Beyond local optimality conditions: the case of maximizing a convex function

Aharon Ben-Tal[1] and Ernst Roos*[2]

[1]Faculty of Industrial Engineering and Management, Technion - Israel Institute of Technology, Shenkar College Israel

[2]Tilburg University, Department of Econometrics & Operations Research

February 25, 2021

Abstract

In this paper, we design an algorithm for maximizing a convex function over a convex feasible set. The algorithm consists of two phases: in phase 1 a feasible solution is obtained that is used as an initial starting point in phase 2. In the latter, a biconvex problem equivalent to the original problem is solved by employing an alternating direction method. The main contribution of the paper is connected to the first phase. We identify a kind of global optimality condition which says that "The maximizer of a convex objective function is the furthest point from the minimizer". Using this principle, we develop several ways to compute this 'furthest point', focusing on methods that lead to computationally efficient algorithms. The performance of the overall algorithm is tested on a wide variety of problems, demonstrating its efficiency.

Keywords: global optimization, convex maximization, alternating direction

1 Introduction

1.1 Contributions

The aim of this paper is to develop an algorithm for solving the following convex maximization problem to global optimality:

$$\max_{\boldsymbol{x}\in\mathbb{R}^n} \{f(\boldsymbol{x}) \mid \boldsymbol{x} \in X\}, \tag{P}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a closed, convex and continuously differentiable function, and $X \subseteq \mathbb{R}^n$ is a compact convex set. This is a nonconvex optimization problem which is known to be NP-hard (Zwart, 1974). Convex maximization has many applications, which we discuss in more detail in Section 1.2.

*Corresponding author: e.j.roos@tilburguniversity.edu

Traditionally, conditions under which a feasible solution $\boldsymbol{x}^*$ of (P) is identified as optimal are based on the body of knowledge in mathematics covered by *Infinitesimal Calculus* (IC). Citing from (Blåsjö, 2015, p. 3): "The basic idea of the calculus is to analyze functions by means of their behavior on a *micro level* [emphasis added]". Consequently, IC does not provide global optimality conditions for nonconvex problems such as (P). Thus, something new is needed to help identify $\boldsymbol{x}^*$ as a global maximizer, not coming from IC-based conditions such as Karash-Kuhn-Tucker optimality conditions (Kuhn, 1976). What we offer is an intuitvely appealing 'global optimality condition' for (P):

"The global maximizer $\boldsymbol{x}^*$ of problem (P) is the furthest feasible point from $\bar{\boldsymbol{x}}$, the minimizer of problem (P)"

To use this condition, several steps must be performed. First, a *distance measure* must be chosen in computing the furthest point from $\bar{\boldsymbol{x}}$. This is presented in detail in Section 3. We elect to use the second-order Taylor approximation at the global minimum, as it captures both the notion of difference as well as function specific characteristics through the gradient and Hessian at the minimum. Second, the minimizer of (P) must be computed. This, however, is a tractable problem of minimizing a convex function over a convex set. Third, the furthest point, denoted by $\hat{\boldsymbol{x}}$, is not taken as the actual maximizer of (P), rather it is used as a promising 'starting point' in a second phase algorithm that converges to a local maximizer of (P). This local maximum is indeed a good candidate to be a global maximum, since it improves the objective function from the value at the furthest point, which is already a good approximation of the true maximizer of (P). Table 3 in Section 6 provides a good demonstration of this claim.

The phase 2 algorithm developed in Section 2 is based on converting (P) into an equivalent *biconvex optimiztion problem*, and solving it with an alternating direction method. This method only requires to solve a problem of maximizing a *linear* function over the convex set X, which is a great advantage in terms of computational tractability. In particular, it means that when X is a polytope, the algorithm we propose can be applied to problems of type (P) with integer valued variables as well.

Phase 1 of our algorithm, in which the furthest point $\hat{\boldsymbol{x}}$ from the minimizer $\bar{\boldsymbol{x}}$ is computed, reduces to the maximization of just a convex *quadratic* function. This is still an NP-hard problem in general, but one that is the most studied convex maximization problem. In fact, for some feasible sets, this problem can be solved to global optimality efficiently. In particular, for an ellipsoidal feasible set X, the problem of finding the furthest point can be converted into an equivalent convex optimization problem. This is known as hidden convexity (Ben-Tal and Teboulle, 1996), which is discussed in more detail in (Ben-Tal and Den Hertog, 2014). For other feasible sets, we exploit this hidden convexity property by approximating the feasible set X by various ellipsoidal sets, see Section 4. In some specific cases, particularly when X is a polytope, we also use a circumscribing box approximation (see Section 4.3). The full algorithm MCF (Maximizing Convex Functions) we suggest is concisely described in Section 5. Section 6 contains numerical experiments on various randomly generated instances as well as instances from the literature.

1.2 Related Literature

Convex maximization problems are found in many real-life applications. Cost minimization problems with cost functions subject to economies of scale and binary linear optimization problems can be formulated as convex maximization problems for example (Zwart, 1974). There are also multiple examples of convex maximization problems that can be found in machine learning, such as misclassification minimization and feature selection (Mangasarian, 1996), as well as nonnegative and sparse principal component analysis (Zass and Shashua, 2007).

Additionally, convex maximization problems naturally appear in a popular approach that solves difference of convex functions (DC) optimization problems. Specifically, the convex-concave method iteratively solves convex minimization and maximization problems (Lipp and Boyd, 2016). DC optimization is used to solve a variety of problems in biology, security, machine learning and many other applications (Le Thi and Dinh, 2018). A different field in which convex maximization problems appear naturally is that of Robust Optimization. Specifically, finding the worst-case scenario for a constraint that is convex in the uncertain parameter is a convex maximization problem.

As convex maximization is NP-hard even in simple cases, there is much research devoted to the approximation of convex maximization problems. Pardalos and Rosen (1986) provide an overview of such traditional methods until the 1980s. Most such methods use linear approximations to bound the objective function from below and come with the disadvantage that the size of the subproblem that is to be solved each iteration grows. The most commonly studied version of convex maximization is that of quadratic maximization. To the best of our knowledge, essentially all methods to solve quadratic maximization problems are based on either cutting plane techniques, iterative numerical methods or decomposition of the feasible set (Audet et al., 2005; Andrianova et al., 2016). More recently, techniques from robust optimization have been used to approximate convex maximization problems (Selvi et al., 2020). In the context of DC optimization, Lipp and Boyd (2016) provide a comprehensive overview of the developed convex maximization approaches. Such approaches often rely on branch and bound or cutting plane methods, which are unable to handle large problems in reasonable time.

2 Phase 2: An Alternating Direction Method

To reformulate (P) into a biconvex optimization problem, we use the fact that, since f is closed and convex, it holds that $f = f^{**}$. Recall that f^* is the convex conjugate of f, defined as

$$f^*(\boldsymbol{y}) = \sup_{\boldsymbol{x}\in\operatorname{dom} f} \left\{ \boldsymbol{y}^\top \boldsymbol{x} - f(\boldsymbol{x}) \right\}.$$

This means that (P) is equivalent to the following problem:

$$\max_{\boldsymbol{x}\in X} \sup_{\boldsymbol{y}\in\operatorname{dom} f^*} \left\{ \boldsymbol{y}^\top \boldsymbol{x} - f^*(\boldsymbol{y}) \right\}. \tag{1}$$

Note that $\boldsymbol{y}^\top \boldsymbol{x} - f^*(\boldsymbol{y})$ is concave separately in $\boldsymbol{x}$ and $\boldsymbol{y}$, i.e., (P) is equivalent to a biconvex optimization problem. Such a problem is commonly solved with an alternating direction method such as the one in (Beck, 2017).

Algorithm 1 Alternating Direction Method

```
Input: Threshold $\epsilon$, initial value $\boldsymbol{x}^0$
Initialization: Iteration counter $k \leftarrow 0$
repeat
    Find optimal $\boldsymbol{y}$: $\boldsymbol{y}^k \in \text{argmax}_{\boldsymbol{y}\in\text{dom}\, f^*} \{\boldsymbol{y}^\top \boldsymbol{x}^k - f^*(\boldsymbol{y})\}$
    Find optimal $\boldsymbol{x}$: $\boldsymbol{x}^{k+1} \in \text{argmax}_{\boldsymbol{x}\in X} \left\{{\boldsymbol{y}^k}^\top \boldsymbol{x}\right\}$
    Update iteration counter $k \leftarrow k+1$
until $\|\boldsymbol{x}^k - \boldsymbol{x}^{k-1}\| \le \epsilon$
return $\boldsymbol{x}^k$
```

Step 4 in Algorithm 1 in which a new iterate $\boldsymbol{y}^k$ is computed has a closed-form solution.

Lemma 1. *The optimization problem*

$$\max_{\boldsymbol{y}\in\text{dom}\, f^*} \left\{\boldsymbol{y}^\top \boldsymbol{x}^k - f^*(\boldsymbol{y})\right\} \tag{2}$$

is solved by

$$\boldsymbol{y}^k = \nabla f(\boldsymbol{x}^k).$$

Proof. Since $\boldsymbol{y}^\top \boldsymbol{x}^k - f^*(\boldsymbol{y})$ is a concave function, any point that satisfies its first-order condition is a global maximum. We find

$$\mathbf{0} \in \frac{\partial}{\partial \boldsymbol{y}} \left(\boldsymbol{y}^\top \boldsymbol{x}^k - f^*(\boldsymbol{y})\right) \iff \boldsymbol{x}^k \in \partial f^*(\boldsymbol{y}),$$

which is equivalent to

$$\boldsymbol{x}^k \in \partial f^*(\boldsymbol{y}) \iff \boldsymbol{y} = \nabla f(\boldsymbol{x}^k).$$

This last equivalence follows from the fact that for any closed and convex function $\boldsymbol{y} \in \partial f(\boldsymbol{x}) \iff \boldsymbol{x} \in \partial f^*(\boldsymbol{y})$, and that f is continuously differentiable. This concludes the proof. □

This result is not only useful for computational purposes, it also allows us to show that the sequence of function values of the iterates $\boldsymbol{x}^k$ is increasing.

Lemma 2. *For any $k = 0, 1, \ldots$, it holds that*

$$f(\boldsymbol{x}^{k+1}) \ge f(\boldsymbol{x}^k).$$

Moreover, if f is strictly convex, it holds that

$$f(\boldsymbol{x}^{k+1}) > f(\boldsymbol{x}^k) \iff \boldsymbol{x}^{k+1} \neq \boldsymbol{x}^k.$$

Proof. First of all, note that by the gradient inequality of a convex function:

$$f(\boldsymbol{x}^{k+1}) - f(\boldsymbol{x}^k) \geq \nabla f(\boldsymbol{x}^k)^\top(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k) = \nabla f(\boldsymbol{x}^k)^\top \boldsymbol{x}^{k+1} - \nabla f(\boldsymbol{x}^k)^\top \boldsymbol{x}^k. \tag{3}$$

Recall that $\boldsymbol{x}^{k+1}$ is defined as

$$\boldsymbol{x}^{k+1} = \operatorname{argmax}_{\boldsymbol{x}\in X} \left\{ {\boldsymbol{y}^k}^\top \boldsymbol{x} - f^*(\boldsymbol{y}^k) \right\},$$

and thus we find

$$\nabla f(\boldsymbol{x}^k)^\top \boldsymbol{x}^{k+1} \geq \nabla f(\boldsymbol{x}^k)^\top \boldsymbol{x}^k.$$

This, together with (3), implies $f(\boldsymbol{x}^{k+1}) \geq f(\boldsymbol{x}^k)$.

Moreover, if f is strictly convex, the inequality in (3) is strict if and only if $\boldsymbol{x}^{k+1} \neq \boldsymbol{x}^k$, which demonstrates the second part of the statement. □

Additionally, we show that if the algorithm is stuck, i.e., $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k$, this solution is a local maximum.

Lemma 3. *Suppose the alternating direction method is stuck, that is, $\boldsymbol{x}^k$ is the unique solution to*

$$\max_{\boldsymbol{x}\in X} \left\{ {\boldsymbol{y}^k}^\top \boldsymbol{x} - f^*(\boldsymbol{y}^k) \right\}. \tag{4}$$

Then $\boldsymbol{x}^k$ is a local maximum of (P).

Proof. First of all, note that by Lemma 1, we know that $\boldsymbol{y}^k = \nabla f(\boldsymbol{x}^k)$. Since $\boldsymbol{x}^k$ is the unique solution to (4), we know that

$$\begin{aligned} & \nabla f(\boldsymbol{x}^k)^\top \boldsymbol{x}^k > \nabla f(\boldsymbol{x}^k)^\top (\boldsymbol{x}^k + \boldsymbol{\epsilon}) && \forall \boldsymbol{\epsilon} : \boldsymbol{x}^k + \boldsymbol{\epsilon} \in X \\ \iff & \boldsymbol{\epsilon}^\top \nabla f(\boldsymbol{x}^k) < 0 && \forall \boldsymbol{\epsilon} : \boldsymbol{x}^k + \boldsymbol{\epsilon} \in X. \end{aligned}$$

Moreover, since f is continuously differentiable, this also means that

$$\boldsymbol{\epsilon}^\top \nabla f(\boldsymbol{x}^k + \boldsymbol{\epsilon}) < 0,$$

for sufficiently small $\boldsymbol{\epsilon}$ such that $\boldsymbol{x}^k + \boldsymbol{\epsilon} \in X$. By the gradient inequality for convex functions we find

$$f(\boldsymbol{x}^k) \geq f(\boldsymbol{x}^k + \boldsymbol{\epsilon}) - \boldsymbol{\epsilon}^\top \nabla f(\boldsymbol{x}^k + \boldsymbol{\epsilon}) \iff f(\boldsymbol{x}^k) > f(\boldsymbol{x}^k + \boldsymbol{\epsilon}),$$

for all $\boldsymbol{\epsilon}$ sufficiently small and such that $\boldsymbol{x}^k + \boldsymbol{\epsilon} \in X$. Hence, $\boldsymbol{x}^k$ is a local maximum of (P). □

These results allow us to prove that the alternating direction algorithm converges to a local maximum of (P). The proof of Theorem 1 is based on the proof of theorem 14.3 in Beck (2017).

Theorem 1. *Suppose the level sets of f are bounded, i.e., for any $\alpha \in \mathbb{R}$, the set $\operatorname{Lev}(f, \alpha) = \{\boldsymbol{x} \in \mathbb{R}^n \mid f(\boldsymbol{x}) \leq \alpha\}$ is bounded. Moreover, suppose that f is bounded on X, and* (4) *has a unique solution for all $\boldsymbol{y} \in \operatorname{dom} f^*$. Then, the alternating direction method converges to a local maximum of* (P).

Proof. Given a starting point $\boldsymbol{x}^0$, let $\{\boldsymbol{x}^k\}_{k\geq 0}$ be the sequence of solutions generated by the alternating direction method. By Lemma 2, we know that $f(\boldsymbol{x}^{k+1}) \geq f(\boldsymbol{x}^k)$. Therefore, $\{f(\boldsymbol{x}^k)\}_{k\geq 0}$ is an increasing sequence that is bounded from above by assumption, and hence is convergent. Denote its limit by $\bar{f} = \sup_{k\geq 0} f(\boldsymbol{x}^k)$.

This implies that $\{\boldsymbol{x}^k\}_{k\geq 0} \subseteq \operatorname{Lev}(f, \bar{f})$, and thus it is a bounded sequence. Let $\bar{\boldsymbol{x}}$ be a limit point of $\{\boldsymbol{x}^k\}_{k\geq 0}$, i.e., the limit of a convergent subsequence $\{\boldsymbol{x}^{k_j}\}_{j\geq 0}$. Similarly, it follows that $\{\boldsymbol{x}^{k_j+1}\}_{j\geq 0}$, by potentially passing to a subsequence, converges to some vector $\boldsymbol{v} \in X$. From the definition of the algorithm, we know that

$$\begin{aligned} & \nabla f(\boldsymbol{x}^{k_j})^\top \boldsymbol{x}^{k_j+1} - f^*(\nabla f(\boldsymbol{x}^{k_j})) \geq \nabla f(\boldsymbol{x}^{k_j})^\top \boldsymbol{x} - f^*(\nabla f(\boldsymbol{x}^{k_j})) && \forall \boldsymbol{x} \in X \\ \iff & \nabla f(\boldsymbol{x}^{k_j})^\top \boldsymbol{x}^{k_j+1} \geq \nabla f(\boldsymbol{x}^{k_j})^\top \boldsymbol{x} && \forall \boldsymbol{x} \in X. \end{aligned}$$

Taking the limit $j \to \infty$ and using that f is continuously differentiable, this means

$$\nabla f(\bar{\boldsymbol{x}})^\top \boldsymbol{v} \geq \nabla f(\bar{\boldsymbol{x}})^\top \boldsymbol{x} \qquad \forall \boldsymbol{x} \in X.$$

Since $\{f(\boldsymbol{x}^{k_j})\}_{j\geq 0}$ and $\{f(\boldsymbol{x}^{k_j+1})\}_{j\geq 0}$ converge to the same value $\bar{f}$, it follows that $f(\bar{\boldsymbol{x}}) = f(\boldsymbol{v})$. Using the gradient inequality for convex functions we find

$$f(\boldsymbol{v}) - f(\bar{\boldsymbol{x}}) \geq \nabla f(\bar{\boldsymbol{x}})^\top (\boldsymbol{v} - \bar{\boldsymbol{x}}) \iff \nabla f(\bar{\boldsymbol{x}})^\top \bar{\boldsymbol{x}} \geq \nabla f(\bar{\boldsymbol{x}})^\top \boldsymbol{v},$$

which implies

$$\nabla f(\bar{\boldsymbol{x}})^\top \bar{\boldsymbol{x}} \geq \nabla f(\bar{\boldsymbol{x}})^\top \boldsymbol{x} \qquad \forall \boldsymbol{x} \in X.$$

Therefore,

$$\bar{\boldsymbol{x}} = \operatorname{argmax}_{\boldsymbol{x}\in X} \left\{ \nabla f(\bar{\boldsymbol{x}})^\top \boldsymbol{x} - f^*(\nabla f(\bar{\boldsymbol{x}})) \right\}.$$

By Lemma 3 we thus know that $\bar{\boldsymbol{x}}$ is a local maximum of (P), which concludes the proof. □

When X is a polytope, the alternating direction method could be equipped with an alternative stopping criterion not based on the threshold ϵ. Specifically, suppose the feasible set is given by

$$X = \left\{ \boldsymbol{x} \in \mathbb{R}^n_+ \;\middle|\; \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \right\},$$

for some $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{b} \in \mathbb{R}^m$, $m < n$. The subproblem to determine the next iterate of $\boldsymbol{x}$, given by

$$\max_{\boldsymbol{x}\in X} \left\{ \nabla f(\boldsymbol{x}^k)^\top \boldsymbol{x} \right\}, \tag{5}$$

is thus a linear program and only a minor perturbation from the linear program solved in the previous iteration:

$$\max_{\boldsymbol{x}\in X} \left\{ \nabla f(\boldsymbol{x}^{k-1})^\top \boldsymbol{x} \right\}. \tag{6}$$

We denote with the subscript B the elements of $\boldsymbol{x}^k$ that are nonzero, i.e., that are in the basis, and with the subscript N the nonbasic components. Similarly, we use these subscripts for the corresponding components of $\nabla f(\boldsymbol{x}^{k-1})$, and we denote the matrices consisting of the columns of $\boldsymbol{A}$ that correspond to the basic and nonbasic variables by $\boldsymbol{B}$ and $\boldsymbol{N}$, respectively. Let $\boldsymbol{r}$ denote the reduced costs of (6), given by

$$\boldsymbol{r}_N^\top = \nabla f(\boldsymbol{x}^{k-1})_B^\top \boldsymbol{B}^{-1}\boldsymbol{N} - \nabla f(\boldsymbol{x}^{k-1})^\top.$$

Then, we know from linear optimization theory that the alternating direction method is stuck, that is, $\boldsymbol{x}^k$ is also a solution to (5), if the perturbation in the objective coefficients, here given by $\nabla f(\boldsymbol{x}^k) - \nabla f(\boldsymbol{x}^{k-1})$, satisfies

$$\begin{aligned} \nabla f(\boldsymbol{x}^k)_N - \nabla f(\boldsymbol{x}^{k-1})_N &\leq \boldsymbol{r}_N \\ \boldsymbol{r}_N + \left(\nabla f(\boldsymbol{x}^k) - \nabla f(\boldsymbol{x}^{k-1})\right)^\top \boldsymbol{B}^{-1}\boldsymbol{N} &\geq \boldsymbol{0}. \end{aligned}$$

Moreover, when $\boldsymbol{r}_N > 0$, we know that $\boldsymbol{x}^k$ is the unique solution to (6), and therefore, by Lemma 3, we know that it is a local maximum.

3 Phase 1: Initialization

Because the alternating direction method is deterministic, the quality of the final solution, i.e., the objective value of the local maximum this method converges to, is fully dependent on the initialization of the algorithm. Throughout this section, we discuss a variety of methods to initialize the algorithm. This initialization is generally done by providing an initial starting point $\boldsymbol{x}^0$, which is the output of phase 1. An alternative output of phase 1 is a direction $\boldsymbol{y}^0$, for example from the direction method explained in Section 3.3, for which line 4 of the first iteration of alternating direction should be skipped.

3.1 Distance

The main idea we use is based on the intuition that the global maximum of f over X is likely to be far from $\bar{\boldsymbol{x}}$, the minimum of f. Figure 1 illustrates that idea in a 1-dimensional setting. Of course, this intuition is not true in general (for the moment, Figure 1 provides a counterexample), however it does generally provide us with a reasonably good starting point. We also remark that this intuition is only valid because the objective function is convex. The quality of this initial solution is discussed in more detail in Section 6.1.

First, it is necessary to decide on a proper distance in talking about the "furthest point". It is tempting to simply use the Euclidean distance. We suggest, however, to use a Hessian distance between a point $\boldsymbol{x}$ and the minimizer $\bar{\boldsymbol{x}}$:

$$d(\boldsymbol{x}, \bar{\boldsymbol{x}}) = (\boldsymbol{x} - \bar{\boldsymbol{x}})^\top \nabla^2 f(\bar{\boldsymbol{x}})(\boldsymbol{x} - \bar{\boldsymbol{x}}).$$

This is appealing as it incorporates some information on the function that is to be maximized. In fact, in the example shown in Figure 1c, described in detail below, x^*, the maximizer of f, is the furthest point from $\bar{x}$ in terms of this Hessian based distance.

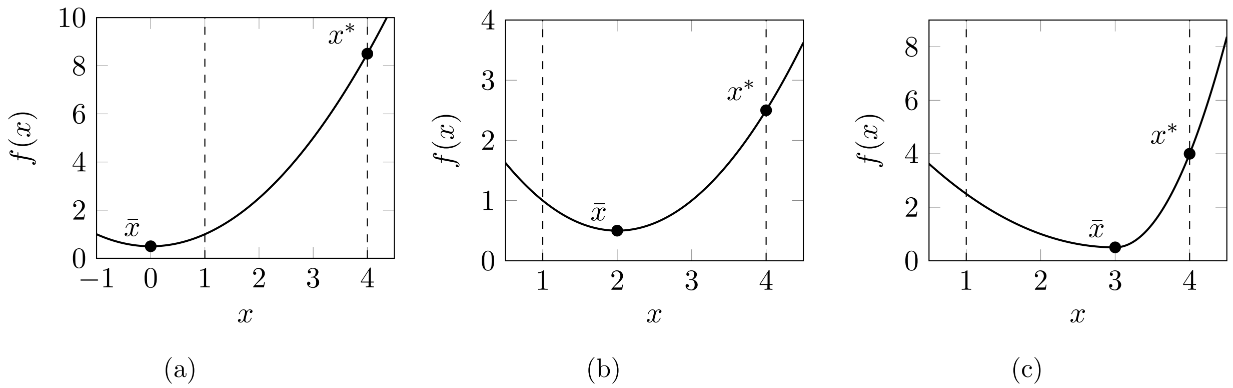

Figure 1: Three 1-dimensional examples of convex functions with a different global minimum $\bar{x}$. In two out of three examples the maximum over $X = [1, 4]$, x^*, is the furthest point in X from $\bar{x}$ in terms of Euclidean distance.

Example 1. *Consider the example in Figure 1c, for which the objective function $f : \mathbb{R} \to \mathbb{R}$ is given by*

$$f(x) = \begin{cases} \frac{1}{2}x^2 - 3x + 5 & \textit{if } x \leq 3 \\ \frac{7}{2}x^2 - 21x + 32 & \textit{if } x > 3, \end{cases}$$

and note that f is continuously differentiable. We consider the feasible set $X = [1, 4]$, from which it follows that the maximum of f over X is attained by $x^* = 4$. The minimum of f is attained by $\bar{x} = 3$, in which the Hessian of f is undefined. We do know, however, that

$$\nabla^2 f(x) = \begin{cases} 1 & \textit{if } x < 3 \\ 7 & \textit{if } x > 3, \end{cases}$$

and hence the Hessian based distance is given by

$$d(x, \bar{x}) = \begin{cases} (x - \bar{x})^2 & \textit{if } x < 3 \\ 7(x - \bar{x})^2 & \textit{if } x > 3, \end{cases}$$

for which it is clear that it is maximized by $x^* = 4$.

So far, we have implicitly assumed $\bar{\boldsymbol{x}}$ to refer to the minimum of f over its domain, in particular that $\nabla f(\bar{\boldsymbol{x}}) = \mathbf{0}$. It is not guaranteed however, that f attains the minimum over its domain. Therefore, we additionally consider the *constrained minimum of f over X* and also initialize the algorithm with the furthest point from this constrained minimum. We discuss using the constrained minimum in more detail in Section 3.4.

Besides the intuition outlined above, there is significant theoretical merit to the idea of initializing the algorithm at the furthest point from the minimum point as well. Indeed, for a strictly convex quadratic function f, the furthest point from the global minimum in terms of the Hessian based distance is exactly the global maximum of f.

Theorem 2. *Let f be a strictly convex quadratic function, i.e.,*

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}^\top \boldsymbol{x},$$

for some $\boldsymbol{A} \in \mathbb{S}^n_{++}, \boldsymbol{b} \in \mathbb{R}^n$. Then the furthest point from $\bar{\boldsymbol{x}}$ in terms of d is the global maximum of f over X:

$$\text{argmax}_{\boldsymbol{x}\in X}\, f(\boldsymbol{x}) = \text{argmax}_{\boldsymbol{x}\in X}\, d(\boldsymbol{x}, \bar{\boldsymbol{x}}).$$

Proof. Note that the unconstrained global minimum of f, $\bar{\boldsymbol{x}}$ is such that

$$\nabla f(\bar{\boldsymbol{x}}) = 0 \iff \boldsymbol{A}\bar{\boldsymbol{x}} + \boldsymbol{b} = 0 \iff \bar{\boldsymbol{x}} = -\boldsymbol{A}^{-1}\boldsymbol{b}.$$

Therefore, we find

$$\begin{aligned} d(\boldsymbol{x}, \bar{\boldsymbol{x}}) &= (\boldsymbol{x} - \bar{\boldsymbol{x}})^\top \nabla^2 f(\bar{\boldsymbol{x}})(\boldsymbol{x} - \bar{\boldsymbol{x}}) \\ &= (\boldsymbol{x} - \bar{\boldsymbol{x}})^\top \boldsymbol{A}(\boldsymbol{x} - \bar{\boldsymbol{x}}) \\ &= \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} - 2\boldsymbol{x}^\top \boldsymbol{A}\bar{\boldsymbol{x}} + \bar{\boldsymbol{x}}^\top \boldsymbol{A}\bar{\boldsymbol{x}} \\ &= \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} + 2\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{A}^{-1}\boldsymbol{b} + \boldsymbol{b}^\top {\boldsymbol{A}^{-1}}^\top \boldsymbol{A}\boldsymbol{A}^{-1}\boldsymbol{b} \\ &= 2f(\boldsymbol{x}) + \boldsymbol{b}^\top {\boldsymbol{A}^{-1}}^\top \boldsymbol{b}, \end{aligned}$$

that is, d is a scalar multiple of f plus a constant. Therefore, they are maximized by the same $\boldsymbol{x}$. □

Similar to this result, it also follows that for general convex f maximizing this Hessian based distance finds the maximum over X of the second-order Taylor expansion of f in $\bar{\boldsymbol{x}}$.

Theorem 3. *Assume the minimum of f over its domain is attained in a stationary point $\bar{\boldsymbol{x}}$. Let $t_2^f(\boldsymbol{x}; \bar{\boldsymbol{x}})$ denote the second-order Taylor expansion of f at $\bar{\boldsymbol{x}}$ evaluated in $\boldsymbol{x}$. Then*

$$\text{argmax}_{\boldsymbol{x}\in X}\, d(\boldsymbol{x}, \bar{\boldsymbol{x}}) = \text{argmax}_{\boldsymbol{x}\in X}\, t_2^f(\boldsymbol{x}, \bar{\boldsymbol{x}}).$$

Proof. Note that the second order Taylor expansion of f at $\bar{\boldsymbol{x}}$ is given by

$$\begin{aligned} t_2^f(\boldsymbol{x}; \bar{\boldsymbol{x}}) &= f(\bar{\boldsymbol{x}}) + (\boldsymbol{x} - \bar{\boldsymbol{x}})^\top \nabla f(\bar{\boldsymbol{x}}) + \frac{1}{2}(\boldsymbol{x} - \bar{\boldsymbol{x}})^\top \nabla^2 f(\bar{\boldsymbol{x}})(\boldsymbol{x} - \bar{\boldsymbol{x}}) \\ &= f(\bar{\boldsymbol{x}}) + (\boldsymbol{x} - \bar{\boldsymbol{x}})^\top \nabla f(\bar{\boldsymbol{x}}) + \frac{1}{2}d(\boldsymbol{x}, \bar{\boldsymbol{x}}) \\ &= f(\bar{\boldsymbol{x}}) + (\boldsymbol{x} - \bar{\boldsymbol{x}})^\top \boldsymbol{0} + \frac{1}{2}d(\boldsymbol{x}, \bar{\boldsymbol{x}}) \\ &= f(\bar{\boldsymbol{x}}) + \frac{1}{2}d(\boldsymbol{x}, \bar{\boldsymbol{x}}), \end{aligned}$$

where the second equality holds by the definition of d, while the third equality holds by the first-order condition of $\bar{\boldsymbol{x}}$ being the minimum of f. This shows that t_2^f is the sum of a constant and a positive scalar multiple of d, and hence they are maximized by the same vectors. □

Inspired by this last result, we in fact suggest to maximize the second-order Taylor expansion instead of the Hessian based distance d. When using the global minimum, Theorem 3 shows that this is the exact same. For the constrained minimum, on the other hand, this means there is an additional linear term in the Taylor expansion: $(\boldsymbol{x} - \bar{\boldsymbol{x}})^\top \nabla f(\bar{\boldsymbol{x}})$. This term is guaranteed to be nonnegative due to the necessary and sufficient conditions for $\bar{\boldsymbol{x}}$ to be the constrained minimum of the convex function f over X, see Lemma 4 in Section 3.4. This means it is advantageous to maximize the Taylor expansion at the *constrained minimum*.

We conclude this discussion on distance with a small example that illustrates the potential benefit of using the second-order Taylor expansion over the Euclidean distance.

Example 2. *We consider a two-dimensional example with*

$$f(\boldsymbol{x}) = e^{(2x_1 - x_2)^2} + x_1^2 + x_2^2 - 4x_1 - 4x_2,$$

and

$$X = \left\{\boldsymbol{x} \in \mathbb{R}^2 \;\middle|\; 0 \leq x_1 \leq 1, -2 \leq x_2 \leq 3\right\},$$

which is problem (P3) from (Enhbat, 1996). First, we note that the gradient and Hessian of f are given by

$$\nabla f(\boldsymbol{x}) = 2 \begin{bmatrix} 2 \\ -1 \end{bmatrix} (2x_1 - x_2) e^{(2x_1 - x_2)^2} + 2\boldsymbol{x} - \begin{bmatrix} 4 \\ 4 \end{bmatrix}$$

$$\nabla^2 f(\boldsymbol{x}) = \begin{bmatrix} 16(2x_1 - x_2)^2 e^{(2x_1-x_2)^2} + 8e^{(2x_1-x_2)^2} + 2 & -8(2x_1 - x_2)^2 e^{(2x_1-x_2)^2} - 4e^{(2x_1-x_2)^2} \\ -8(2x_1 - x_2)^2 e^{(2x_1-x_2)^2} - 4e^{(2x_1-x_2)^2} & 4(2x_1 - x_2)^2 e^{(2x_1-x_2)^2} + 2e^{(2x_1-x_2)^2} + 2 \end{bmatrix}.$$

Then, we find the minimum of f over X:

$$\bar{\boldsymbol{x}} \approx \begin{bmatrix} 1.323 \\ 2.338 \end{bmatrix}.$$

Thus, based on the Euclidean distance, the starting point would be given by $\boldsymbol{x}^0 = [0 \quad -2]^\top$, as it is the furthest from $\bar{\boldsymbol{x}}$. The alternating direction method would then subsequently find

$$\boldsymbol{y}^0 = \begin{bmatrix} 8e^4 - 4 \\ -4e^4 - 8 \end{bmatrix}, \ \boldsymbol{x}^1 = \begin{bmatrix} 1 \\ -2 \end{bmatrix}, \ \boldsymbol{y}^1 = \begin{bmatrix} 16e^{16} - 2 \\ -8e^{16} - 6 \end{bmatrix}, \ \boldsymbol{x}^2 = \boldsymbol{x}^1,$$

which is the global optimum of f over X.

If, on the other hand, we use the second-order Taylor expansion, we first find that

$$\nabla^2 f(\bar{\boldsymbol{x}}) = \begin{bmatrix} 10 & -4 \\ -4 & 4 \end{bmatrix},$$

and from that find that

$$\mathrm{argmax}_{\boldsymbol{x} \in X}\, d(\boldsymbol{x}, \bar{\boldsymbol{x}}) = \begin{bmatrix} 1 \\ -2 \end{bmatrix},$$

which is the global maximum. Hence, when using the second-order Taylor expansion, the initial starting point provided to the alternating direction method is already globally optimal.

3.2 Finding the Furthest Point

While starting at the furthest point from the global minimum is intuitively very appealing, it is not computationally tractable in general. In fact, one still has to maximize a convex function over the original feasible set. The main difference with the original problem, however, is that regardless of f, we are now tasked with maximizing a convex quadratic. We first discuss the case where the feasible set is an ellipsoid, for which the problem reduces to a regular convex optimization problem due to hidden convexity. Then we discuss how these results can be used to initialize the algorithm for non-ellipsoidal feasible sets.

For now, let X be a full dimensional ellipsoid, i.e.,

$$X = \{\boldsymbol{Q}\boldsymbol{u} + \boldsymbol{q} \mid \|\boldsymbol{u}\|_2 \leq 1\},$$

for some $\boldsymbol{Q} \in \mathcal{S}^n_{++}$ and $\boldsymbol{q} \in \mathbb{R}^n$. In this case, we wish to solve the problem

$$\max_{\boldsymbol{x}} \ (\boldsymbol{x} - \bar{\boldsymbol{x}})^\top \nabla^2 f(\bar{\boldsymbol{x}})(\boldsymbol{x} - \bar{\boldsymbol{x}}) \tag{7a}$$

$$\text{s.t.} \ \left\|\boldsymbol{Q}^{-1}(\boldsymbol{x} - \boldsymbol{q})\right\|_2 \leq 1. \tag{7b}$$

The optimal solution to this optimization problem can be obtained from solving a convex optimization problem by using the approach by Ben-Tal and Den Hertog (2014) if the matrices $\nabla^2 f(\bar{\boldsymbol{x}})$ and $\boldsymbol{Q}^{-1}\boldsymbol{Q}^{-1}$ are simulatenously diagonalizable. Since we assume $\boldsymbol{Q}$ to be positive definite, this is always the case, and thus (7) can be solved efficiently.

Specifically, let $\boldsymbol{Q}^{-1}\boldsymbol{Q}^{-1} = \boldsymbol{L}\boldsymbol{L}^\top$ be the Cholesky factorization of $\boldsymbol{Q}^{-1}\boldsymbol{Q}^{-1}$, and define $\boldsymbol{A} = \boldsymbol{L}^{-1}\nabla^2 f(\bar{\boldsymbol{x}})\boldsymbol{L}^{-\top}$. Then, compute the Schur decomposition $\boldsymbol{U}^\top\boldsymbol{A}\boldsymbol{U} = \text{diag}(\boldsymbol{\alpha})$ and set $\boldsymbol{S} = \boldsymbol{L}^{-\top}\boldsymbol{U}$. Then, $\boldsymbol{S}$ diagonalizes both $\boldsymbol{Q}^{-1}\boldsymbol{Q}^{-1}$ and $\nabla^2 f(\bar{\boldsymbol{x}})$, since $\boldsymbol{S}^\top\boldsymbol{Q}^{-1}\boldsymbol{Q}^{-1}\boldsymbol{S} = \boldsymbol{I}$ and $\boldsymbol{S}^\top\nabla^2 f(\bar{\boldsymbol{x}})\boldsymbol{S} = \text{diag}(\boldsymbol{\alpha})$. By using the one-to-one change of variables $\boldsymbol{x} = \boldsymbol{S}\boldsymbol{z}$, and defining $\boldsymbol{\beta} = \boldsymbol{S}^\top\nabla^2 f(\bar{\boldsymbol{x}})\bar{\boldsymbol{x}}$ and $\boldsymbol{\delta} = -2\boldsymbol{S}^\top\boldsymbol{Q}^{-1}\boldsymbol{q}$, problem (7) is reduced to

$$\min_{\boldsymbol{z}} \ \sum_i -\frac{1}{2}\alpha_i z_i^2 + \beta_i z_i \tag{8a}$$

$$\text{s.t.} \ \sum_i z_i^2 + \delta_i z_i \leq 1. \tag{8b}$$

Introducing auxiliary variables $y_i = \frac{1}{2}z_i^2$ for $i = 1, \dots, n$ and further relaxing the equality constraint we end up with the following convex relaxation of (8):

$$\min_{\boldsymbol{z},\boldsymbol{y}} \ -\boldsymbol{\alpha}^\top\boldsymbol{y} + \boldsymbol{\beta}^\top\boldsymbol{z} \tag{9a}$$

$$\text{s.t.} \ 2 \cdot \boldsymbol{1}^\top\boldsymbol{y} + \boldsymbol{\delta}^\top\boldsymbol{z} \leq 1 \tag{9b}$$

$$\frac{1}{2}z_i^2 - y_i \leq 0 \qquad i = 1, \dots, n. \tag{9c}$$

An optimal solution $(\boldsymbol{z}^*, \boldsymbol{y}^*)$ to (9) is an optimal solution to (8) if some regularity conditions hold. Then problem (9) has a solution such that for all i it holds that $y_i = \frac{1}{2}z_i^2$, i.e., this $\boldsymbol{z}^*$ solves (8). Subsequently, an optimal solution to (7) is obtained by applying the change of variables, i.e., $\boldsymbol{x} = \boldsymbol{S}\boldsymbol{z}$. We remark that the same result holds when X is defined by a two-sided quadratic constraint:

$$\alpha \leq \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}^\top \boldsymbol{x} \leq \beta.$$

For more details, we refer to (Ben-Tal and Den Hertog, 2014).

3.3 Computing the Furthest Feasible Point

When X is not an elllipsoid, finding the furthest point from $\bar{\boldsymbol{x}}$ is in general hard. In such cases, we choose to approximate X and find the furthest point in this approximation. Specifically, we use ellipsoidal approximations and a circumscribing box that are described in more detail in Section 4. Of course, using this approximation of the furthest point as an initial value does not come with the attractive theoretical properties described above. Therefore, next to using this approximate furthest point, we propose two alternative methods that attempt to find initial values that lead the algorithm to converge to the global maximum. For now, we assume we have some approximation of X, denoted by X', for which we can find the furthest point from $\bar{\boldsymbol{x}}$ efficiently. This approximate furthest point will be denoted by $\hat{\boldsymbol{x}}$.

Line Method

The first of the suggested initial values is found by constructing the line through $\bar{\boldsymbol{x}}$ and $\hat{\boldsymbol{x}}$ and subsequently setting $\boldsymbol{x}^0$ to the furthest of the two intersection points of this line and the boundary of X. This method will be referred to as the *line* method throughout the rest of the paper. When X is a polytope, these intersection points can be found efficiently through closed-form formulas. For other feasible sets, bisection can be used, given that the number of constraints is sufficiently small. For a higher number of constraints, one can use the (actually optimal) procedure proposed by (Gal et al., 1978; Ben-Tal and Nemirovski, 2001).

Direction Method

The second alternative we propose is an initial value for $\boldsymbol{y}$. Specifically, we suggest starting with $\boldsymbol{y}^0 = \hat{\boldsymbol{x}} - \bar{\boldsymbol{x}}$ and skip the first half of the first iteration, that is, find

$$\boldsymbol{x}^1 \in \operatorname{argmax}_{\boldsymbol{x} \in X} \left\{ (\hat{\boldsymbol{x}} - \bar{\boldsymbol{x}})^\top \boldsymbol{x} \right\}. \tag{10}$$

This method will be referred to as the *direction* method throughout the rest of the paper. It is, in fact, guaranteed in this alternative start that $\boldsymbol{x}^1$ is farther from $\bar{\boldsymbol{x}}$ than the initial value obtained from the *line* method.

Strictly from a distance point of view, this would indicate that the *direction* method should yield the best initial value. However, recall that for non-quadratic objective functions, starting at the furthest point is no guarantee. Indeed, in numerical experiments we find that using $\hat{\boldsymbol{x}}$ as $\boldsymbol{x}^0$ or using the initial values provided by the *line* and *direction* methods can all lead to better final solutions than the others. Therefore, from here on out, we always simply initialize the phase 2 algorithm with all options to maximize the likelihood of finding the global maximum.

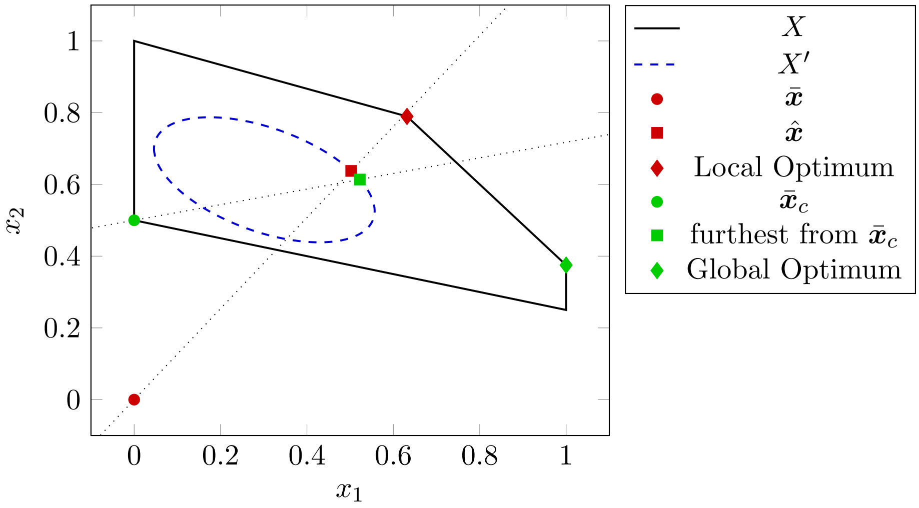

Figure 2: Illustration of initalization based on an inner ellipsoidal approximation for the unconstrained and constrained minimum for (11) from Example 3.

3.4 Furthest Point from the Constrained Minimum

So far, we have mainly considered initializing the algorithm as far as possible from the unconstrained global minimum. This is not necessarily always the best approach, however. Besides the fact that not all convex functions attain their minimum over their domain, using the constrained minimum, i.e., the minimum of f over X can yield a better final solution. This is illustrated in the following example.

Example 3. *We consider maximizing a quadratic function over a polytope:*

$$\max_{\boldsymbol{x}\in\mathbb{R}^2} \ \boldsymbol{x}^\top \begin{bmatrix} \frac{5}{4} & \frac{5}{6} \\ \frac{5}{6} & 1 \end{bmatrix} \boldsymbol{x} \tag{11a}$$

$$s.t. \ \frac{1}{3}x_1 + x_2 \leq 1 \tag{11b}$$

$$\frac{1}{4}x_1 + x_2 \geq \frac{1}{2} \tag{11c}$$

$$\frac{3}{4}x_1 + \frac{2}{3}x_2 \leq 1 \tag{11d}$$

$$0 \leq x_1 \leq 1. \tag{11e}$$

Figure 2 shows the feasible region together with an ellipsoidal inner approximation. It also depicts the constrained and unconstrained minimum of f together with the furthest points from them on the ellipsoid, the line connecting those two points and the solution the phase 2 algorithm converges to after initialization. In this example, initialization based on the constrained minimum of f does in fact lead to the algorithm finding the global optimum, while using the unconstrained global minimum of f does not.

We remark that for the constrained minimum, which we will denote by $\bar{\boldsymbol{x}}_c = \min_{\boldsymbol{x}\in X} f(\boldsymbol{x})$, the Hessian based distance is not equal to the second-order Taylor expansion. It is true, however, that the second-order Taylor expansion is an upper bound for this distance plus a constant. To see this, first recall the following result from analysis (Boyd and Vandenberghe, 2004, p. 139).

Lemma 4.

$$\bar{\boldsymbol{x}}_c \in \operatorname{argmin}_{\boldsymbol{x}\in X} f(\boldsymbol{x}) \qquad \Longleftrightarrow \qquad \nabla f(\bar{\boldsymbol{x}}_c)^\top(\boldsymbol{x}-\bar{\boldsymbol{x}}_c) \geq 0 \ \ \forall \boldsymbol{x} \in X.$$

Therefore, we find that

$$\begin{aligned} t_2^f(\boldsymbol{x};\bar{\boldsymbol{x}}_c) &= f(\bar{\boldsymbol{x}}_c) + (\boldsymbol{x}-\bar{\boldsymbol{x}}_c)^\top \nabla f(\bar{\boldsymbol{x}}_c) + \frac{1}{2}(\boldsymbol{x}-\bar{\boldsymbol{x}}_c)^\top \nabla^2 f(\bar{\boldsymbol{x}}_c)(\boldsymbol{x}-\bar{\boldsymbol{x}}_c) \\ &= f(\bar{\boldsymbol{x}}_c) + (\boldsymbol{x}-\bar{\boldsymbol{x}}_c)^\top \nabla f(\bar{\boldsymbol{x}}_c) + \frac{1}{2}d(\boldsymbol{x},\bar{\boldsymbol{x}}_c) \\ &\geq f(\bar{\boldsymbol{x}}_c) + \frac{1}{2}d(\boldsymbol{x},\bar{\boldsymbol{x}}_c), \end{aligned}$$

where the second equality holds by the definition of d. We note that, by Lemma 4, this reasoning only works for the constrained minimum $\bar{\boldsymbol{x}}_c$. We suggest therefore, to maximize the distance d shifted with this first-order term, i.e., the second-order Taylor expansion for the constrained minimum instead. This still yields a quadratic for which all methods discussed in this Section apply. We remark that for the values of $\boldsymbol{x}$ we are interested in, that is, those far from $\bar{\boldsymbol{x}}_c$, the quadratic term generally dominates the expression. There is an additional advantage of using the constrained minimum that originates in the use of the second-order Taylor expansion at the minimum. When the minimum in the feasible set is used, the approximation quality of the second-order Taylor expansion with respect to the original objective is generally better over the feasible set.

3.5 Random Initialization

Since the alternating direction method used in the second phase of the algorithm requires very little computation time, it is often possible to simply ‘try’ many different initial values. For the sake of comparison, we therefore also consider initializing the algorithm in a random fashion. Specifically, we draw $\boldsymbol{y}^0$ uniformly from the surface of the n-dimensional unit ball. Intuitively, this means we initialize our algorithm by maximizing over the feasible set in a uniform, random direction. We emphasize that the randomness is contained to the initialization of the algorithm, i.e., the alternating direction method is executed based on a randomly generated starting point. For problem instances whose global maximum is located in a narrow region of the feasible set, random initialization works particularly well, since converging to narrow regions of the feasible set is very likely after random initialization. Example 4 in Section 4.3 is such an example for which random initialization often finds the global maximum.

Of course, random initialization does not perform well for all instances. In particular, for larger instances the computation time needed to obtain a high quality solution through random initialization grows significantly. This can be explained by the fact that both the iterations in phase 2 of the algorithm take longer and a higher number of initial values are needed to find the global maximum. This disadvantage is investigated in more detail in Section 6.1. Furthermore, as is to be expected with any random method, this initialization method is rather unpredictable. While it seems to perform well in general, it fails to find even a reasonable solution for some instances. Table 5 in Section 6.2 contains such an example.

4 Approximations of the Feasible Set

As discussed in Section 3, we suggest approximating the feasible set in phase 1 to find the furthest point efficiently. We emphasize this approximation is only used in phase 1 to obtain a tractable problem. In phase 2, the original feasible set is used. Specifically, we discuss the maximum volume inscribed ellipsoid in Section 4.1 and inscribed and circumscribing ellipsoids around the analytic center in Section 4.2. Additionally, we discuss approximating the feasible set with a box in Section 4.3. This approximation is particularly relevant for instances where ellipsoidal approximations poorly represent the feasible set. Throughout this section, we also discuss the type of feasible sets to which the approximations are especially suited. To underline this, we present results on randomly generated instances for all these feasible set approximations, as well as random initialization as discussed in Section 3.5.

4.1 Maximum Volume Inscribed Ellipsoid

The maximum volume inscribed ellipsoid is a well-known inner approximation of a convex set that can be found by solving a convex optimization problem whenever that set is a polytope or the intersection of ellipsoids. Specifically, when X is a polytope, that is,

$$X = \left\{ \boldsymbol{x} \in \mathbb{R}^n \;\middle|\; \boldsymbol{a}_i^\top \boldsymbol{x} \leq b_i \quad i = 1, \ldots, m \right\},$$

for some $\boldsymbol{a}_i \in \mathbb{R}^n$, $b_i \in \mathbb{R}$ for $i = 1, \ldots, m$, the maximum volume inscribed ellipsoid can be found by solving (Boyd and Vandenberghe, 2004):

$$\begin{aligned} \min_{\boldsymbol{Q}, \boldsymbol{q}} \quad & \log \det \boldsymbol{Q}^{-1} \\ \text{s.t.} \quad & \|\boldsymbol{Q}\boldsymbol{a}_i\|_2 + \boldsymbol{a}_i^\top \boldsymbol{q} \leq b_i \qquad\qquad i = 1, \ldots, m. \end{aligned}$$

Let $(\boldsymbol{Q}^*, \boldsymbol{q}^*)$ denote the optimal solution of this optimization problem. The maximum volume inscribed ellipsoid is then given by

$$\mathcal{E} = \{ \boldsymbol{Q}^* \boldsymbol{u} + \boldsymbol{q}^* \mid \|\boldsymbol{u}\|_2 \leq 1 \}.$$

Whenever X is the intersection of ellipsoids, i.e.,

$$X = \left\{ \boldsymbol{x} \in \mathbb{R}^n \;\middle|\; \boldsymbol{x}^\top \boldsymbol{A}_i \boldsymbol{x} + 2\boldsymbol{b}_i^\top \boldsymbol{x} + c_i \leq 0 \quad i = 1, \ldots, m \right\},$$

the maximum volume inscribed ellipsoid is found from the solution of (Boyd and Vandenberghe,

	Objective Value				Computation Time (s)			
	Random	MVIE	AC	Box	Random	MVIE	AC	Box
# 1	132.444	132.558	**132.582**	132.274	2.16	62.98	4.81	0.84
# 2	137.705	**137.850**	137.751	137.795	1.91	63.46	4.31	0.88
# 3	132.280	**132.403**	132.402	132.249	2.18	67.88	4.73	0.85
# 4	135.762	**135.843**	135.834	135.777	2.17	56.93	4.16	0.83
# 5	134.212	**134.249**	134.210	134.212	2.18	62.66	4.37	0.86

Table 1: Objective values and computation times for five polytopic instances with $n = 100$ and $2n$ constraints and an exponential objective for the four different initialization methods: random, based on the maximum volume inscribed ellipsoid (MVIE), based on ellipsoids around the analytic center (AC) and based on a bounding box. For each instance the best found objective value is marked in bold.

2004):

$$\begin{aligned} \min_{\boldsymbol{Q},\boldsymbol{q},\boldsymbol{\lambda}} \quad & \log\det \boldsymbol{Q}^{-1} \\ \text{s.t.} \quad & \begin{bmatrix} -\lambda_i - c_i + \boldsymbol{b}_i^\top \boldsymbol{A}_i^{-1}\boldsymbol{b}_i & 0 & (\boldsymbol{q} + \boldsymbol{A}_i^{-1}\boldsymbol{b}_i)^\top \\ 0 & \lambda_i \boldsymbol{I} & \boldsymbol{Q} \\ \boldsymbol{d} + \boldsymbol{A}_i^{-1}\boldsymbol{b}_i & \boldsymbol{Q} & \boldsymbol{A}_i^{-1} \end{bmatrix} \succeq 0 \qquad i = 1, \dots, m. \end{aligned}$$

The maximum volume inscribed ellipsoid, unfortunately, can be somewhat costly to compute in higher dimensions as it requires solving a semidefinite program. Additionally, it is limited to feasible sets with a specific structure. The minimum volume inscribed ellipsoid generally performs best on polytopes that are not particularly ‘narrow’. This difference is most pronounced for exponential objectives rather than quadratic objectives. Table 1 lists the results for five of those instances with $n = 100$. Specifically, we randomly generate polytopes in n dimensions with $2n$ constraints by intersecting randomly generated halfspaces with the hyperrectangle $\{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{0} \leq \boldsymbol{x} \leq \boldsymbol{1}\}$. The $2n$ random halfspaces are generated by generating a vector $\boldsymbol{s}_j \in \mathbb{R}^n$ uniformly distributed on the surface of the hypersphere, and a scalar $r_j \in \mathbb{R}$ uniformly distributed from the interval $\left[-\frac{1}{2}\|\boldsymbol{s}_j\|_1, \frac{1}{2}\|\boldsymbol{s}_j\|_1\right]$. We then consider the constraint $\boldsymbol{s}_j^\top(\boldsymbol{x} - \frac{1}{2}\boldsymbol{1}) \leq r_j$ if $r_j > 0$ and $\boldsymbol{s}_j^\top(\boldsymbol{x} - \frac{1}{2}\boldsymbol{1}) \geq r_j$ if $r_j \leq 0$. This construction ensures that the center of the hyperrectangle, $\frac{1}{2}\boldsymbol{1}$, always remains feasible. The objective function is given by

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} e^{\alpha_i x_i} - \boldsymbol{\beta}^\top \boldsymbol{x},$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are drawn from a uniform distribution on $[0, 1]^n$.

The objective values of the best solution found by each initialization approach indeed indicates that the MVIE performs especially well for these instances. Its main downside however, is also obvious from Table 1: the high computation time required to find the MVIE. The ellipsoids around the analytic center, discussed in the next section, find solutions that are close in quality

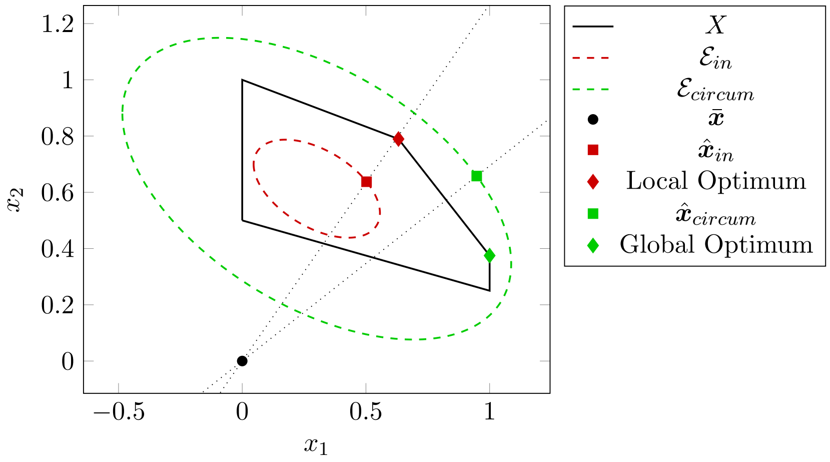

Figure 3: Illustration of the feasible region and approximating ellipsoids for (11) from Example 3.

in significantly less time. This is especially useful in higher dimensions. Because of this observation, we elect to not include the MVIE as an initialization approach in the MCF algorithm in Section 5.

4.2 Inscribed and Circumscribing Ellipsoids around the Analytic Center

Suppose X is a compact set defined by m convex inequalities, that is,

$$X = \{\boldsymbol{x} \in \mathbb{R}^n \mid g_i(\boldsymbol{x}) \leq 0 \quad i = 1, \ldots, m\}.$$

Then, the analytic center of X is defined as the minimum of the logarithmic barrier function:

$$\phi(\boldsymbol{x}) = -\sum_{i=1}^{m} \log -g_i(\boldsymbol{x}).$$

We will denote the analytic center by $\boldsymbol{x}_{ac}$. Jarre (1994) shows that the ellipsoids defined by

$$\mathcal{E}_{in} = \left\{\boldsymbol{x}_{ac} + \boldsymbol{u} \;\middle|\; \boldsymbol{u}^\top \nabla^2 \phi(\boldsymbol{x}_{ac}) \boldsymbol{u} \leq 1\right\}, \tag{12}$$

$$\mathcal{E}_{circum} = \left\{\boldsymbol{x}_{ac} + (1 + \frac{2}{\sqrt{m}}) m \boldsymbol{u} \;\middle|\; \boldsymbol{u}^\top \nabla^2 \phi(\boldsymbol{x}_{ac}) \boldsymbol{u} \leq 1\right\}, \tag{13}$$

are an inscribed and circumscribing ellipsoid of X, respectively, whenever ϕ is self-concordant. We remark that even though these ellipsoids only differ in radius and are identical in shape, they do lead to different starting points and possibly to different final solutions. We illustrate that in the following stylized example.

Example 3. *(Continued.) Figure 3 shows the feasible region of* (11)*, the analytic center and both $\mathcal{E}_{in}$ and $\mathcal{E}_{circum}$, as well as the local and global maximum to which the algorithm converges after initialization based on the inscribed and circumscribing ellipsoid, respectively. Interestingly, after initialization based on the inscribed ellipsoid the algorithm converges to a local, not global,* *optimum. After initialization based on the circumscribing ellipsoid, however, the algorithm converges to the global optimum. This example indicates therefore that there is merit to considering initial values based on both ellipsoids, even though their shape is identical.*

Since computing the analytic center is the computationally expensive part of these approximations, we suggest initialization based on both the inscribed and circumscribing ellipsoids. Additionally, we note that the *line* method cannot always be used together with the circumscribing ellipsoid, as the line through $\bar{\boldsymbol{x}}$ and $\hat{\boldsymbol{x}}$ does not necessarily intersect X in that case.

4.3 Bounding Box

For some instances, initialization based on approximating the feasible set with ellipsoids fails to find the global maximum. We first present a pathological example that illustrates the conditions under which this happens.

Example 4. *We consider the two-dimensional example*

$$\max_{\boldsymbol{x}\in\mathbb{R}^2} \ \frac{1}{2}\boldsymbol{x}^\top\boldsymbol{x} - \boldsymbol{b}^\top\boldsymbol{x} \tag{14a}$$

$$\text{s.t. } x_1 + 10x_2 \leq 10 \tag{14b}$$

$$\boldsymbol{x} \geq \boldsymbol{0}, \tag{14c}$$

for three different values of $\boldsymbol{b} \in \mathbb{R}^2$. Note that the global minimum of this problem is given by $\bar{\boldsymbol{x}} = \boldsymbol{b}$. Figure 4 shows the feasible region, the analytic center and the inscribed and circumscribing ellipsoids around it, and a circumscribing box. It also depicts all considered values of $\boldsymbol{b}$ and the furthest points from these global minima on the approximations. We consider the following three different values for $\boldsymbol{b}$:

$$\boldsymbol{b}^1 = \begin{bmatrix} 4.9 \\ 0.1 \end{bmatrix}, \qquad \boldsymbol{b}^2 = \begin{bmatrix} 4.9 \\ 0 \end{bmatrix}, \qquad \boldsymbol{b}^3 = \begin{bmatrix} 4.9 \\ -0.2 \end{bmatrix},$$

such that the global minimum of the objective function is inside, on the boundary of and outside of the feasible set, respectively. We remark that for all these values of $\boldsymbol{b}$, the optimal solution to (14) *is $\boldsymbol{x}^* = \begin{bmatrix}10 & 0\end{bmatrix}^\top$.*

Because x_1 is by far the most influential of the two variables on the distance to the global minimum, the furthest points on the ellipsoids are the same for all values of $\boldsymbol{b}$ we consider: the point on the ellipsoid with the smallest x_1. Graphically, it seems that initialization based on any ellipsoid will lead the MCF algorithm to converge to $\boldsymbol{x} = \begin{bmatrix}0 & 1\end{bmatrix}^\top$. Indeed, for any method (line, direction or $\hat{\boldsymbol{x}}$, the algorithm converges to this local maximum.

Interestingly, when the alternating direction method is initialized with a random direction $\boldsymbol{y}^0$, it is reasonably likely to converge to the global maximum, as any $\boldsymbol{y}^0$ on the arc from $\begin{bmatrix}0 & -1\end{bmatrix}^\top$ to $\begin{bmatrix}0.0995 & 0.995\end{bmatrix}^\top$ on the unit circle leads to $\boldsymbol{x}^1$ as the global maximum.

This example has two key characteristics that causes initialization based on ellipsoidal approximation to fail to find the global maximum. First, the global maximum is located in a

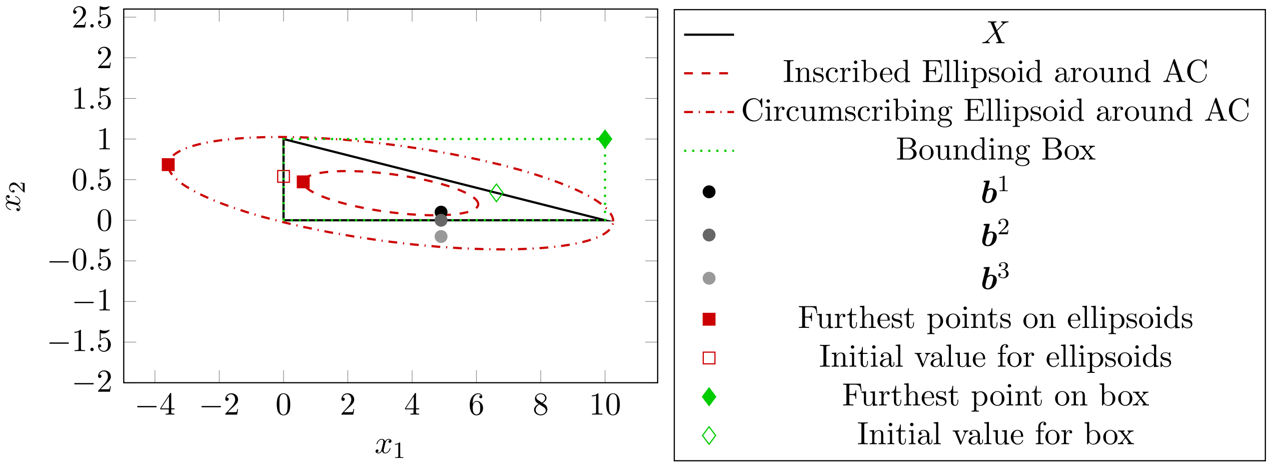

Figure 4: The feasible set and approximating ellipsoids of (14) for three different values of $\boldsymbol{b}$.

narrow area of the feasible set, at $[10\ \ 0]^\top$. Such a narrow region of the feasible set is hard to approximate well using an ellipsoid. Second, the global minimum of the objective is in some sense on the same side of the approximating ellipsoids as the global maximum. Even though the global maximum is in fact the furthest point from the global minimum in the feasible set, the ellipsoidal based initialization methods fail to converge to the global maximum as the furthest points from the global minimum on those ellipsoids are nowhere near the global maximum.

Approximating the feasible set by a different shape than an ellipsoid can help alleviate this issue. Specifically, we suggest constructing the smallest bounding box by maximizing and minimizing in each elementary direction over the feasible set. This is done by solving $\min_{\boldsymbol{x}\in X} x_i$ and $\max_{\boldsymbol{x}\in X} x_i$ for $i = 1, \dots, n$. While this box has exponentially many extreme points, 2^n to be precise, finding the furthest point from the global minimum in terms of Euclidean distance is easy: for each coordinate simply choose either the minimum or maximum value based on which one's difference to the global minimum is larger. Then, for this furthest point the same ideas as for ellipsoidal approximation can be employed: the *line* and *direction* methods as well as simply initializing $\boldsymbol{x}^0$ with the furthest point.

Example 4. *(Continued.) For the polytope considered in* (14)*:*

$$X = \left\{\boldsymbol{x} \in \mathbb{R}^2 \;\middle|\; \boldsymbol{x} \geq \boldsymbol{0},\ \ x_1 + 10x_2 \leq 10\right\},$$

it is clear that the smallest bounding box as described above is given by

$$X' = \left\{\boldsymbol{x} \in \mathbb{R}^2 \;\middle|\; 0 \leq x_1 \leq 10,\ \ 0 \leq x_2 \leq 1\right\}.$$

Then, the furthest point in X' from the global minimum $\boldsymbol{b}$ is clearly given by $[10\ \ 1]^\top$ for all of $\boldsymbol{b}^1$, $\boldsymbol{b}^2$ and $\boldsymbol{b}^3$. For the first coordinate, for example, we find $|10 - 4.9| > |0 - 4.9|$ and therefore the first coordinate of the furthest point must be 10. Initialization based on this furthest point ensures the algorithm converges to the global maximum.

	Objective Value				Computation Time (s)			
	Random	MVIE	AC	Box	Random	MVIE	AC	Box
# 1	**563.58**	509.65	509.65	**563.58**	12.94	2.74	2.70	0.72
# 2	**1614.13**	1545.08	1545.08	1545.08	17.44	2.76	3.27	1.02
# 3	**447.29**	312.92	422.90	**447.29**	62.55	6.04	5.49	3.56
# 4	**1816.00**	**1816.00**	**1816.00**	**1816.00**	11.72	1.97	2.23	0.58
# 5	**1610.13**	940.55	**1610.13**	**1610.13**	10.98	2.20	2.50	0.69

Table 2: Objective values and computation times for five instances with $n = 100$, a feasible set given by the intersection of ellipsoids and an exponential objective for the four different initialization methods: random, based on the maximum volume inscribed ellipsoid (MVIE), based on ellipsoids around the analytic center (AC) and based on a bounding box. For each instance the best found objective value is marked in bold.

A second setting in which the bounding box performs remarkably well is when the feasible set is the intersection of ellipsoids. Depending on generation, such sets can also be very narrow, which results in issues for the ellipsoidal approximations. In Table 2 the results for five instances with a feasible set equal to the intersection of 3 ellipsoids are given. Specifically, the feasible set is given by

$$X = \left\{ \boldsymbol{x} \in \mathbb{R}^{100} \;\middle|\; \left\| \boldsymbol{L}_i^\top \boldsymbol{L}_i(\boldsymbol{x} + \boldsymbol{q}_i) \right\|_2 \leq 1 \quad i = 1, 2, 3 \right\},$$

where $\boldsymbol{L}_i \in \mathbb{R}^{n \times n}$ is drawn from a uniform distribution on $[0, 1]^{n \times n}$ and $\boldsymbol{q}_i \in \mathbb{R}^n$ is drawn uniformly from the surface of the unit ball in $\mathbb{R}^n$. The objective function is given by

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} e^{\alpha_i x_i} - \boldsymbol{\beta}^\top \boldsymbol{x},$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are drawn from a uniform distribution on $[0, 1]^n$. While random initialization manages to find the best solution for all five instances it is rather costly in terms of computation time. This is mainly caused by the high number of iterations the phase 2 algorithm needs to converge. The box method, on the other hand, is remarkably fast and also finds high quality solutions.

5 Full Algorithm

The complete algorithm we suggest using to solve the convex maximization problem (P) is listed as Algorithm MCF. The structure of this algorithm follows the structure we have outlined in the previous sections, i.e., it is a two phase approach. Depending on the type of feasible set, different initialization options are used in phase 1 to provide starting values for the alternating direction method in phase 2. Specifically, for ellipsoidal feasible sets, no approximation is needed and thus initialization can be limited to random and furthest from the global maximum. For all other feasible sets, we consider three different initialization options: random, bounding box and analytic center. Recall that we elect to not use the maximum volume inscribed ellipsoid as the quality of solutions found by that approach is at most a minor improvement over the analytic center approach, while the computation time required is significantly higher.

Algorithm 2 MCF

```
1: Input: Maximum time $T^{max}$, number of random initializations $N_R$, feasible set $X$, objective function $f$, gradient $\nabla f$, Hessian $\nabla^2 f$, global minimum $\bar{\boldsymbol{x}}_g$, constrained minimum $\bar{\boldsymbol{x}}_c$, threshold $\epsilon$
2: Initialization: Set initialization counter $k \leftarrow 1$, elapsed time $T \leftarrow 0$, set of candidate solutions $C = \emptyset$
3: $k^{max} \leftarrow 3$
4: if $X$ is an ellipsoid then
5:     $k^{max} \leftarrow k^{max} - 1$
6: end if
7: if $n > 100$ then
8:     Skip random initialization: $k \leftarrow 2$
9: end if
10: while $k \leq k^{max}$ and $T \leq T^{max}$ do
11:     Create empty initial value sets $\mathcal{I}_x \leftarrow \emptyset$, $\mathcal{I}_y \leftarrow \emptyset$
12:     if $k = 1$ then
13:         for $i = 1, \dots, N_R$ do
14:             Generate random direction $\boldsymbol{y} \sim U\left([0,1]^n\right)$ and add to initial values: $\mathcal{I}_y \leftarrow \mathcal{I}_y \cup \{\boldsymbol{y}\}$
15:         end for
16:     else
17:         if $X$ is an ellipsoid then
18:             Set $X' \leftarrow X$
19:         else
20:             if $k = 2$ then
21:                 Set $X' \leftarrow \{\boldsymbol{x} \in \mathbb{R}^n \mid \min_{\boldsymbol{x} \in X} x_i \leq x_i \leq \max_{\boldsymbol{x} \in X} x_i \quad i = 1, \dots, n\}$
22:             else if $k = 3$ then
23:                 Set $X' \leftarrow \mathcal{E}_{in}$                                ▷ (12) in Section 4.2
24:                 Set $X'' \leftarrow \mathcal{E}_{circum}$                           ▷ (13) in Section 4.2
25:             end if
26:         end if
27:         for $\bar{\boldsymbol{x}} \in \{\bar{\boldsymbol{x}}_g, \bar{\boldsymbol{x}}_c\}$ do
28:             Find the furthest point in $X'$: $\hat{\boldsymbol{x}} \in \operatorname{argmax}_{\boldsymbol{x} \in X'} d(\boldsymbol{x}, \bar{\boldsymbol{x}})$
29:             if $X' \subseteq X$ or $\bar{\boldsymbol{x}} = \bar{\boldsymbol{x}}_c$ then
30:                 Find the intersections of the line through $\bar{\boldsymbol{x}}$ and $\hat{\boldsymbol{x}}$ with the boundary of $X$ and denote by $\boldsymbol{x}'$ the furthest of these from $\bar{\boldsymbol{x}}$          ▷ Section 3.3
31:                 Include $\boldsymbol{x}'$ as initial value: $\mathcal{I}_x \leftarrow \mathcal{I}_x \cup \{\boldsymbol{x}'\}$
32:             end if
33:             Update sets with initial values: $\mathcal{I}_x \leftarrow \mathcal{I}_x \cup \{\hat{\boldsymbol{x}}\}$ and $\mathcal{I}_y \leftarrow \mathcal{I}_y \cup \{\hat{\boldsymbol{x}} - \bar{\boldsymbol{x}}\}$
34:         end for
35:         if $k = 3$ then
36:             Repeat the above procedure on lines 25 - 32 for $X''$
37:         end if
```

Algorithm 2 MCF (Continued)

```
        end if
        for $\boldsymbol{y}^0 \in \mathcal{I}_y$ do
            Find $\boldsymbol{x}^1 \in \operatorname{argmax}_{\boldsymbol{x} \in X} \boldsymbol{x}^\top \boldsymbol{y}^0$
            Add $\boldsymbol{x}^1$ to initial values: $\mathcal{I}_x \leftarrow \mathcal{I}_x \cup \{\boldsymbol{x}^1\}$
        end for
        for $\boldsymbol{x}^0 \in \mathcal{I}_x$ do
            Use the Alternating Direction method to find $\boldsymbol{x}^*$ with initial value $\boldsymbol{x}^0$ and threshold $\epsilon$    ▷ Algorithm 1 in Section 2
            Update candidate solutions: $C \leftarrow C \cup \{\boldsymbol{x}^*\}$
        end for
        Increment initialization counter $k \leftarrow k + 1$ and update $T$ to the elapsed time since initialization
    end while
    return Best solution $\boldsymbol{x}^* \in \operatorname{argmax}_{\boldsymbol{x} \in C} f(\boldsymbol{x})$
```

Given an initialization option, the algorithm adds a number of initial values for $\boldsymbol{x}^0$ and $\boldsymbol{y}^0$ to the sets $\mathcal{I}_x$ and $\mathcal{I}_y$, respectively. For random initialization, this is limited to uniform random values of $\boldsymbol{y}^0$. For the other three initialization options, on the other hand, the furthest point $\hat{\boldsymbol{x}}$ on the approximate feasible set X' from both the global and constrained minimum is determined. This furthest point $\hat{\boldsymbol{x}}$ is added to $\mathcal{I}_x$ and the direction from the minimum to the furthest point, i.e., $\hat{\boldsymbol{x}} - \bar{\boldsymbol{x}}$ is added to $\mathcal{I}_y$. Additionally, if X' is an inner approximation of X or the constrained minimum is considered, the *line* method is employed to obtain an extra initial value for $\boldsymbol{x}^0$. In the second phase of the algorithm, the alternating direction method is used to find locally optimal solutions based on all obtained initial values.

6 Numerical Results

In this section we discuss the results from running our MCF algorithm on selected instances from the literature and a set of randomly generated instances. For all those instances we report the best objective value found for the three considered initialization methods separately, i.e., when initializing the algorithm randomly, based on the ellipsoids around the analytic center and based on the bounding box, all for the unconstrained an constrained minimum of f. Specifically, for initialization based on the analytic center, this means the reported objective value is the best of ten objective values: for both the unconstrained and constrained minimum of f, the *line*, *direction* and $\hat{\boldsymbol{x}}$ methods for the inscribed ellipsoid and the *direction* and $\hat{\boldsymbol{x}}$ methods for the circumscribing ellipsoid. Note that we do not use the *line* method with the circumscribing ellipsoid, as the line in question is not guaranteed to intersect with the feasible set, and in our experience it rarely does in higher dimensions. Similarly, the reported objective value for the bounding box is the best of four objective values: for both the unconstrained and constrained minimum of f, the *direction* and $\hat{\boldsymbol{x}}$ methods. We remark that the MCF algorithm we propose in Section 5 yields only a single final solution, which corresponds to the best of the four objective values we report throughout this section. All numerical results are conducted on a desktop with 16 GB RAM and a 3.6 GHz AMD Ryzen 5 processor.

6.1 Algorithm Performance

First we discuss, for a small number of random instances, the behavior of the different phase 1 approaches as well as the alternating direction algorithm used in phase 2 of the algorithm. We consider 5 instances with $n = 25$, a polytope as feasible set and an exponential objective. Specifically, the objective function is given by

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} e^{\alpha_i x_i} - \boldsymbol{\beta}^\top \boldsymbol{x},$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are drawn from a uniform distribution on $[0,1]^n$. The feasible set is constructed by randomly generating $2n$ constraints by intersecting randomly generated halfspaces with the hyperrectangle $\{\boldsymbol{x} \in \mathbb{R}^n : \ \mathbf{0} \leq \boldsymbol{x} \leq \mathbf{1}\}$. The $2n$ random halfspaces are generated by generating a vector $\boldsymbol{s}_j \in \mathbb{R}^n$ uniformly distributed on the surface of the hypersphere, and a scalar $r_j \in \mathbb{R}$ uniformly distributed from the interval $\left[-\frac{1}{2}\|\boldsymbol{s}_j\|_1, \frac{1}{2}\|\boldsymbol{s}_j\|_1\right]$. We then consider the constraint $\boldsymbol{s}_j^\top(\boldsymbol{x} - \frac{1}{2}\mathbf{1}) \leq r_j$ if $r_j > 0$ and $\boldsymbol{s}_j^\top(\boldsymbol{x} - \frac{1}{2}\mathbf{1}) \geq r_j$ if $r_j \leq 0$. This construction ensures that the center of the hyperrectangle, $\frac{1}{2}\mathbf{1}$, always remains feasible.

For each instance we report the average objective function value of the initial starting point found in phase 1 by all three initialization methods in Table 3. Specifically, for the analytic center and bounding box methods this average is over the initial values found by the *line* and *direction* methods as well as $\hat{\boldsymbol{x}}$ for both the constrained and unconstrained minimum, while for random initialization this is simply the average over 100 random extreme points as described in Section 3.5. Additionally, we report the average number of iterations it took the alternating direction method to converge to a local maximum from that initial value in phase 2 as well as the objective value of the best found local maximum.

The results in Table 3 illustrate the value of using the furthest point as well as the Hessian based distance. Indeed, the initial objective values reported for the analytic center approach are

		Initial Objective			# Iterations			Final Objective		
	n	Random	AC	Box	Random	AC	Box	Random	AC	Box
# 1	25	27.89	32.20	29.98	2.7	1.5	1.8	33.32	33.32	33.32
# 2	25	29.09	33.58	30.95	4.2	2.4	3.4	35.02	35.02	35.01
# 3	25	29.43	32.59	30.20	3.7	2.2	2.0	33.98	33.97	33.98
# 4	25	30.24	34.96	31.61	3.9	2.2	2.6	36.54	36.54	35.51
# 5	25	26.01	28.73	27.74	4.0	1.6	2.4	29.79	29.79	29.79

Table 3: The average initial objective value after phase 1, the number of iterations in phase 2 and the best found final objective for five instances with polytopic feasible sets with $2n$ constraints and an exponential objective listed separately for all three initialization methods.

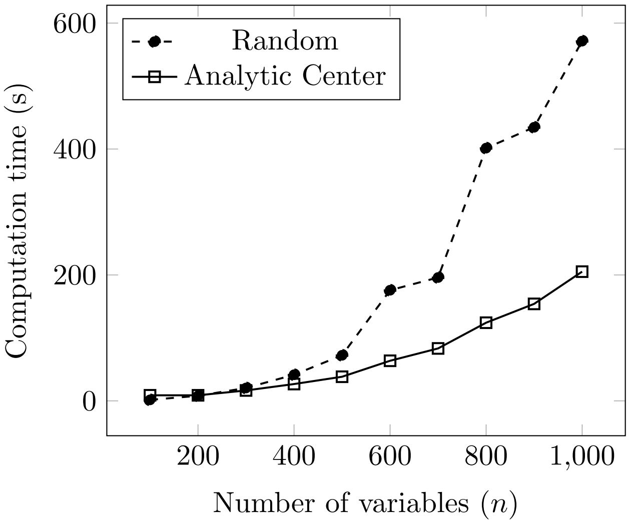

Figure 5: Total computation time in seconds for instances with polytopic feasible sets with $2n$ constraints in different dimensions for random initialization with 100 runs.

reasonably close to the final objective values found. The box approach also clearly yields better initial solutions than the random approach, though significantly worse initial solutions than the analytic center approach. Nevertheless, all methods converge to (almost) identical final solutions. We suspect the lower quality of the initial box solutions is caused by the box approach having to use the Euclidean distance instead of the Hessian based distance for tractability reasons. This once again underlines the importance of choosing the appropriate distance measure. Related to the initial objective values is the number of iterations needed. Mostly, we observe the trend that a better initial solution means less iterations are needed to converge in phase 2. Indeed, the analytic center method, which has the best initial solutions by far, only needs 2 or 3 iterations to converge, while random initialization requires a few extra.

Important is to note that while random initialization does well in relatively low dimensions, it does not scale particularly well. This can be explained by considering the reasons why random initialization both finds high quality solutions and does not require much computation time. Through initializing $\boldsymbol{y}^0$ by random sampling from the unit ball, one essentially starts at a random extreme point of the feasible set. Then, in phase 2 of the algorithm, one obtains in some sense the 'closest' local optimum. In small and medium sized problems, maximizing a linear function over the feasible set requires very little computation time, and thus one can use enough random initial values to essentially guarantee finding the global maximum. As one moves to larger problems, however, two things change. First, the number of extreme points can increase exponentially, and the number of local optima increases accordingly. Hence, many more initial values are required to ensure finding globally optimal, or even high quality solutions. Second, depending on the complexity of the feasible set, solving the optimization problem in each iteration requires an increasing amount of time. In some sense, therefore, random initialization scales exceptionally badly in the number of dimensions of the problem. Figure 5 illustrates this increase by plotting the computation time for polytopic instances of dimensions $n = 100$ through

			Objective Value				Computation Time (s)		
			Selvi et al.	MCF Algorithm			MCF Algorithm		
	n	m	(2020)	Random	AC	Box	Random	AC	Box
# 1	20	10	394.751	394.751	248.712	394.751	0.45	1.08	0.07
# 2	20	10	884.751	884.751	738.712	884.751	0.43	1.06	0.08
# 3	10	15	4,674.7	4,674.7	4,674.7	4,674.7	0.40	0.99	0.04
# 4	50	62	175,710	175,706	175,706	175,706	0.43	1.48	0.16
# 5	100	130	692,610.0	692,613	692,613	692,613	0.61	2.10	0.37
# 6	200	240	6,020,800	6,020,787	6,020,787	6,020,787	0.75	3.43	0.94
# 7	240	280	1,855,700	1,855,740	1,855,740	1,855,740	0.95	3.88	1.26

Table 4: Objective values and computation times for the instances from section 4.3 of (Selvi et al., 2020) with a polytope as feasible set with n variables and m constraints for the three different initialization methods: random, based on ellipsoids around the analytic center (AC) and based on a bounding box.

$n = 1000$ for both random initialization and initialization based on the analytic center.

The performance of the algorithm on a large set of randomly generated instances is depicted in Tables 8 through 10 in Appendix A. Table 7 provides an overview of the type of objective and feasible set for all instances in those tables. In general, we see very similar performance to what is presented above in Table 3. The four suggested initialization methods all find relatively comparable solutions and mostly differ in terms of computation time. As the computation time for random, analytic center and box initialization is fairly low, this underlines our decision to simply use the best solution found in phase 1.

6.2 Quadratic Maximization

We focus on convex quadratic maximization with polytopic feasibility. First, we consider the 7 instances solved in section 4.3 of Selvi et al. (2020), whose method is restricted to polytopes. Table 4 list the objective value for the various initialization methods we propose as well as the upper and lower bounds reported by Selvi et al. (2020).

For the first two instances we find solutions whose objective value is significantly below the objective value reported by Selvi et al. (2020) for analytic center based initialization. These instances have the same feasible region, and its shape does not naturally lend itself to approximation by an ellipsoid, which explains the poor performance. Observe that both random initialization and initialization based on the bounding box yields the global maximum, as they perform well in such instances. For all other instances all initialization methods find the same solution.

Additionally, we consider two problems from (Enhbat, 1996) with different dimensions. Specifically, we consider P10 and P12, both of which concern maximizing a convex quadratic function over a polytope. Table 5 contains the objective values and computation times for all

			Objective Value			Computation Time (s)		
		Enhbat		MCF Algorithm				
	n	(1996)	Random	AC	Box	Random	AC	Box
P10	3	721.4	721.4	721.4	721.4	0.21	1.53	0.02
P10	10	83712	81339	83712	83712	0.20	1.61	0.04
P10	30	6440531	5456454	6440531	6440531	0.21	1.92	0.08
P10	60	101506747	72378785	101506747	101506747	0.22	2.49	0.15
P10	80	319560716	208178879	319560716	319560716	0.22	2.65	0.20
P10	100	778330545	514010719	778330545	778330545	0.23	2.98	0.25
P10	150	3927744505	2517210100	3927744505	3927744505	0.24	3.77	0.39
P12	2	45.5	45.5	45.5	45.5	0.23	1.51	0.02
P12	5	3604	3604.25	3604.25	3604.25	0.33	1.52	0.02
P12	10	109333.5	109333.5	109333.5	109333.5	0.34	1.66	0.03
P12	30	25766625.5	25766625.5	25766625.5	25766625.5	0.32	1.89	0.08
P12	40	108196334	108196334	108196334	108196334	0.33	2.10	0.10
P12	70	1767930209	1767930209	1767930209	1767930209	0.35	2.54	0.18
P12	80	3444342668	3444342668	3444342668	3444342668	0.33	2.66	0.20
P12	90	6203290501	6203290501	6203290501	6203290501	0.34	2.85	0.23
P12	99	9986343609	9986343609	9986343609	9986343609	0.35	2.98	0.25

Table 5: Objective values and computation times for P10 and P12 from (Enhbat, 1996) in different dimensions n with $2n$ constraints for the three different initialization methods: random, based on ellipsoids around the analytic center (AC) and based on a bounding box.

three initialization methods. Table 5 also lists the objective value found by Enhbat (1996). Interesting to note is that P10 is the first instance where random initialization falls short by a large margin. Particularly in higher dimensions ($n \geq 20$), it does not manage to converge to a local maximum whose objective value is even close to what the ellipsoidal based initialization methods find. Other than that, we note that both the analytic center and the bounding box method find the same objective value as reported by Enhbat (1996) for all instances. The box method in particular converges remarkably fast.

6.3 Problems with Integer Variables

As mentioned before, our approach can also be applied to problems that have linear constraints and integer variables. The only change such integer constraints bring is the complexity of the optimization problem that is to be solved in phase 2 of the algorithm, where a linear function is maximized over the feasible set. In this section we consider the first 5 instances from section 4.3 of (Selvi et al., 2020), that were also considered in Section 6.2, with the added constraint that all variables must be integer. Table 6 reports the objective value and computation time for the four different initialization methods. In terms of objective value the different initialization approaches perform similar to the continuous case in Section 6.2. Specifically, the random and bounding

			Objective			Computation Time (s)		
	n	m	Random	AC	Box	Random	AC	Box
# 1	20	10	**370.0**	238.0	**370.0**	4.67	2.69	0.34
# 2	20	10	**860.0**	728.0	**860.0**	4.73	1.20	0.36
# 3	10	15	**4660.5**	**4660.5**	**4660.5**	0.89	1.76	0.06
# 4	50	62	**175300.9**	**175300.9**	**175300.9**	3.13	2.66	0.24
# 5	100	130	**691844.8**	**691844.8**	**691844.8**	8.76	4.63	0.55

Table 6: Objective values and computation time for the first five instances from section 4.3 of (Selvi et al., 2020) with n variables and m constraints additionally restricted to integer solutions for the three different initialization methods: random, based on ellipsoids around the analytic center (AC) and based on a bounding box. For each instance the best found objective value is marked in bold.

box approaches find the global maximum for instances 1 and 2, while the ellipsoidal approach does not. For the other instances, all approaches converge to the same solution. Clearly, the impact of n on the computation time is significantly higher for integer problems.

7 Conclusion

Here we discuss the practical conclusions that can be drawn from the computational testing of the performance of Algorithm MCF. The three phase 1 initialization methods perform differently for problem (P). *Random* initialization is efficient for problems of relatively small dimension, and is neither competitive, nor reliable, for higher dimensions (see, e.g., Tables 5 and 10). *Analytic center* initialization is quite reliable and only fails to deliver a good starting point for phase 2 in few cases (e.g., narrow polytopes). This method scales well in terms of computation time as the dimension grows. *Box* initialization is both fast and reliable, particularly for problems with a polytope as feasible set. It mainly falls short on ellipsoidal feasible sets.

Recall that Algorithm MCF uses all three phase 1 methods and reports the best found result. We can afford using all initialization methods due to the efficiency of the algorithm. In particular, this is due to the fact that phase 2, the Alternating Direction Method (ADM), requires very few steps, as it starts from a good candidate for the global maximizer ("the furthest point from the minimizer") provided by phase 1. Moreover, when the feasible set of problem (P) is a polytope, ADM only needs to solve linear optimization problems. This implies in fact that in this case Algorithm MCF is capable of solving problem (P) with integer valued variables.

In future research, we plan to extend our approach to non-differentiable convex objective functions. Specifically, the biconvex reformulation can be extended to include an additional regularization term which still enables the use of an alternating direction method. In fact, this still yields a closed form solution for the direction $\boldsymbol{y}$ in terms of the proximal operator of the objective function. Additionally, we plan to extend the approach to DC programming, where the aim is to maximize the difference of convex functions. The main challenge in this extension is the existence of multiple minima of DC functions. Therefore, the notion of distance to a minimum does no longer provide a global optimality condition naturally. However, when the concave part of the DC objective is linearized, which is commonly done in DC programming (Le Thi et al., 2014; Lipp and Boyd, 2016), a problem of type (P) is obtained to which Algorithm MCF can be applied.

Acknowledgments

The research of the second author was supported by the Netherlands Organisation for Scientific Research (NWO) Research Talent [Grant 406.17.511].

References

Andrianova, A., Korepanova, A., and Halilova, I. (2016). One algorithm for branch and bound method for solving concave optimization problem. In *IOP Conference Series: Materials Science and Engineering*, volume 158, page 012005. IOP Publishing.

Audet, C., Hansen, P., and Savard, G. (2005). *Essays and surveys in global optimization*, volume 7. Springer Science & Business Media.

Beck, A. (2017). *First-order methods in optimization*, volume 25. SIAM.

Ben-Tal, A. and Den Hertog, D. (2014). Hidden conic quadratic representation of some nonconvex quadratic optimization problems. *Mathematical Programming*, 143(1-2):1–29.

Ben-Tal, A. and Nemirovski, A. (2001). *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*. SIAM.

Ben-Tal, A. and Teboulle, M. (1996). Hidden convexity in some nonconvex quadratically constrained quadratic programming. *Mathematical Programming*, 72(1):51–63.

Blåsjö, V. (2015). *Intuitive Infinitesimal Calculus*. Intellectual Mathematics.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge university press.

Enhbat, R. (1996). An algorithm for maximizing a convex function over a simple set. *Journal of Global Optimization*, 8(4):379–391.

Gal, S., Bachelis, B., and Ben-Tal, A. (1978). On finding the maximal range of validity of a constrained system. *SIAM Journal on Control and Optimization*, 16(3):473–503.

Jarre, F. (1994). Optimal ellipsoidal approximations around the analytic center. *Applied Mathematics and Optimization*, 30(1):15–19.

Kuhn, H. W. (1976). Nonlinear programming: A historical view. In Cottle, R. W. and Lemke, C. E., editors, *SIAM-AMS Proceedings*, volume 9, pages 1–26. American Mathematical Society.

Le Thi, H. A. and Dinh, T. P. (2018). Dc programming and dca: thirty years of developments. *Mathematical Programming*, 169(1):5–68.

Le Thi, H. A., Huynh, V. N., and Dinh, T. P. (2014). Dc programming and dca for general dc programs. In *Advanced Computational Methods for Knowledge Engineering*, pages 15–35. Springer.

Lipp, T. and Boyd, S. (2016). Variations and extension of the convex–concave procedure. *Optimization and Engineering*, 17(2):263–287.

Mangasarian, O. (1996). Machine learning via polyhedral concave minimization. In *Applied Mathematics and Parallel Computing*, pages 175–188. Springer.

Pardalos, P. M. and Rosen, J. B. (1986). Methods for global concave minimization: A bibliographic survey. *Siam Review*, 28(3):367–379.

Selvi, A., Ben-Tal, A., Brekelmans, R., and Den Hertog, D. (2020). Convex maximization via adjustable robust optimization. *Available on Optimization Online*.

Zass, R. and Shashua, A. (2007). Nonnegative sparse PCA. In Schölkopf, B., Platt, J. C., and Hoffman, T., editors, *Advances in Neural Information Processing Systems*, volume 19, pages 1561–1568. MIT Press.

Zwart, P. B. (1974). Global maximization of a convex function with linear inequality constraints. *Operations Research*, 22(3):602–609.

A Additional Tables

Instance #	X	$f(\boldsymbol{x})$
1-10	$\{\boldsymbol{Dx} \le \boldsymbol{d}\}, \; \boldsymbol{D} \in \mathbb{R}^{2n \times n}$	$\boldsymbol{x}^\top \boldsymbol{Q}^\top \boldsymbol{Qx} + \boldsymbol{q}^\top \boldsymbol{x}, \;\; Q_{ij} \sim U(0,1), \; q_i \sim U(0,1)$
11-20	$\{\boldsymbol{Dx} \le \boldsymbol{d}\}, \; \boldsymbol{D} \in \mathbb{R}^{2n \times n}$	$\sum_{i=1}^n e^{\alpha_i x_i} - \boldsymbol{\beta}^\top \boldsymbol{x}, \;\; \alpha_i \sim U(0,1), \; \beta_i \sim U(0,1)$
21-30	$\left\{\|\boldsymbol{x} - \boldsymbol{c}^i\|_2 \le 1, i = 1, \ldots, 5\right\}$	$\boldsymbol{x}^\top \boldsymbol{Q}^\top \boldsymbol{Qx} + \boldsymbol{q}^\top \boldsymbol{x}, \;\; Q_{ij} \sim U(0,1), \; q_i \sim U(0,1)$
31-40	$\left\{\|\boldsymbol{x} - \boldsymbol{c}^i\|_2 \le 1, i = 1, \ldots, 5\right\}$	$\sum_{i=1}^n e^{\alpha_i x_i} - \boldsymbol{\beta}^\top \boldsymbol{x}, \;\; \alpha_i \sim U(0,1), \; \beta_i \sim U(0,1)$
41-50	$\left\{\left\|\boldsymbol{L}_i^\top \boldsymbol{L}_i(\boldsymbol{x} + \boldsymbol{c}_i)\right\|_2 \le 1, i = 1, 2, 3\right\}$	$\boldsymbol{x}^\top \boldsymbol{Q}^\top \boldsymbol{Qx} + \boldsymbol{q}^\top \boldsymbol{x}, \;\; Q_{ij} \sim U(0,1), \; q_i \sim U(0,1)$

Table 7: Instance type description for Tables 8, 9 and 10.

		Objective				Computation Time (s)			
	n	Random	MVIE	AC	Box	Random	MVIE	AC	Box
# 1	25	1500.93	1500.93	1500.93	1500.93	0.48	2.05	2.14	0.10
# 2	25	1591.05	1591.05	1591.05	1591.05	0.47	2.00	2.18	0.09
# 3	25	1503.87	1503.87	1503.87	1503.87	0.56	2.02	2.13	0.09
# 4	25	1506.97	1506.97	1506.97	1506.97	0.53	1.95	2.07	0.11
# 5	25	2039.13	2039.13	2039.13	2039.13	0.41	1.95	2.14	0.08
# 6	100	91923.01	91923.01	91923.01	91923.01	1.59	66.62	5.12	0.84
# 7	100	89345.70	89345.70	89345.70	89345.70	1.61	63.73	4.33	0.83
# 8	100	80896.40	80896.40	80896.40	80896.40	1.96	63.43	4.46	0.85
# 9	100	88155.89	88155.89	88155.89	88155.89	1.77	63.25	4.67	0.87
# 10	100	86758.52	86758.52	86758.52	86758.52	1.91	70.93	4.39	0.87
# 11	25	33.32	33.32	33.32	33.32	0.49	1.94	2.16	0.08
# 12	25	35.02	35.01	35.02	35.01	0.68	2.03	2.27	0.01
# 13	25	33.98	33.97	33.97	33.98	0.67	2.00	2.16	0.09
# 14	25	36.54	36.54	36.54	36.51	0.64	2.01	2.20	0.09
# 15	25	29.79	29.79	29.79	29.79	0.69	1.97	2.16	0.09
# 16	100	132.44	132.56	132.58	132.27	2.16	62.98	4.81	0.84
# 17	100	137.71	137.85	137.75	137.80	1.91	63.46	4.31	0.88
# 18	100	132.28	132.40	132.40	132.25	2.18	67.88	4.73	0.85
# 19	100	130.50	130.81	130.76	130.82	1.75	59.17	4.72	0.83
# 20	100	134.21	134.25	134.21	134.21	2.18	62.66	4.37	0.86

Table 8: Objective values and computation times for instances with a polytopic feasible set with $2n$ constraints for the four different initialization methods: random, based on the maximum volume inscribed ellipsoid (MVIE) based on ellipsoids around the analytic center (AC) and based on a bounding box

		Objective				Computation Time (s)			
	n	Random	MVIE	AC	Box	Random	MVIE	AC	Box
# 21	25	127.66	127.66	127.66	127.66	0.83	1.76	1.67	0.10
# 22	25	67.00	67.00	67.00	67.00	0.85	1.72	1.68	0.10
# 23	25	109.07	109.07	109.07	82.69	0.81	1.65	1.63	0.10
# 24	25	132.80	132.80	132.80	132.80	0.92	1.64	1.67	0.10
# 25	25	124.38	124.38	124.38	124.38	0.84	1.63	1.62	0.1
# 26	100	1827.31	1827.31	1827.31	1827.31	4.86	106.11	2.60	1.74
# 27	100	1728.63	1728.63	1728.63	1728.63	5.21	133.23	2.560	1.77
# 28	100	1524.69	1524.69	1524.69	1524.69	5.20	131.14	2.65	1.75
# 29	100	1809.34	1809.34	1809.34	1809.34	6.13	107.13	2.64	1.93
# 30	100	1841.35	1841.35	1841.35	1841.35	4.44	98.27	2.50	1.77
# 31	25	27.14	27.14	27.14	27.14	1.76	1.72	1.73	0.15
# 32	25	26.76	26.76	26.76	26.76	1.76	1.67	1.72	0.15
# 33	25	27.16	27.16	27.16	27.16	1.56	1.67	1.72	0.14
# 34	25	26.58	26.58	26.58	26.58	1.56	1.68	1.71	0.14
# 35	25	26.46	26.46	26.46	26.46	1.15	1.62	1.67	0.11
# 36	100	103.39	103.39	103.39	103.39	7.01	87.21	2.74	1.90
# 37	100	104.47	104.47	104.47	104.47	7.22	87.15	2.81	1.97
# 38	100	103.34	103.34	103.34	103.34	7.58	102.80	2.86	2.02
# 39	100	103.47	103.47	103.47	103.47	7.40	95.89	2.92	2.02
# 40	100	104.21	104.21	104.21	104.21	7.20	108.51	2.91	2.03

Table 9: Objective values and computation times for instances with the intersection of five balls as feasible set for the four different initialization methods: random, based on the maximum volume inscribed ellipsoid (MVIE) based on ellipsoids around the analytic center (AC) and based on a bounding box.

		Objective				Computation Time (s)			
	n	Random	MVIE	AC	Box	Random	MVIE	AC	Box
# 41	25	369.01	369.01	369.01	369.01	12.95	2.74	2.70	0.72
# 42	25	672.09	672.09	672.09	672.09	17.44	2.76	3.27	1.02
# 43	25	683.30	683.30	683.30	683.30	62.55	6.04	5.49	3.56
# 44	25	461.58	461.58	461.58	461.58	11.72	1.97	2.23	0.58
# 45	25	390.81	390.81	369.82	390.81	10.98	2.20	2.50	0.69
# 46	100	2102.90	2082.78	2082.78	2102.90	275.37	160.59	36.45	16.18
# 47	100	2569.39	2569.39	2569.39	2569.39	145.93	147.19	13.09	8.88
# 48	100	2732.37	2732.37	2732.37	2732.37	103.22	154.00	9.78	7.53
# 49	100	2386.82	2386.82	2386.82	2386.82	104.61	121.78	10.05	7.60
# 50	100	2516.33	2516.33	2516.33	2516.33	151.81	125.74	15.60	10.71

Table 10: Objective values and computation times for instances with the intersection of three ellipsoids as feasible set for the four different initialization methods: random, based on the maximum volume inscribed ellipsoid (MVIE) based on ellipsoids around the analytic center (AC) and based on a bounding box.