max\left\{\min\left\{\frac{1}{4}Q(\mathbf{x}^k), \frac{9}{16L^2}Q(\mathbf{x}^k)^3\right\}, \gamma_{\mathcal{M}} S(\mathbf{x}^k)^2\right\} \leq \frac{\Delta}{K+1}.$$

Hence,

$$S(\mathbf{x}^k) \leq \sqrt{\frac{\Delta}{\gamma_{\mathcal{M}}(K+1)}}, \tag{4.12}$$

and

$$\min\left\{\frac{1}{4}Q(\mathbf{x}^k), \frac{9}{16L^2}Q(\mathbf{x}^k)^3\right\} \leq \frac{\Delta}{K+1},$$

which implies that

$$Q(\mathbf{x}^k) \leq \max\left\{\frac{4\Delta}{K+1}, \sqrt[3]{\frac{16L^2\Delta}{9(K+1)}}\right\}. \tag{4.13}$$

Therefore, by (4.12) and (4.13), there exists $k \in \{0, 1, \ldots, K\}$ such that

$$\mathcal{E}(\mathbf{x}^k) = \max\{S(\mathbf{x}^k), Q(\mathbf{x}^k)\} \leq \max\left\{\frac{4\Delta}{K+1}, \sqrt{\frac{\Delta}{\gamma_{\mathcal{M}}(K+1)}}, \sqrt[3]{\frac{16L^2\Delta}{9(K+1)}}\right\},$$

which proves the relation (4.10).

(ii) The required immediately follows from part (i) by using simple algebra.

□

# 5 Implementation and Numerical Illustrations

The focus of this paper is theoretical. Nevertheless, to supplement our theoretical analysis and in order to demonstrate the simplicity and versatility of the TD method, in this final section we briefly discuss the implementation aspects of the method and present two simple experiments.

## 5.1 Implementation aspects

In the core of the TD method stands the computation of the second-order direction (cf. Step 2 of the algorithm) which generically can be formulated as solving a nonconvex quadratic minimization problem of the form:

$$(NQP) \qquad \min\{z^T Q z - 2b^T z : \ \|z - c\| \leq r, z \in C\},$$

where $Q, b, c$ are given, and easy to identify.

In general, (NQP) is intractable. However, when it belongs to the class of *quadratically constrained quadratic problems* (QCQP) (e.g. whenever $C$ is described by quadratic constraints, a setting that is common in many applications), there are various reasonable conditions under which it admits an exact SDP-relaxation, and thus can be solved in polynomial time; see e.g., [19], and [4], where the latter also provides a concise review on the topic. When the number of constraints is also small, the problem can be solved quite fast (cf. [4, Section 6.1]) by the branch and bound method developed by [4] (which by itself is not limited to the QCQP setting). Moreover, many modern applications are modeled as nonconvex problems comprising very few constraints (see e.g., [20]), which means that the TD method can be used in a variety of situations to obtain better solutions compared to first-order methods, making it a very attractive and tractable choice for obtaining (SO) points in nonconvex problems.

We emphasize that whenever (NQP) can be solved efficiently, the TD method can be executed efficiently.

## 5.2 Numerical Illustrations

All of the numerical experiments will share the following:

- We will always compare the TD method to the FOBO it utilizes.
- The TD method will always use the single-step operator of the classic Projected Gradient Descent (PGD) method with a backtracking-determined step-size as its FOBO.
- Whenever a stopping criteria is required, we use the rule $\|\mathbf{x}^k - \mathbf{x}^{k+1}\|_2 \leq 10^{-5}$.

Note that under these settings, a TD step and a PGD step executed on the same point can differ from each other only if the Hessian at that point is not a PSD matrix. Hence, any difference in the performance of the two compared methods must be due to the way the TD method exploits the second-order information at points with a non-PSD Hessian.

### 5.2.1 The Egg Crate Experiment

The egg crate experiment compares the PGD and the TD methods on the nonconvex, unconstrained, multi-dimensional egg crate problem:

$$\min_{\mathbf{x}\in\mathbb{R}^n} \left\{ f(\mathbf{x}) \equiv \|\mathbf{x}\|^2 + 25 \sum_{i=1}^{n} \sin^2(x_i) \right\}. \tag{5.1}$$

The optimal solution of (5.1) is $\mathbf{x}^* = \mathbf{0}$, with an optimal function value of $f(\mathbf{x}^*) = 0$.

The two-dimensional egg crate function [25, Appendix A] is considered a benchmark function for algorithm comparison, mainly in the field of global nature-based search methods; see for example [25] and references therein. Figure 1 illustrates the oddity of the function in the two-dimensional case.

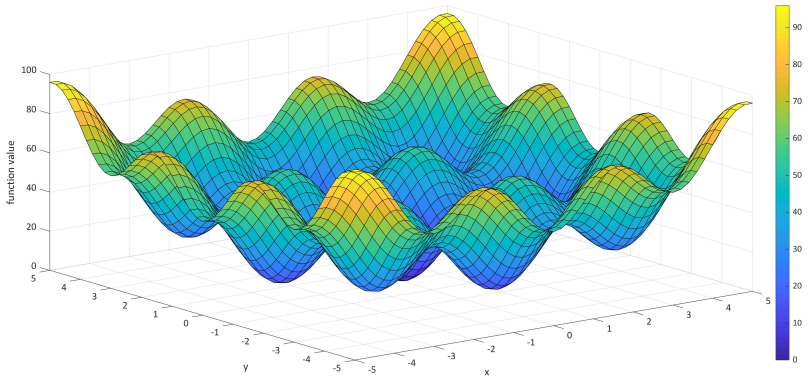


Figure 1: The egg crate function plotted on $[-5, 5]^2$.

To solve the four-dimensional egg crate problem, we ran both methods (the TD method used $r = 1$) from the same $10^5$ random starting points generated by the uniform distribution

over the $\ell_\infty$-norm ball with radius 2. The percentage of runs in which each method obtained the optimal solution, and the time it took for each method to obtain the optimal solution (when obtained), were measured. Additionally, the percentage of points passed through by the TD method (until termination) in which the Hessian was not a PSD matrix was calculated per random starting point. The overall average results are listed in Table 1.

| | % reached global opt. | avg. time until reaching global opt. | avg. %non-PSD pt. |
|---|---|---|---|
| PGD | 31.74% | $7.33\times10^{-4}$ sec | |
| TD | 38.45% | $5.91\times10^{-4}$ sec | 31.42% |

Table 1: Performance summary of the PDG and TD methods in the attempt of attaining the global solution in the $10^5$ runs.

The TD method indeed attains the optimal solution in a larger portion of the experiments, and faster, compared to the PGD method. This can be affiliated with the fact that, in average, the TD method updated the decision variable according to the second-order step in 31.42% of the points it passed.

### 5.2.2 Binary Classification with norm-ball constraint

In this section we test the TD method on a synthetic binary classification problem with an $\ell_2$-norm ball constraint (e.g. [20, Section 4.1]):

$$\begin{array}{ll} \min & \sum_{i=1}^{T}\left((1+e^{\mathbf{a}_i^T\mathbf{x}})^{-1}-y_i\right)^2 \\ \text{s.t.} & \|\mathbf{x}\|_2^2 \le \alpha^2. \end{array}$$

The choice $C \equiv \mathcal{B}_{[\mathbf{0},\alpha]}$ means that for any $r > 2\alpha$ the inclusion

$$\{\mathbf{d}\in\mathbb{R}^n : \|\mathbf{x}+\mathbf{d}\|_2 \le \alpha\} \subseteq D_{\mathbf{x},r} \qquad \forall \mathbf{x}\in\mathcal{B}_{[\mathbf{0},\alpha]}$$

holds true, and thus a quadratic constraint can be omitted from the problem in Step 2 of the TD method. Consequently, computing $\mathbf{u}^k$ reduces to solving the problem

$$\min_{d\in\mathbb{R}^n}\left\{f''(\mathbf{x}^k;\mathbf{d}) \;:\; f'(\mathbf{x}^k;\mathbf{d})\le 0, \|\mathbf{x}^k+\mathbf{d}\|_2^2\le\alpha^2\right\}. \tag{5.2}$$

In this experiment we chose $r = 5$ and $\alpha = 2$, and solved (5.2) using the Matlab package implementing the branch and bound procedure developed by [4].[1] Since (5.2) includes only two constraints, a global solution was quickly obtained in all the runs of the TD method.

This experiment compared the average function value and running time of the TD and PGD methods in various sizes of the problem. For each problem's size, one hundred problems were randomly generated where: $\{\mathbf{a}_i\}_{i=1}^T$ was generated from the uniform distribution over the box set $[-60,40]^n$, $\mathbf{y}$ was generated uniformly from $\{0,1\}^n$, and the starting point was

| $n$ | $T$ | TD runtime ($10^{-2}$ sec.) | PGD runtime ($10^{-2}$ sec.) | TD fun. val. | PGD fun. val. |
|---|---|---|---|---|---|
| 50 | 100 | 0.57 | 9.71 | 15.22 | 15.32 |
| 50 | 200 | 0.72 | 75.14 | 37.16 | 38.04 |
| 100 | 150 | 2.05 | 21.58 | 12.30 | 14.70 |
| 100 | 300 | 1.73 | 56.57 | 48.47 | 49.60 |
| 100 | 500 | 3.50 | 60.09 | 103.18 | 103.71 |
| 150 | 200 | 3.62 | 17.87 | 18.80 | 19.70 |
| 200 | 300 | 5.90 | 34.32 | 30.10 | 30.90 |
| 200 | 600 | 8.27 | 81.06 | 106.19 | 113.52 |
| 200 | 800 | 10.47 | 549.01 | 159.96 | 161.95 |
| 250 | 350 | 6.35 | 36.98 | 39.20 | 40.00 |
| 300 | 400 | 10.69 | 74.71 | 42.25 | 46.15 |

Table 2: Average function value and running time of the TD and PGD methods in varying sizes of the problem.

always the zeros vector. The results are listed in Table 2.

The results demonstrate that, on average, the TD method converges much faster compared to the PGD method, and always to points with lower function value.

# References


[1] A. Auslender. Computing points that satisfy second order necessary optimality conditions for unconstrained minimization. *SIAM Journal on Optimization*, 20(4):1868–1884, 2010.

[2] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty. *Nonlinear programming: theory and algorithms*. John Wiley & Sons, 2006.

[3] A. Beck. *Introduction to nonlinear optimization*, volume 19 of *MOS-SIAM Series on Optimization*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2014.

[4] A. Beck and D. Pan. A branch and bound algorithm for nonconvex quadratic optimization with ball and linear constraints. *Journal of Global Optimization*, 69(2):309–342, 2017.

[5] C. Berge. *Topological Spaces: including a treatment of multi-valued functions, vector spaces, and convexity*. Courier Corporation, 1997.

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


[1] See the link `https://www.tau.ac.il/~becka/BB_Documentation.7z`.

[7] C. Cartis, N. I. M. Gould, and P. L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. Part I: motivation, convergence and numerical results. *Math. Program.*, 127(2, Ser. A):245–295, 2011.

[8] C. Cartis, N. I. M. Gould, and P. L. Toint. An adaptive cubic regularization algorithm for nonconvex optimization with convex constraints and its function-evaluation complexity. *IMA J. Numer. Anal.*, 32(4):1662–1695, 2012.

[9] C. Cartis, N. I. M. Gould, and P. L. Toint. Second-order optimality and beyond: characterization and evaluation complexity in convexly constrained nonlinear optimization. *Found. Comput. Math.*, 18(5):1073–1107, 2018.

[10] A. R. Conn, N. I. M. Gould, and P. L. Toint. *Trust region methods*, volume 1. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000.

[11] F. Facchinei and S. Lucidi. Convergence to second order stationary points in inequality constrained optimization. *Mathematics of Operations Research*, 23(3):746–766, 1998.

[12] A. V. Fiacco. *Introduction to sensitivity and stability analysis in nonlinear programming, Volume 165 (Mathematics in Science and Engineering)*. Academic Press, 1983.

[13] A. Forsgren and W. Murray. Newton methods for large-scale linear inequality-constrained minimization. *SIAM Journal on Optimization*, 7(1):162–176, 1997.

[14] P. E. Gill and W. Murray. Newton-type methods for unconstrained and linearly constrained optimization. *Mathematical Programming*, 7(1):311–350, 1974.

[15] N. I. M. Gould, S. Lucidi, M. Roma, and P. L. Toint. Exploiting negative curvature directions in linesearch methods for unconstrained optimization. *Optimization methods and software*, 14(1-2):75–98, 2000.

[16] A. Griewank. The modification of newton's method for unconstrained optimization by bounding cubic terms. Technical report, Technical report NA/12, 1981.

[17] E. R. Hansen. Global optimization using interval analysis: the one-dimensional case. *Journal of Optimization Theory and Applications*, 29(3):331–344, 1979.

[18] P. Huard. Optimization algorithms and point-to-set-maps. *Mathematical Programming*, 8(1):308–331, 1975.

[19] V. Jeyakumar and G. Li. Trust-region problems with linear inequality constraints: exact sdp relaxation, global optimality and robust optimization. *Mathematical Programming*, 147(1-2):171–206, 2014.

[20] S. Mei, Y. Bai, and A. Montanari. The landscape of empirical risk for nonconvex losses. *The Annals of Statistics*, 46(6A):2747–2774, 2018.

[21] J. J. More and D. C. Sorensen. On the use of directions of negative curvature in a modified newton method. *Mathematical Programming*, 16(1):1–20, 1979.

[22] Y. Nesterov and B .T Polyak. Cubic regularization of newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

[23] J. M. Ortega and W. C. Rheinboldt. *Iterative solution of nonlinear equations in several variables*, volume 30. Siam, 1970.

[24] G. Di Pillo, S. Lucidi, and L. Palagi. Convergence to second-order stationary points of a primal-dual algorithm model for nonlinear programming. *Mathematics of Operations Research*, 30(4):897–915, 2005.

[25] X. Yang. *Nature-Inspired Metaheuristic Algorithms: Second Edition*. LUNIVER PR, 2010.

[26] Y. Yuan. Recent advances in trust region algorithms. *Mathematical Programming*, 151(1):249–281, 2015.

[27] W. I. Zangwill. *Nonlinear programming: a unified approach*, volume 196. Prentice-Hall Englewood Cliffs, NJ, 1969.