# A Parallel Linear Active Set Method

**E. Dov Neimand and Şerban Sabău**

## 1 Introduction

For years, interior point methods have dominated the field of linear constrained convex minimization [16, 20]. These methods, though powerful, often exhibit three downsides. First, many interior point methods do not lend themselves to parallel implementations without imposing additional criteria. Second, they often require the feasible space be nonempty, [12], or even require a starting feasible point, and when one is unavailable fall back on a second optimization problem, Phase I Method [5]. Third, they typically terminate when they are within an $\epsilon > 0$ distance of the true optimal point, rendering their complexity a function of their accuracy [5, 10].

Here we introduce a linear-inequality-constrained convex minimization method that alleviates these drawbacks. Our method can offer superior performance to state-of-the-art methods when the number of processors is polynomial as a function of the number of constraints in Euclidean space. When this is not the case, though computationally more complex, our method's simple implementation, non-asymptotic convergence, and broad applicability offer considerable value.

Minimization of convex objective functions over non-convex polyhedra struggles to balance slower accurate methods, those with global solutions, against heuristic algorithms that offer a local optimum or pseudo optimal points that may or may not

The work of the authors is based upon research supported by the National Science Foundation under the CAREER Award Number # 1653756.

E. D. Neimand (✉) · Ş. Sabău
Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ, USA
e-mail: eneimand@stevens.edu; ssabau@stevens.edu

 
B. Goldengorin, S. Kuznetsov (eds.), *Data Analysis and Optimization*, Springer Optimization and Its Applications 202,
https://doi.org/10.1007/978-3-031-31654-8_16

be in the feasible space, Diamond et al. [7]. We present a second algorithm, modified from the first that optimizes over non-convex polyhedra. The method does not compromise on accuracy and has similar complexity to the convex method. Our non-convex method takes advantage of information about the non-convex polyhedron's faces for improved performance over the convex algorithm.

For a simple brute force approach to the three problems facing standard interior point methods, [19] presents a progenitor to Algorithm 1, in finding the projection, $\Pi_P(y)$, of a point, $y \in \mathbb{R}^n$, onto a convex polyhedron, $P \subset \mathbb{R}^n$. Their algorithm first checks if $y \in P$, and if it is not, considers each subset of $P$'s defining inequality constraints, as equality constraints. Projections onto these sets of equality constraints are easily found. A filter removes the affine projections that are outside $P$, and of those that remain, the closest to $y$ is $\Pi_P(y)$.

In expanding from polyhedral projections in $\mathbb{R}^n$ to a generic convex objective function in a Hilbert space, our algorithm makes use of a black-box linear-equality constrained convex minimization method for our objective function $f : \mathbb{H} \to \mathbb{R}$. Textbooks and papers on unconstrained minimization in Hilbert spaces are now ubiquitous, [3, 4, 6] provide examples. Recently [11] and [14] presented unconstrained minimization methods atop the plethora of preceding research. Given a set of linear-equality constraints, Boyd et al. [5], suggests eliminating the linear equality constraints with a change in variable, reducing the problem to unconstrained minimization in fewer dimensions. Reliance on our black-box method is well-founded.

Unconstrained convex functions can often be optimized quickly. Some functions, like projection functions can be optimized in $O(n^3)$ operations over an affine space in $\mathbb{R}^n$, Plesnik [15]. Note that there is no $\epsilon > 0$ term in the complexity.

Our algorithm employs a test that, together with the black box method, reviews a set of linear inequality constraints, $L$. The test passes $L$ only if the black-box method can generate the constrained optimal point by treating $L$'s elements as equality constraints. Necessary criteria often allow for the test to fast fail $L$ without using the black-box method, instead looking back at previous applications of the test on subsets of $L$ that have one less inequality than $L$. This fast fail, as a function of the number of dimensions, has quadratic sequential complexity, and can be completely multi-threaded down to near constant complexity. When the test is unable to fast fail, it resorts to calling the black-box method on the inequality turned equality constraints in $L$. In both cases the test generates the optimal point of $f$ over $L$.

Iterative and largely parallel application of the test over growing sets of inequality constraints yields Algorithm 1, which returns $\arg\min_P f$. Algorithm 1 does not employ the test for sets larger than $\min(r, n)$, where $r$ is the total number of constraints and $n \in \mathbb{N} \cup \{\infty\}$ the dimension of the Hilbert space $\mathbb{H}$. Unlike [19], which continues to project onto all the affine spaces after computing and in order to confirm $\Pi_P(y)$, Algorithm 1 ceases its search as soon as the black-box method computes the optimal points.

Our algorithm does not utilize an iterative minimization sequence and therefor preserves valuable properties of the underlying unconstrained minimization method.

When $\arg\min_{\mathbb{H}} f$ finds an exact answer without the need for an iteration arriving within an $\epsilon$ distance of the optimal point, so too does our algorithm.

Because of the finite number of operations required to compute the projection onto an arbitrary affine space, our methods excel as a projection function. Recently, Rutkowski, [17], made progress with non-asymptotic parallel projections in a Hilbert space. Where the number of inequality constraints is $r$, we figure the complexity of their algorithm to be $O(2^{r-1}r^3)$ before parallelization, and $O(r^3)$ over $2^{r-1}$ processors. Our method compares favorably with theirs as a function of the number of constraints.

*Contributions of the Paper:* Our methods have distributed complexity. We eliminate common assumptions like the needs for nonempty feasible spaces, a starting feasible point, and a nonempty interior. We develop polyhedral properties to construct easy-to-check, necessary conditions that allow for skipping many of the affine spaces that slow down their forebears. All these reasons will likely lead to the common usage of our convex algorithm on systems capable of large scale multi threading and our non-convex algorithm when even a small amount of multi threading is available and an accurate result is required.

For a quick peek at our algorithm's complexity, let our objective function $f : \mathbb{H} \to \mathbb{R}$, where $\mathbb{H}$ is an $n \in \mathbb{N} \cup \{\infty\}$ dimensional Hilbert space with the standard inner product, $\langle \cdot, \cdot \rangle$, be the projection function, $O(n^3)$, and have $r \in \mathbb{N}$ inequality constraints. If $r >> n$, the complexity comes out to $O(r^{n+1}n^4)$. This complexity result is weaker than the polynomial time of interior point methods reviewed by Polik et al. [16], however when a large number of threads are available to process the problem in parallel, the time complexity of the algorithm is $O(n^4)$, constant as a function of the number of inequalities.

In Sect. 2, we introduce definitions necessary for reading the algorithm. In Sect. 3, we present the algorithm. In Sect. 4, we state and prove the algorithm's foundation. In Sect. 5, we prove that the algorithm works and find its complexity. In Sect. 6, we expand our work to minimization over non-convex polyhedra and present Algorithm 2, the adaptation of Algorithm 1 for non-convex polyhedra.

## 2 Some Definitions

We present a handful of prerequisite definitions before proceeding to our algorithm.

**Definition 2.1** Let $P$ be a **convex polyhedron** and $\mathcal{H}_P$ a finite collection of $r \in \mathbb{N}$ closed half-spaces in $\mathbb{H}$, an $n \in \mathbb{N} \cup \{\infty\}$ dimensional Hilbert space. This lets $P = \bigcap \mathcal{H}_P$, the intersection of the $r$ half spaces in $\mathcal{H}_P$. For all $H \in \mathcal{H}_P$ we define the boundary hyperplane $\partial H$, the vector $\mathbf{n}_H \in \mathbb{H}$ normal to $\partial H$, and $b_H \in \mathbb{R}$ such that $H = \{\mathbf{x} \in \mathbb{H} | \langle \mathbf{x}, \mathbf{n}_H \rangle \leq b_H\}$. For any $H \in \mathcal{H}_P$ we say that $H$ is a half-space of $P$ and $\partial H$ a hyperplane of $P$.

We use the term polyhedron to refer to convex polyhedra. For the non-convex polyhedra we address in Sect. 6, we state their non convexity explicitly.

*Example 2.2* Examples of polyhedra include $\mathbb{H}$, $\varnothing$, $\{42\}$, a rectangle, and a set we'll call the 'A' polyhedron, a simple unbounded example we will use to illustrate more complex ideas later on. 'A'$:= \{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^2 | \mathbf{y} \leq \frac{1}{2}$ and $\mathbf{x} + \mathbf{y} \leq 1$ and $-\mathbf{x} + \mathbf{y} \leq 1\}$. We have $\mathcal{H}_{\text{'A'}} = \{\bar{F}, \grave{G}, \acute{H}\}$ with $\bar{F} := \{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^2 | \mathbf{y} \leq \frac{1}{2}\}$, $\grave{G} := \{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^2 | \mathbf{x} + \mathbf{y} \leq 1\}$, and $\acute{H} := \{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^2 | -\mathbf{x} + \mathbf{y} \leq 1\}$. Both the name of the 'A' polyhedron and the half-space accents were selected for their iconicity to avoid confusion when we come back to this example.

For a convex objective function, $f : \mathbb{H} \to \mathbb{R}$, constrained to a polyhedron, $P$, the minimization algorithm below determines if $P$ is empty, or finds the set $\arg\min_P f$. Throughout the paper we will use $f : \mathbb{H} \to \mathbb{R}$ for an arbitrary convex objective function constrained by an arbitrary polyhedron, $P$.

*Example 2.3* Given some $y \in \mathbb{H}$, let $f(\mathbf{x}) = \|\mathbf{x} - \mathbf{y}\|$. We consider the projection problem $\Pi_P(\mathbf{y}) := \arg\min_P f$. Here, $f$ is strictly convex and the optimal set $\arg\min_P f$ will always have a unique value, Boyd et al. [5].

**Definition 2.4** We say $A$ is an **affine space of** $P$ if it is a nonempty intersection of a subset of $P$'s hyperplanes. We will denote the set of $P$**'s affine spaces** with $\mathcal{A}_P := \{\bigcap_{H \in \eta} \partial H | \eta \subseteq \mathcal{H}_P\} \setminus \{\varnothing\}$. Note that $\mathcal{A}_P$ has at most $\sum_{i=1}^{n} \binom{r}{i} \leq \min(r^n, 2^r)$ elements since the intersection of more than $n$ distinct hyperplanes will be an empty set, or redundant with an intersection of fewer hyperplanes.

*Example 2.5* If $\mathcal{H}_P = \{F, G, H\}$ then $\mathcal{A}_P = \{\mathbb{H}, \partial H, \partial G, \partial F, \partial H \cap \partial G, \partial H \cap \partial F, \partial F \cap \partial G, \partial H \cap \partial G \cap \partial F\}$. If $P \subset \mathbb{R}^3$, $\partial H$ might be a plane, $\partial H \cap \partial G$ a line, and $\partial H \cap \partial G \cap \partial F$ a single point. However, if any of those intersections are empty then they are not included in $\mathcal{A}_P$. We have $\mathbb{H} \in \mathcal{A}_P$ since if we choose $\eta = \varnothing$ then for all $\mathbf{x} \in \mathbb{H}$ we trivially have $\mathbf{x} \in H$ for all $H \in \eta$, therefor $\mathbf{x} \in \bigcap_{H \in \varnothing} H = \mathbb{H}$.

*Example 2.6* Consider the 'A' polyhedron from Example 2.2. It's worth noting that 'A' has an affine space, in this case the point $\partial \grave{G} \cap \partial \acute{H}$, that is disjoint with 'A'. The affine space that is a point at the top of the 'A' is outside of our polyhedron, but still a member of $\mathcal{A}_{\text{'A'}}$. This is a common occurrence.

**Definition 2.7** For $A \in \mathcal{A}_P$, we define the $P$**-cone** of $A$ as $P_A := \bigcap\{H \in \mathcal{H}_P | \partial H \supseteq A\}$, the polyhedron whose hyperplanes, a subset of the hyperplanes of $P$, intersect to equal $A$. These are subsets of $f$'s linear-inequality constraints.

*Example 2.8* We have $\mathbb{H} \in \mathcal{A}_P$, so it is appropriate to note that for a polyhedron, $P$ we have $P_{\mathbb{H}} = \mathbb{H}$. We comment on this here since in the algorithm presented below we will consider the $P$-cone for every $A \in \mathcal{A}_P$.

*Example 2.9* If we use the 'A' polyhedron (2.2), then the 'A'-cone of the top point $\text{'A'}_{\partial \acute{H} \cap \partial \grave{G}} = \acute{H} \cap \grave{G}$. Note that $\bar{F} \cap \grave{G} \cap \acute{H} = \text{'A'} \subset \text{'A'}_{\partial \bar{F} \cap \partial \grave{G}}$.

**Definition 2.10** For $A, B \in \mathcal{A}_P$, we say that $B$ is an **immediate superspace** of $A$ if $B \supsetneq A$ and there exists an $H \in \mathcal{H}_P$ such that $A = \partial H \cap B$. We will also say that $A$ is an **immediate subspace** of $B$. We will denote the set of all of $A$'s superspaces with $\mathcal{B}_A$.

**Algorithm 1:** Finds $\arg\min_P f$

**Input**: A set of half-spaces $\mathcal{H}_P$ and a function $f : \mathbb{H} \xrightarrow{conv.} \mathbb{R}$
**Output**: $\arg\min_P f$
1 **for** $i \leftarrow 0$ *to* $\min(n, r)$ **do**
2     **for** $A \in \mathcal{A}_P$ *with* $\operatorname{codim}(A) = i$ ***in parallel*** **do**
3         **if** $\exists B \in \mathcal{B}_A$ *s.t.* $m_B \cap (P_A \setminus A) \neq \varnothing$ **then**
4             $m_A \leftarrow m_B \cap P_A$
5         **else**
6             $m_A \leftarrow \arg\min_A f$ is computed and saved.
7             **if** $m_A \cap P \neq \varnothing$ **then**
8                 **return** $m_A \cap P$

9 **return** $\arg\min_P f$ is empty.

*Example 2.11* In the 'A' example (2.2). The immediate superspaces of $\partial\grave{G} \cap \partial\acute{H}$ are $\partial\grave{G}$ and $\partial\acute{H}$. The immediate superspace of $\partial\bar{F}$ is $\mathbb{R}^2$. Observe that if an arbitrary $A$ has co-dimension $i$, then its immediate superspaces have co-dimensions $i - 1$.

## 3 The Optimization Algorithm

Algorithm 1 uses the test presented in the *if else* statement on Line 3 to find the optimal point of $f$ in $P$ by iterating over all the affine spaces of $P$ until an affine space $A \in A_P$ that has nonempty $\arg\min_A f \cap P$ is found, and then returns the optimal point courtesy of the black-box method. In Theorem 5.3 below, we guarantee that the algorithm returns $\arg\min_P f$.

In the Algorithm 1, for some $A \in \mathcal{A}_P$ we use $m_A$ as a place to store $\arg\min_{P_A} f$, previously computed with a call to the black-box method.

In the introduction we described the use of a test to determine if an affine space $A \in \mathcal{A}_P$ is the active set of constraints. What we really want to know is, does $\min_A f = \min_P f$? For that matter, does such an $A$ even exist? And if it does, how will the test recognize it?

We prove our results regarding the answers to these questions in Sects. 4 and 5, but we'll work through a couple of examples for finding that $A$ now. Yes, such an $A$ does exist, and when we refer to the test that recognizes that $A$, we're referring to lines 3 and 7. The purpose of these examples is to aid in an intuitive understanding of the algorithm.

*Example 3.1* Consider a polyhedron, $P \subseteq \mathbb{R}^3$, with a typical vertex, $A$, to which we will apply the test, optimizing some strictly convex function, $f$.

When we say that $A$ is a typical vertex, we mean that it's the intersection of three planes. That lets us build $P_A$, a polyhedral cone, as the intersection of the three plane's half spaces.

The test first looks at all the immediate superspaces of $A$. We find each of these by removing one of the three planes. Each of $A$'s three immediate superspace is the intersection of two planes. These lines are the edges of the cone that is $P_A$, and they intersect at $A$. We'll call these lines $B, C$ and $D$. Each one has its own $P$-cone, $P_B$, $P_C$ and $P_D$. These cones are all the intersections of two of $P_A$'s three half spaces.

By the time we arrive at the test for $A$, the algorithm has already computed the optimal points for each of the cones, $P_B$, $P_C$ and $P_D$. Those optimal points were stored respectively as $m_B, m_C$ and $m_D$. Still on Line 3, the test checks if any of those points are in $P_A$. If so, then $A$ is not the active constraint set. This is the fast fail since we don't need to compute $\arg\min_A f$. Suppose, without loss of generality, the test found that $m_C \in P_A$. A nice result of the fast fail is that we now know that $m_C$ is the optimal point of $P_A$. That is, $m_A \leftarrow m_C$, which, if there were more dimensions, would be useful later on.

If all $m_B, m_C$ and $m_D$ are outside of $P_A$, then we progress to the **else** statement now knowing that $\min_{P_A} f = \min_A f$. And that's where the black-box method comes in, because it can compute $\arg\min_A f$. We save that computation as $m_A$ for future use.

There's one last thing to do. We've verified that $m_B, m_C, m_D \in {P_A}^c$, and computed $m_A$. If $m_A \in P$, then $m_A$ is the optimal point over $P$ and the algorithm concludes. If it's not, we move on to apply the test to some other affine space of $P$.

By checking the affine spaces in order of co-dimension, we ensure that we've already done the work on immediate superspaces to set the test up for success.

There are lots of *why* questions to be asked about Example 3.1. Sections 4 and 5 should answer those questions. You can find a complete and detailed run through of Algorithm 1 in Example 3.2.

*Example 3.2* We will revisit Example 2.2 by walking the problem $\Pi_{\text{`}A\text{'}}(1, 1)$ through Algorithm 1. Refer to Fig. 1 throughout this example for your convenience.

We begin Line 1 with $i \leftarrow 0$, setting us up to consider on Line 2 all the affine spaces in $\mathcal{A}_P$ with co-dimension 0. The only such affine space is $\mathbb{H}$, so $A \leftarrow \mathbb{H}$. On Line 3, we note that $\mathbb{H}$ has no immediate superspaces, so $\mathcal{B}_{\mathbb{H}} = \varnothing$, and the condition in the **if**, statement is false. We proceed to the **else** statement and compute $m_{\mathbb{H}} \leftarrow \Pi_{\mathbb{H}}(1, 1) = (1, 1)$. We now check the condition on Line 7 and find $m_{\mathbb{H}}$ as $(1, 1)$ is not in $P$. The condition is false. The inner loop completes an iteration, and with no more affine spaces of co-dimension 0, the inner loop concludes. The outer loop on Line 1 progresses to $i \leftarrow 1$, to look at all of $P$'s affine spaces of co-dimension 1 on Line 2.

There are three affine spaces of co-dimension 1, $\partial\acute{H}$, $\partial\grave{G}$, and $\partial\bar{F}$. Each affine space of co-dimension 1 has the same set of immediate superspaces, $\mathcal{B}_{\partial\acute{H}} = \mathcal{B}_{\partial\grave{G}} = \mathcal{B}_{\partial\bar{F}} = \{\mathbb{H}\}$.

On Line 2, we will arbitrarily look at $A \leftarrow \partial\acute{H}$ first, though ideally all three affine spaces would be considered in parallel. On Line 3, we review every $B \in \mathcal{B}_{\partial\acute{H}} = \{\mathbb{H}\}$ to check if $m_B \in P_{\partial\acute{H}} \setminus \partial\acute{H}$. There's just the one, $m_{\mathbb{H}} = (1, 1)$, so the check is easy.

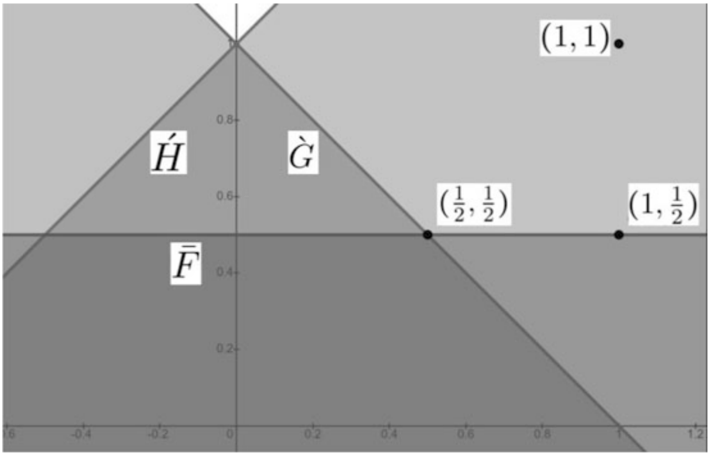


**Fig. 1** Example 3.2

Is $(1, 1) \in P_{\partial \acute{H}} \setminus \partial \acute{H}$? We have $P_{\partial \acute{H}} = H$. Yes, $-1 + 1 < 1$. The condition on Line 3 is true. We proceed to Line 4 and assign $m_{\partial \acute{H}} \leftarrow (1, 1)$. Completing the inner loop iteration for $\acute{H}$, we move onto $A \leftarrow \partial \grave{G}$ and $A \leftarrow \partial \bar{F}$.

For both $A \leftarrow \partial \grave{G}$ and $A \leftarrow \partial \bar{F}$, on Line 3 we have $m_B$ as $(1, 1)$. We check the condition on Line 3. Is $m_B$ as $(1, 1)$ in $\bar{F} \setminus \partial \bar{F}$? Is it in $\grave{G} \setminus \partial \grave{G}$? No. Both $A$ as $\partial \bar{F}$ and $\partial \grave{G}$ go to the **else** statement where we compute $m_{\partial \bar{F}} = \Pi_{\partial \bar{F}}(1, 1) = (1, \frac{1}{2})$ and $m_{\partial \grave{G}} = \Pi_{\partial \grave{G}}(1, 1) = (\frac{1}{2}, \frac{1}{2})$. However, on line 7, different things happen to them. We check $m_{\grave{G}}$ and $m_{\bar{F}}$ for membership in $P$ on Line 7. The point $(1, \frac{1}{2}) \in P^c$, but the point $(\frac{1}{2}, \frac{1}{2}) \in P$, taking $A$ as $\partial \grave{G}$ to the **return** statement on Line 8. We conclude $\Pi_{\text{'A'}}(1, 1) = (\frac{1}{2}, \frac{1}{2})$.

Note that if both conditions on Line 7 had turned out false, we now know $m_{\bar{F}}, m_{\acute{H}}$, and $m_{\grave{G}}$, preparing us for the next iteration of the outer loop where we consider affine spaces of co-dimension $i \leftarrow 2$.

*Remark 3.3* Below, in Theorem 5.11 we present the complexity of Algorithm 1. If the Hilbert space is finite dimensional, uses the standard inner product, $r >> n$, and the black-box method takes $M(n)$ operations, then the complexity of the algorithm is $O(r^n \cdot (r \cdot n + M(n)))$ when run sequentially, and $O(n(n + M(n)))$ when run in parallel.

## 4 Polyhedral Proofs

In this section we present novel necessary and sufficient conditions for an affine-space $A$ to have $\min_A f = \min_P f$ and guarantee $A$'s existence for the case when

$\arg\min_P f \neq \varnothing$. While The Sufficient Criteria (4.14) require the computation $\min_A f$, The Necessary Criteria (4.10) do not. This significantly reduces the number of affine spaces over which we call the black-box method to calculate $\arg\min_A f$.

## 4.1 Preliminary Proofs

**Definition 4.1** For $\mathbf{a}, \mathbf{b} \in \mathbb{H}$, we use $\overset{\longmapsto}{\mathbf{a}, \mathbf{b}}$ to denote the closed **line segment** from **a** to **b** and $\overline{\mathbf{a}, \mathbf{b}}$ to denote the **line** containing **a** and **b**.

We include Lemma 4.2 and 4.3 for the reader's convenience. They are proved in Neimand et al. [13].

**Lemma 4.2** *Let $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{H}$. If $H$ is a half-space such that $\boldsymbol{a} \in H$ and $\boldsymbol{b} \in H^c$, then $\partial H \cap \overset{\longmapsto}{\boldsymbol{a}, \boldsymbol{b}}$ has exactly one point.*

**Lemma 4.3** *Let $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$ be distinct points in $\mathbb{H}$ with $\boldsymbol{b} \in \overset{\longmapsto}{\boldsymbol{a}, \boldsymbol{c}}$.*

1. $\|\boldsymbol{a} - \boldsymbol{b}\| + \|\boldsymbol{b} - \boldsymbol{c}\| = \|\boldsymbol{a} - \boldsymbol{c}\|$
2. $\|\boldsymbol{a} - \boldsymbol{b}\| < \|\boldsymbol{a} - \boldsymbol{c}\|$.
3. *If $f : \mathbb{H} \to \mathbb{R}$ is convex and $f(\boldsymbol{a}) < f(\boldsymbol{c})$ then $f(\boldsymbol{b}) < f(\boldsymbol{c})$.*
4. *If $f : \mathbb{H} \to \mathbb{R}$ is convex and $f(\boldsymbol{a}) \leq f(\boldsymbol{c})$ then $f(\boldsymbol{b}) \leq f(\boldsymbol{c})$.*

**Definition 4.4** We use the following notations. For any $X \subset \mathbb{H}$ we use $\operatorname{aff}(X)$ to denote the **affine hull** of $X$, $B_r(\mathbf{y})$ to denote the open **ball** centered at $\mathbf{y} \in \mathbb{H}$ with a radius of $r \in \mathbb{R}$, $\operatorname{int}(X)$ for the **interior** of $X$, and $\operatorname{relint} X$ to denote the **relative interior** of $X$.

**Lemma 4.5** *Let $K \subseteq P$ be a nonempty convex set and $A$ be the smallest space with regards to inclusion in $\mathcal{A}_P$ such that $K \subseteq A$, and let $y \in \operatorname{relint} K$, then for some $H \in \mathcal{H}_P$ (Def. 2.1), if $y \in \partial H$ then $A \subseteq \partial H$.*

***Proof*** Let $H \in \mathcal{H}_P$ such that $\mathbf{y} \in \partial H \cap \operatorname{relint} K$. There exists an $\epsilon > 0$ and $N := B_\epsilon(\mathbf{y}) \cap \operatorname{aff}(K)$, such that $N \subseteq K \subseteq P \cap A$.

Let us falsely assume $A$ is not a subset of $\partial H$. If $K \subseteq \partial H$, then by the definition of $A$, $A \subseteq \partial H$ in contradiction to the false assumption we just made. Therefor, $K$ is not a subset of $\partial H$ and there exists an $\mathbf{a} \in K \setminus \partial H$. Since $K \subseteq P$ it follows that $\mathbf{a} \in \operatorname{int}(H)$.

Let $t_\epsilon := 1 + \frac{\epsilon}{2\|\mathbf{a}-\mathbf{y}\|} \in \mathbb{R}$ and $\mathbf{y}_\epsilon := (1 - t_\epsilon)\mathbf{a} + t_\epsilon \mathbf{y}$. Observe that $\|\mathbf{y}_\epsilon - \mathbf{y}\| = \|(1 - t_\epsilon)\mathbf{a} + t_\epsilon \mathbf{y} - \mathbf{y}\| = \frac{\epsilon}{2\|\mathbf{a}-\mathbf{y}\|}\|\mathbf{a} - \mathbf{y}\| = \frac{\epsilon}{2}$, giving $\mathbf{y}_\epsilon \in B_\epsilon(\mathbf{y}) \cap \overline{\mathbf{a}, \mathbf{y}}$. Note that any line containing two points in an affine space is entirely in that affine space; since $\mathbf{a}, \mathbf{y} \in \operatorname{aff} K$, we have $\overline{\mathbf{a}, \mathbf{y}} \subseteq \operatorname{aff} K$. Since $\mathbf{y}_\epsilon \in \overline{\mathbf{a}, \mathbf{y}}$, we have $\mathbf{y}_\epsilon \in \operatorname{aff} K$, and we may conclude $\mathbf{y}_\epsilon \in N$.

Let $t_y := (\|\mathbf{a} - \mathbf{y}\| + 2^{-1}\epsilon)^{-1}\|\mathbf{a} - \mathbf{y}\|$. From our earlier definition of $\mathbf{y}_\epsilon$, we have $\mathbf{y}_\epsilon = (-2^{-1}\|\mathbf{a} - \mathbf{y}\|^{-1}\epsilon)\mathbf{a} + 2^{-1}\|\mathbf{a} - \mathbf{y}\|^{-1}(2\|\mathbf{a} - \mathbf{y}\| + \epsilon)\mathbf{y}$. By isolating $\mathbf{y}$ and substituting in $t_y$, we get $\mathbf{y} = (1 - t_y)\mathbf{a} + t_y\mathbf{y}_\epsilon$, giving $\mathbf{y} \in \overset{\longmapsto}{\mathbf{a}, \mathbf{y}_\epsilon}$.

If $\mathbf{y}_\epsilon$ is in $\operatorname{int}(H)$, then by convexity of $\operatorname{int}(H)$, we have $\overline{\mathbf{a}, \mathbf{y}_\epsilon} \subset \operatorname{int}(H)$, including $\mathbf{y}$, a contradiction to $\mathbf{y} \in \partial H$.

If $\mathbf{y}_\epsilon$ is in $\partial H$, we have two points of $\overline{\mathbf{a}, \mathbf{y}}$ in $\partial H$. It follows that $\overline{\mathbf{a}, \mathbf{y}} \subseteq \partial H$ and $\mathbf{a} \in \partial H$, a contradiction.

All that remains is for $\mathbf{y}_\epsilon \in H^c \subseteq P^c$. But $\mathbf{y}_\epsilon \in N$ and $N \subseteq P$, a contradiction. □

**Proposition 4.6** *Let $K \subseteq P$ be a nonempty convex set and $A$ be the smallest space with regards to inclusion in $\mathcal{A}_P$ such that $K \subseteq A$. Then for any $\boldsymbol{x} \in \operatorname{relint} K$ there exists an $\epsilon > 0$ such that $P_A \cap B_\epsilon(\boldsymbol{x}) = P \cap B_\epsilon(\boldsymbol{x})$.*

***Proof*** We may assume that $\mathcal{H}_P$ is nonempty and that $A \neq \mathbb{H}$, otherwise the proof is trivial.

Let $x \in \operatorname{relint} K$. Let $Q \subseteq \mathbb{H}$ be a polyhedron such that $\mathcal{H}_Q = \mathcal{H}_P \setminus \mathcal{H}_{P_A}$. Then we can define $\epsilon := \min_{y \in \partial Q} \|\mathbf{y} - \mathbf{x}\|$. If we falsely assume $\epsilon = 0$, then there exists an $H \in \mathcal{H}_Q$ with $\mathbf{x} \in \partial H \cap P$. Since $\mathbf{x} \in \operatorname{relint} K$, we may conclude from Lemma 4.5 that $A \subset \partial H$ and that $H \in \mathcal{H}_{P_A}$, a contradiction. We may conclude $\epsilon > 0$.

($\subseteq$) Let $\mathbf{y} \in B_\epsilon(\mathbf{x}) \cap P_A$. Let's falsely assume $\mathbf{y} \in P^c$. There exists an $H \in \mathcal{H}_P$ such that $\mathbf{y} \in H^c$. We have $\mathcal{H}_P = \mathcal{H}_Q \,\dot{\cup}\, \mathcal{H}_{P_A}$. Since $\mathbf{y} \in P_A$ it follows that $H \in \mathcal{H}_Q$. Since $\mathbf{x} \in P \subseteq H$, by Lemma 4.2 we may consider the unique $\partial H \cap \overline{x, y}$, and from Lemma 4.3 conclude that $\|\partial H \cap \overline{\mathbf{x}, \mathbf{y}} - \mathbf{x}\| < \|\mathbf{x} - \mathbf{y}\| < \epsilon$, a contradiction to our choice of epsilon. We may conclude that $P_A \cap B_\epsilon(\mathbf{x}) \subseteq P \cap B_\epsilon(\mathbf{x})$.

($\supseteq$) With $P \subseteq P_A$, it follows that $P_A \cap B_\epsilon(\mathbf{x}) \supseteq P \cap B_\epsilon(\mathbf{x})$. □

**Lemma 4.7** *For any convex $K \subset \mathbb{H}$, the set $\arg\min_K f$ is convex.*

Lemma 4.7 is proved in Niemand et al. [13].

## 4.2 *The Necessary Criteria*

**Definition 4.8** If $\arg\min_P f \neq \varnothing$, we define the **min space** of $f$ on $P$ as the smallest $A \in \mathcal{A}_P$ with regards to inclusion that has $\arg\min_P f \subseteq A$. Equivalently, the min space is the intersection of all the hyperplanes of $P$ that contain $\arg\min_P f$. Where $f$ and $P$ are implied, we omit them.

*Remark 4.9* If $\arg\min_P f \neq \varnothing$, then the min space exists and is unique. If there are no hyperplanes of $P$ that contain $\arg\min_P f$, giving $\arg\min_P f \subseteq \arg\min_{\mathbb{H}} f$, then the min space is $\mathbb{H}$.

**Theorem 4.10 (The Necessary Criteria)** *Let $A$ be the min space for some $f$ on $P$, then $A$ meets the Necessary Criteria which are as follows:*

1. $\arg\min_P f \subseteq \arg\min_A f$
2. $\arg\min_A f = \arg\min_{P_A} f$

*Proof (4.10.1)* From Definition 4.8, we have $\min_A f \leq \min_P f$.

Let's falsely assume there exists a $\mathbf{a} \in A$ such that $f(\mathbf{a}) < \min_P f$ and let $\mathbf{x} \in \operatorname{relint} \arg\min_P f$.

By Proposition 4.6, there exists an $\epsilon > 0$ such that $B_\epsilon(\mathbf{x}) \cap P = B_\epsilon(\mathbf{x}) \cap P_A$. The line segment $\overline{\mathbf{a}, \mathbf{x}}$ is entirely in $A \subset P_A$, so we may choose $t_y := 1 - \frac{\epsilon}{2\|\mathbf{a}-\mathbf{x}\|} \in (0, 1)$ so that $\mathbf{y} := (1 - t_y)\mathbf{a} + t_y\mathbf{x} \in \overline{\mathbf{a}, \mathbf{x}} \cap B_\epsilon(\mathbf{x}) \cap P_A$. Since $\mathbf{y} \in \overline{\mathbf{a}, \mathbf{x}}$, by Lemma 4.3.3 we have $f(\mathbf{y}) < f(\mathbf{x}) = \min_P f$. Proposition 4.6 gives $\mathbf{y} \in P$, a contradiction. □

*Proof (4.10.2)* Let's falsely assume that there exists an $\mathbf{x} \in (P_A \setminus A)$ such that $f(\mathbf{x}) \leq \min_P f$, which by Definition 4.8 has $\arg\min_P f \subset A$, and let $\mathbf{y} \in \operatorname{relint} \arg\min_P f$. Then by Proposition 4.6, we can let $\epsilon > 0$ such that $B_\epsilon(\mathbf{y}) \cap P = B_\epsilon(\mathbf{y}) \cap P_A$.

Since $\mathbf{x} \in P_A \setminus A$, it follows from convexity of $P_A$ that $\overline{\mathbf{x}, \mathbf{y}} \setminus \{\mathbf{y}\} \subset P_A \setminus A$. If there was a second point beside $\mathbf{y}$ in $A$, then by the definition of an affine space, $\mathbf{x}$ would be in $A$ as well.

As in 4.10.1, we may choose a $\mathbf{z} \in \overline{\mathbf{x}, \mathbf{y}} \cap B_\epsilon(\mathbf{y}) \subset P \cap P_A$ with a distance of $\frac{\epsilon}{2}$ from $\mathbf{y}$. We have $\mathbf{z} \in P \setminus A$, and by Lemma 4.3, $f(\mathbf{z}) \leq f(\mathbf{y})$. If $f(\mathbf{z}) = f(\mathbf{y})$, this stands in contradiction to $\arg\min_P f \subseteq A$. If $f(\mathbf{z}) < f(\mathbf{y})$, we have a contradiction to $\mathbf{y} \in \arg\min_P f$.

We may conclude that for all $\mathbf{x} \in P_A \setminus A$, $f(\mathbf{x}) > \min_P f$. From 4.10.1, we see that if $\mathbf{x} \in A$, then $f(\mathbf{x}) \geq \min_P f$. Combining these two and the fact that $\arg\min_P f \subseteq A$, we achieve the desired result. □

## 4.3 The Sufficient Criteria

For those affine spaces that meet the necessary criteria (4.10), we next consider The Sufficient Criteria

**Definition 4.11** Let $A, B \in \mathcal{A}_P$, with $A \subsetneq B$. We can say that $B$ **disqualifies** $A$ from $P$, with regards to $f$, if $B$ is the min space of $f$ on $P_A$. If there is no such $B$, then we say $A$ is a **candidate** for $f$ on $P$. Where $f$ and $P$ are implied, they are omitted.

**Lemma 4.12** *The min space is a candidate.*

***Proof*** Let $A, B \in \mathcal{A}_P$ such that $B$ disqualifies $A$. The min space of $P_A$ is $B$. There exists an $\mathbf{x} \in \arg\min_{P_A} f \setminus A$, otherwise $A$ would be the min space over $P_A$ and not $B$. But this is a contradiction to The Necessary Criteria (4.10.2). □

**Lemma 4.13** *If and only if $A \in \mathcal{A}_P$ is a candidate, then* $\arg\min_A f = \arg\min_{P_A} f$.

***Proof*** Let $A$ be a candidate, and falsely assume $\arg\min_A f \neq \arg\min_{P_A} f$. This means $P_A$ has a min space other than $A$, and that min space disqualifies $A$, in contradiction to $A$ being a candidate.

Let $\arg\min_A f = \arg\min_{P_A} f$. Let's falsely assume there exists a $B$ that disqualifies $A$. That means there exists an $\mathbf{x} \in (P_A \setminus A) \cap \arg\min_{P_B} f$ in contradiction to $\arg\min_{P_A} f = \arg\min_A f$. □

**Proposition 4.14 (The Sufficient Criteria)** *Let $A$ be a candidate and $\arg\min_A f \cap P \neq \varnothing$. Then $\arg\min_A f \cap P = \arg\min_P f$.*

***Proof*** Let $A$ be a candidate of $P$ with $\arg\min_A f \cap P \neq \varnothing$.

Let $\mathbf{x} \in \arg\min_{P_A} f \cap P$. Since $P \subseteq P_A$, for all $\mathbf{y} \in P$ we have $f(x) \geq y$. But $\mathbf{x} \in P$ so $\mathbf{x} \in \arg\min_P f$. Therefore, (1) $\arg\min_{P_A} f \cap P \subseteq \arg\min_P f$. Let $\mathbf{x} \in \arg\min_P f$. Since $f(\mathbf{x}) = \min_P f = \min_{P_A} f$ and $\mathbf{x} \in P_A$ we have $\mathbf{x} \in \arg\min_{P_A} f$. That is to say, (2) $\arg\min_P f \subseteq \arg\min_{P_A} f$. We also have (3) $\arg\min_P f \subseteq P$. We can combine the three set inequalities to conclude $\arg\min_{P_A} f \cap P = \arg\min_P f$.

To complete the proof we again recall Lemma 4.13, and note $\arg\min_A f \cap P = \arg\min_{P_A} f \cap P = \arg\min_P f$. □

Let $A \in \mathcal{A}_P$. If for all $B \in \mathcal{A}_P$ with $B \supsetneq A$ we know $\arg\min_B f$, we can use disqualification to determine that $A$ is not the min space, without expensively computing $\arg\min_A f$. Furthermore, if $B$ disqualifies $A$, then we also know $\arg\min_{P_A} f = \arg\min_B f = \arg\min_{P_B} f$ which was previously computed. This is not our fast fail, $A$ may have too many superspaces, but we're getting closer.

*Remark 4.15* If $A$ is the intersection of $m$ hyperplanes of $P$, then it has $m$ immediate superspaces, each can be generated by taking the intersection of $m - 1$ of the hyperplanes that intersect to make $A$. Note that $m < \min(n, r)$ since $A$ can't be the intersection of more than the total number of hyperplanes, or more hyperplanes than there are dimensions.

**Theorem 4.16** *Let $A \in \mathcal{A}_P$, then $A$ is disqualified from $P$, if and only if there exists an immediate superspace, $B$, such that $\arg\min_{P_B} f \cap (P_A \setminus A)$ is nonempty.*

***Proof*** ($\Rightarrow$) Let $A \in \mathcal{A}_P$, such that $A$ is disqualified from $P$.

If $A$ is disqualified by some $B \in \mathcal{B}_A$, then there exists an $\mathbf{x} \in \arg\min_B f \cap P_A \cap A^c$. Since $B$ is a min space for $P_A$, the Necessary Criteria (4.10) give $\arg\min_B f = \arg\min_{P_B} f$ achieving the desired result.

If $A$ is disqualified by some $C$ that is not an immediate superspace of $A$, then $C$ is the min space of $A$, and there exists a $\mathbf{c} \in \arg\min_C f = \arg\min_{P_C} f$ (Theorem 4.10) with $\mathbf{c} \in P_A \setminus A$, such that for all $\mathbf{x} \in P_C$, we have $f(\mathbf{x}) \geq f(\mathbf{c})$. Since $\mathbf{c} \in P_A$ and $P_A \subseteq P_B$ we have $\mathbf{c} \in P_B$. Since $P_B \subset P_C$ we have $\mathbf{c} \in \arg\min_{P_B} f$, the desired result.

($\Leftarrow$) Let $B$ be an immediate superspace of $A$, and let $\mathbf{b} \in \arg\min_{P_B} f \cap (P_A \setminus A)$. Since $P_A \subsetneq P_B$ and $\mathbf{x} \in \arg\min_{P_B} f$, then $\mathbf{b} \in \arg\min_{P_A} f$. The min space of $P_A$ contains $\mathbf{b}$, which is in $A^c$, so that space is not $A$, and therefor disqualifies $A$. □

*Remark 4.17* Let $A \in \mathcal{A}_P$. Proposition 4.16 and its results allow us to determine if $A$ meets the necessary criteria by looking exclusively at $A$'s immediate superspaces, $\mathcal{B}_A$, and their $P$-cones. For any $B \in \mathcal{B}_A$ there exists an $H_B \in \mathcal{H}_P$ such that

$A = \partial H_B \cap B$; if $\arg\min_{P_B} f \cap (H_B \setminus A)$ is nonempty, then $\arg\min_{P_B} f \cap (P_A \setminus A)$ is nonempty, and $A$ is disqualified. We check if any $B \in \mathcal{B}_A$ disqualifies $A$ by confirming $\langle \arg\min_{P_B} f, n_{H_B} \rangle \leq b_H$ is nonempty. If the complexity of computing the inner product is $n$ and $m := \operatorname{codim} A = |\mathcal{B}_A| \leq n$, then when $\arg\min_{P_B} f$ is known for all $B \in \mathcal{B}_A$, Remark 4.15 and Theorem 4.16 let us check the Necessary Criteria for $A$ in $O(m \cdot n)$. This same check, when the inequality holds, yields $\arg\min_{P_A} f$. This is the fast fail.

We can now detail the test method introduced in Sect. 1. If the fast fail is successful for an affine space $A$, then we have the optimal points of the disqualifying set that are in $P_A$ as $\arg\min_{P_A} f$; there is no need for any additional computation. If the fast fail is unsuccessful, then $A$ is a candidate and Lemma 4.13 tells us we can use the black-box method to compute $m_A \leftarrow \arg\min_A f$ where $m_A := \arg\min_{P_A} f$. With the test complete and knowledge of $\arg\min_{P_A} f$, we prepare to apply the test to $A$'s immediate sub-spaces.

This result lends itself to Algorithm 1, wherein we begin by finding the optimum over $\mathbb{H}$, then at each iteration find the optimum of all the $P$-cones of the immediate sub-spaces, until one of those spaces meets the necessary and sufficient criteria.

# 5 Algorithm Proofs and Analysis

## 5.1 *Proof of Function*

**Lemma 5.1** *When the if else statement in Algorithm 1 Line 3 accesses $m_B$ for some $B \in \mathcal{B}_A$ that $m_B$ has already been saved to memory.*

***Proof*** We will prove by induction on the affine space's co-dimension. The base $A$ is $\mathbb{H}$, since it is the only affine space of $P$ with co-dimension 0. The Hilbert space has no immediate superspaces, that is $\mathcal{B}_{\mathbb{H}} = \varnothing$, and therefor $m_B$ for some $B \in \mathcal{B}_{\mathbb{H}}$ is never called. For an affine space with co-dimension $j$, we will assume that all the affine spaces of co-dimension $j - 1$ had their requisite input available. We note that every affine space, $B$ of co-dimension $j - 1$ was put up for review by Line 2, and generated an $m_B$ on Line 4 or Line 6. The superspaces of $A$'s and the minimums over their $P$-cones are all available. □

**Lemma 5.2** *The if else statement on Line 3 goes to the else statement, if and only if $A$ is a candidate.*

***Proof*** Let's assume conditions in the *if* statement are not met and the *else* statement is reached. This means the *if* statement on Line 3 determined that for every immediate superspace, $B \in \mathcal{B}_A$, we have $m_B \cap (P_A \setminus A) = \varnothing$. Equivalently, $\arg\min_{P_B} f \subset (P_A \setminus A)^c$ which by Corollary 4.16 gives $A$ as a candidate.

Let $A$ be a candidate, then the *if* statement on Line 3 will find that for all $B \in \mathcal{B}_A$ we have $m_B \cap (P_A \setminus A) = \varnothing$, and the *else* statement will be reached. □

**Theorem 5.3** *The return set of Algorithm 1 is equal to* $\arg\min_P f$.

***Proof*** By Remark 4.9, if $\arg\min_P f \neq \varnothing$, the min space exists, and by Lemma 4.12 the min space is a candidate. The two *for* loops will iterate over every affine space of $P$ until a candidate is found that meets the sufficient criteria, checked with a true statement on line 7 and a false one on line 3. By Corollary 4.14 the min space meets the Sufficient Criteria (4.14). If $\arg\min_P f$ is nonempty, then a *return* set is guaranteed.

Let $A$ be candidate (see Lemma 5.2) and the *else* statement reached. If Line 7 finds that The Sufficient Criteria (4.14) are met, then the conditions for Proposition 4.14 are satisfied, insuring the algorithm returns $\arg\min_P f$.

If $\arg\min_P f = \varnothing$, then the conditions for The Sufficient Criteria (4.14) are never met and the *if* statement on Line 7 will reject every $A$. Once all the affine spaces have been reviewed, the final *return* statement is called and an empty set is returned. □

*Example 5.4* Referring back to Example 3.2, $\partial \grave{G}$, whose minimum is the minimum for 'A' is not the min space; $\partial \grave{G} \cap \partial \bar{F}$ is. However $\partial \grave{G}$ is a candidate and The Sufficient Criteria are met. What the min space definition gives us is that if a minimum exists, we can find its min space. But our set of candidates that meet the Sufficient Criteria is broader.

Proposition 4.14 insures that, in spite of the algorithm not having found the min space, $\arg\min_P f$ is still returned.

## 5.2 *Complexity*

**Lemma 5.5** *If $f$ is strictly convex, then for any convex $K$,* $\arg\min_K f$ *has at most one element.*

We will limit the scope of this complexity analysis to strictly-convex $f$. This significantly simplifies our work and implementation of the algorithm by insuring that each $m_B$ in Algorithm 1 has a single element. Computing weather $m_B \cap P_A = \varnothing$ then becomes $m_B \in P_A$.

**Definition 5.6** For clarity, we use brackets to indicate the computational complexity of a process, as a function of $n$ and possibly some $\epsilon > 0$. Thus $[\langle \cdot, \cdot \rangle]$ is the number of steps it takes to compute inner product, ranging from $n$ to $n^3$ for finite inner products and likely a function of $\epsilon$ for infinite Hilbert spaces. For some affine space $A$, we have $[\arg\min_A f]$ as the number of steps it takes to compute our black-box method.

**Corollary 5.7** *Checking if $m_B \cap P_A \neq \varnothing$ on Line 3 has the same complexity as computing inner product, $O([\langle \cdot, \cdot \rangle])$.*

***Proof*** This is a direct result of Remark 4.17. □

**Lemma 5.8** *Checking if* $\exists B \in \mathcal{B}_A$ *s.t.* $m_B \cap (P_A \setminus A) \neq \varnothing$ *on Line 3 has* $O(\min(n, r) \cdot [\langle\cdot, \cdot\rangle])$ *sequential computational complexity and* $O([\langle\cdot, \cdot\rangle])$ *time complexity if run in parallel over* $min(n, r)$ *processors.*

***Proof*** By Corollary 5.7, Checking $m_B \cap P_A \neq \varnothing$ has complexity $O([\langle\cdot, \cdot\rangle])$. A loop checks this once for each $B \in \mathcal{B}_A$, with Remark 4.15 giving $|\mathcal{B}_A| \leq \min(n, r)$. Each $B \in \mathcal{B}_A$ can be checked for $m_B \cap (P_A \setminus A) \neq \varnothing$ independently of one another, so they can all be checked in parallel. □

**Lemma 5.9** *The if statement on Line 7 is* $O(r \cdot [\langle\cdot, \cdot\rangle])$ *sequential computational complexity and* $O([\langle\cdot, \cdot\rangle])$ *when run in parallel over r processors.*

***Proof*** Checking if a point is in $P$ requires checking that the point is in each $H \in \mathcal{H}_P$. Checking if a point is in a half-space is $O([\langle\cdot, \cdot\rangle])$ and since these $r$ checks are independent of one another, they can be done in parallel. □

**Lemma 5.10** *Running the entire if else statement that begins on Line 3 has* $O(r \cdot [\langle\cdot, \cdot\rangle] + [\arg\min_A f])$ *sequential computational complexity, or* $O([\langle\cdot, \cdot\rangle] + [\arg\min_A f])$ *time complexity if run in parallel over r processors.*

***Proof*** We saw in Lemma 5.8 the *if* statement's complexity. If there is no fast fail, the *else* portion computes $\arg\min_A f$.

The inner *if* statement on Line 7 is $O(r)$, so adding these three components we get $O(\min(n, r) \cdot [\langle\cdot, \cdot\rangle] + [\arg\min_A f] + r \cdot [\langle\cdot, \cdot\rangle])$ computational complexity. In simplifying, note that $\min(n, r) \leq r$.

For the parallel case, we have, $O([\langle\cdot, \cdot\rangle] + [\arg\min_A f] + [\langle\cdot, \cdot\rangle])$, which also simplifies to the desired expression.

The same $r$ threads that are used on Line 3 can be used again on Line 7, so there's no need for more than $r$ processors. □

**Theorem 5.11** *Algorithm 1 has* $O(\min(r^n, 2^r) \cdot (r \cdot [\langle\cdot, \cdot\rangle] + [\arg\min_A f]))$ *sequential computational complexity, and* $O(\min(n, r) \cdot ([\langle\cdot, \cdot\rangle] + [\arg\min_A f]))$ *time complexity when run in parallel over* $O(\min(r^{\frac{1}{2}} \cdot 2^{r+\frac{1}{2}}, r^{n+1}))$ *processors.*

***Proof*** For computational complexity we note that the two *for* loops in Algorithm 1 iterate over all the affine spaces in $\mathcal{A}_P$, so we multiply our results from Lemma 5.10 by $|\mathcal{A}_P|$.

For the parallel case, the outer loop cannot be run in parallel. The inner can. The number of iterations for the inner loop, for any $i \leq \min(r, n)$ is $\binom{r}{i}$, because each affine space of co-dimension $i$ is the intersection of $i$ hyperplanes of $P$. Consequently, with $\max_{i<\min(r,n)} \binom{r}{i}$ processors, the inner loop approaches $O(1)$ parallel time complexity. The number of iterations of the outer loop is $\min(n, r)$.

We note that $r$ is a maximum number of iterations for the outer loop since the co-dimension of an affine space $A \in \mathcal{A}_P$ is the number of hyperplanes that intersect to make $A$. That number of hyperplanes, and therefore the co-dimension, cannot exceed the number of $P$'s hyperplanes, $r$. We have $n$ as a maximum because the intersection of more than $n$ hyperplanes will be an empty set or redundant with the intersection of fewer hyperplanes.

All that remains is to compute $\max_{i \le \min(n,r)} \binom{r}{i}$. If $n > \frac{r}{2}$, Pascal's triangle tells us that we have the maximum at $i = \frac{r}{2}$, the Central Binomial Coefficient. Stirling's formula [18] tells us $\binom{r}{\frac{r}{2}} \sim (\pi r)^{-\frac{1}{2}} 2^{r+\frac{1}{2}}$. If $n < \frac{r}{2}$, then the maximum number of processors for the inner loop becomes $\binom{r}{n} \le r^n$. This puts the total number of processors for the inner loop at $O(\min(r^{-\frac{1}{2}} \cdot 2^{r+\frac{1}{2}}, r^n))$.

Multiplying by the number of processors we need for the *if else* statement gives us the desired result. □

When $r >> n$ we have polynomial sequential complexity as a function of $r$, and parallel complexity that's constant using a polynomial number of threads, as function of $r$. When $n >> r$ then sequential and parallel complexities, as well as the number of processors, as a function of $n$ are the complexity of the black-box method plus the inner product method.

Note that unlike many interior point methods, the complexity is not a function of accuracy; outside of the black-box method, there is no $\epsilon$ term that compromises speed with the desired distance from the correct answer.

## 6 Non-Convex Polyhedra

This section expands the results of the previous section to conclude with a multi-threaded algorithm for computing the global minimum in the case of non-convex polyhedral constraints. Since the algorithm is not dependent on a starting feasible point, we find all the local optimum as they meet the necessary criteria, and the optimal of the points that meet the necessary criteria is the global optimum. Our non-convex constraints algorithm exploits the representation of non-convex polyhedra to achieve faster results than the convex algorithm presented above.

We will work with the description from [8] for non-convex polyhedra, where the polyhedron is represented by its faces, where each face, a convex polyhedron itself, has knowledge of its own faces and its neighbors. Together with the definition of non-convex polyhedra in [9], we define a non-convex polyhedron as follows.

**Definition 6.1** A non-convex polyhedron $P \subset \mathbb{R}^n$ is the union of a set of convex polyhedra, $\mathcal{P}$. Namely, $P = \bigcup \mathcal{P}$. We denote the set of faces of $P$ with $\mathcal{F}_P$ and include $P \in \mathcal{F}_P$ as the lone exception to the requirement that $P$'s faces be convex. Note that $\mathcal{F}_P$ is closed to intersections.

**Definition 6.2** We can redefine $P$'s affine spaces, $\mathcal{A}_P$ so that $\mathcal{A}_P = \{A | \forall \mathcal{P} \exists Q \in \mathcal{P}$, with $A \in \mathcal{A}_Q$ and $\exists F \in \mathcal{F}_P$ such that $\operatorname{aff} F = A\} \cup \{\mathbb{R}^n\}$.

**Lemma 6.3** *If $P$ is convex, then $\mathcal{A}_P$ under Definition 6.2 is a subset of $\mathcal{A}_P$ under Definition 2.4, and that subset includes every affine space that has a non empty intersection with $P$.*

***Proof*** Let $A \in \mathcal{A}_P$ for Definition 6.2. Then there exists some $Q \in \mathcal{P}$ and $F \in \mathcal{F}_P$ so that $\operatorname{aff} F = A$. Each $n-1$ dimensional face in $\mathcal{F}$ has $\operatorname{aff} F = \partial H$ for some $H \in \mathcal{H}_P$, and each lower dimensional face is an intersection of those hyperplanes. We may conclude that $A \in \mathcal{A}_P$ for Definition 2.4 since it is the intersection of hyperplanes of $P$. The intersection of $A$ and $P$ is nonempty since $A$ contains a face of $P$. □

Though $\partial P \subseteq \bigcup \mathcal{A}_P$, in many cases, $\mathcal{A}_P$ under Definition 6.2 is substantially smaller than it is under Definition 2.4. Definition 6.2 excludes affine spaces that have an empty intersection with $P$. The pruning is possible because of the additional information in our non-convex polyhedral representation.

We use the following result to construct $\mathcal{A}_P$, Definition 6.2.

**Lemma 6.4** *A necessary condition for a set of $n-1$-dimensional faces $\phi \subseteq \mathcal{F}_P$ to have* $\operatorname{aff}(\bigcap_{F\in\phi} F) \in \mathcal{A}_P$ *is that the angles between every pair of faces in $\phi$ is less than* 180 *degrees.*

***Proof*** Let $F, G \in \phi$ with the angle between them greater than 180 degrees, we can choose a point $\mathbf{x} \in \operatorname{int}(F)$ so that the angle between $\overline{\mathbf{x}, \Pi_{F\cap G}(\mathbf{x})}$ and $\overline{\Pi_G(\mathbf{x}), \Pi_{F\cap G}(\mathbf{x})}$ is greater than 180 degrees. While $\mathbf{x} \in P$ and $\Pi_G(\mathbf{x}) \in P$ the line $\overline{\mathbf{x}, \Pi_G(\mathbf{x})}$, excluding its endpoints, is outside of $P$. There is no convex set with faces $F$ and $G$, and therefor it is possible to construct an arrangement for $\mathcal{P}$ without the affine space. □

We can restrict the elements of $\mathcal{A}_P$ because an optimal point $\mathbf{x}$ over $P$ is also the optimal point over some polyhedron $Q \in \mathcal{P}$, and therefore it can be found with the necessary criteria by looking at all the affine spaces of $Q$ that contain faces of $P$.

Algorithms exist for decomposing non-convex polyhedra into their convex components, [2], however we achieve better results by maintaining the non-convex form. By iterating over $\mathcal{A}_P$ from definition 6.2, we iterate over every face of each polyhedron in $\mathcal{P}$ that might contain $P$'s optimal point.

**Corollary 6.5** *Let $G \in \mathcal{P}$, then if the optimal point $x$ of $P$ has $x \in G$, either $x \in \arg\min_{\mathbb{R}^n} f$ or $x \in \partial P$.*

***Proof*** We may consider the more general statement: If $\mathbf{x}$ is an optimal point of $P$, then $\mathbf{x} \in \arg\min_{\mathbb{R}^n} f$ or $\mathbf{x} \in \arg\min_{\partial P} f$ which is a direct result of the convexity of $f$. □

For purposes of checking the necessary criteria, we need to define the $P$-cone of an affine space, $A \in \mathcal{A}_P$, where $P$ is non convex. The natural choice is to find a convex $Q \in \mathcal{P}$ and use $Q_A$. However, since we don't know the composition of $\mathcal{P}$, we need a practical way to build $P_A$. We do this exactly as we did in Algorithm 1.

**Definition 6.6** If $A \in \mathcal{A}_P$, then there exists an $F \in \mathcal{F}_P$ such that $\operatorname{aff} F = A$. Every such $F$ is the intersection $n-1$ dimensional faces, $\phi \subseteq \mathcal{F}_P$ such that $F = \bigcap \phi$. For each $G \in \phi$ we have an $H_G \in \mathcal{H}_P$ such that $\partial H_G = \operatorname{aff} G$. Then $P_A = \bigcap_{G\in\phi} H_G$.

**Lemma 6.7** *If $P$ is convex, then Definition 6.6 is equivalent to Definition 2.7.*

*Remark 6.8* Let $Q, R$ be convex polyhedra with $A \in \mathcal{A}_Q \cap \mathcal{A}_R$ and $\mathcal{H}_{Q_A} = \mathcal{H}_{R_A}$, then if $A$ meets the Necessary Criteria 4.10 for $Q$, it also does for $R$. That is to say, the elements of $\mathcal{P}$ don't matter, only the neighborhood of $A$.

**Definition 6.9** We redefine a min space and say that $A \in \mathcal{A}_P$ is a min space on a non-convex polyhedron, $P$, if there is a convex polyhedron $Q \subseteq P$ such that $A$ is a min space on $Q$.

Existence of a min space (Definition 6.9) is immediate from the definition of a non-convex polyhedron, though unlike in Definition 4.8, it is not unique. The following corollary follows.

**Corollary 6.10** *Each min space (Definition 6.9) meets The Necessary Criteria 4.10.*

***Proof*** The necessary conditions for a space to be a min space remain the same, because for any $\mathbf{x} \in \arg\min_P f$ we have a $Q \in \mathcal{P}$ so that $\mathbf{x} \in \arg\min_Q f$. □

This means that if some $A \in \mathcal{A}_P$ meets the Necessary Criteria (4.10), exactly which $Q \in \mathcal{P}$ it's in doesn't matter.

The sufficient conditions, checking if $\mathbf{x} \in P$ change a bit. We don't know the polyhedra of $\mathcal{Q}$ and it will not work to check if the point is in all of the half spaces of $P$, since $P$ is not necessarily the intersection of half spaces. We therefor do not check The Sufficient Criteria (4.14).

**Proposition 6.11 (The Sufficient Criteria for a Non-convex Polyhedron)** *Let $\mathcal{M}$ be the set of affine spaces that are candidates and have that for each $A \in \mathcal{M}$ there exists an $F \in \mathcal{F}_P$ such that* $\operatorname{aff} F = A$ *with* $\arg\min_A f \in F$, *then* $\arg\min_P f = \arg\min\{f(\boldsymbol{x}) | \boldsymbol{x} \in \bigcup \mathcal{M}\}$.

***Proof*** Let $\mathbf{x} \in A \in \mathcal{M}$, then by the assumptions set above, $\mathbf{x} \in P$.

Remark 6.10 gives us $\arg\min_P f = \arg\min\{f(\mathbf{x}) | \mathbf{x} \in P$ and $\mathbf{x} \in \arg\min_A f$ where $A$ meets the Nec. Criteria $\}$. □

Since the minimum on the right hand side of the equation is a taken from a finite set, it's easy to compute.

*Remark 6.12* We have $P \in \mathcal{F}_P$, often with $\operatorname{aff} P = \mathbb{R}^n \in \mathcal{A}_P$. If $\mathbb{R}^n \in \mathcal{M}$, we can check $\arg\min_{\mathbb{R}^n} f$ for membership in $P$ with an algorithm like the one in Akopyan et al. [1]. For checking membership in any other $F \in \mathcal{F}_P$, we note that $F$ is a convex polyhedron. Checking membership in a $F$ is substantially faster than checking membership $P$.

With the curated $\mathcal{A}_P$, and the adjusted membership test, Algorithm 1 may proceed as above, except that when a point is found to be in $P$, it is saved and the algorithm continues. On completion, the minimum of all the points that have been saved is the minimum of $P$. If the set of saved points is empty, there is no minimum. For details, see Algorithm 2.

**Algorithm 2:** Finds $\arg\min_P f$ for a non-convex polyhedron $P$

**Input**: A set of faces $\mathcal{F}_P$ and a function $f : \mathbb{R}^n \xrightarrow{conv.} \mathbb{R}$
**Output**: $\min_P f$
$\mathcal{M} \leftarrow \varnothing$
**for** $i \leftarrow 0$ *to* $\min(n, r)$ **do**
  **for** $A \in \mathcal{A}_P$ *with* $\operatorname{codim}(A) = i$ ***in parallel*** **do**
    **if** $\exists B \in \mathcal{B}_A$ *s.t.* $m_B \cap (P_A \setminus A) \neq \varnothing$ **then**
      $m_A \leftarrow m_B \cap P_A$
    **else**
      $m_A \leftarrow \arg\min_A f$ is computed and saved.
      Let $F \in \mathcal{F}$ such that aff $F = A$
      **if** $m_A \cap F \neq \varnothing$ **then**
        add $m_A$ to $\mathcal{M}$.
**return** $\arg\min\{f(x) | x \in \bigcup \mathcal{M}\}$

## References


1. Akopyan, A., Bárány, I., Robins, S.: Algebraic vertices of non-convex polyhedra. Adv. Math. **308**, 627–644 (2017). Available from: https://doi.org/10.1016/j.aim.2016.12.026
2. Bajaj, C.L., Dey, T.K.: Convex decomposition of polyhedra and robustness. SIAM J. Comput. **21**(2), 339–364 (1992). Available from: https://doi.org/10.1137/0221025
3. Balakrishnan, A.V.: Introduction to Optimization Theory in a Hilbert Space. Lecture Notes in Operations Research and Mathematical Systems, vol. 42. Springer, Berlin (1971)
4. Bauschke, H.H., Combettes, P.L.: Convex Analysis and Monotone Operator Theory in Hilbert Spaces. CMS Books in Mathematics/Ouvrages de Mathématiques de la SMC. Springer, New York (2011). With a foreword by Hédy Attouch. Available from: https://doi.org/10.1007/978-1-4419-9467-7
5. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004). Available from: https://doi.org/10.1017/CBO9780511804441
6. Debnath, L., Mikusiński, P.: Introduction to Hilbert Spaces with Applications, 2nd edn. Academic, San Diego (1999)
7. Diamond, S., Takapoui, R., Boyd, S.: A general system for heuristic minimization of convex functions over non-convex sets. Optim. Methods Softw. **33**(1), 165–193 (2018). Available from: https://doi.org/10.1080/10556788.2017.1304548
8. Edelsbrunner, H.: Algebraic decomposition of non-convex polyhedra. In: 36th Annual Symposium on Foundations of Computer Science (Milwaukee, WI, 1995), pp. 248–257. IEEE Comput. Soc. Press, Los Alamitos (1995). Available from: https://doi.org/10.1109/SFCS.1995.492480
9. Edelsbrunner, H.: Geometry and Topology for Mesh Generation, vol. 7. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, Cambridge (2006). Reprint of the 2001 original
10. Frank, M., Wolfe, P.: An algorithm for quadratic programming. Naval Res. Logist. Quart. **3**, 95–110 (1956). Available from: https://doi.org/10.1002/nav.3800030109
11. Gasnikov, A., Kabanikhin, S., Mohammed, A., Shishlenin, M.: Convex optimization in Hilbert space with applications to inverse problems. arXiv preprint arXiv:1703.00267 (2017)
12. Laiu, M.P., Tits, A.L.: An infeasible-start framework for convex quadratic optimization, with application to constraint-reduced interior-point methods. arXiv (2019). Available from: https://arxiv.org/abs/1912.04335

13. Neimand, E.D., Sabau, S.: Polyhedral constrained optimization using parallel reduction of linear inequality to equality constraints. arXiv (2021). Available from: https://arxiv.org/abs/2109.07565
14. Okelo, N.B.: On certain conditions for convex optimization in Hilbert spaces. Khayyam J. Math. **5**(2), 108–112 (2019). Available from: https://doi.org/10.22034/kjm.2019.88084
15. Plesník, J.: Finding the orthogonal projection of a point onto an affine subspace. Linear Algebra Appl. **422**(2–3), 455–470 (2007). Available from: https://doi.org/10.1016/j.laa.2006.11.003
16. Pólik, I., Terlaky, T.: Interior point methods for nonlinear optimization. In: Nonlinear Optimization, vol. 1989. Lecture Notes in Math., pp. 215–276. Springer, Berlin (2010). Available from: https://doi.org/10.1007/978-3-642-11339-0_4
17. Rutkowski, K.E.: Closed-form expressions for projectors onto polyhedral sets in Hilbert spaces. SIAM J. Optim. **27**(3), 1758–1771 (2017). Available from: https://doi.org/10.1137/16M1087540
18. Sloane, N.J.A.: Sequence a000984 central binomial coefficients. Available from: https://oeis.org/A000984
19. user2566092 (https://math.stackexchange.com/users/87313/user2566092). Find point on polyhedron nearest given point. Mathematics Stack Exchange. https://math.stackexchange.com/q/1134236 (version: 2015-02-05)
20. Wright, S.J.: Primal-Dual Interior-Point Methods. Society for Industrial and Applied Mathematics (1997). Available from: https://epubs.siam.org/doi/abs/10.1137/1.9781611971453