A RANDOMIZED CUTTING PLANE METHOD WITH PROBABILISTIC GEOMETRIC CONVERGENCE*

F. DABBENE†, P. SHCHERBAKOV‡, AND B. POLYAK§

Abstract. We propose a randomized method for general convex optimization problems; namely, the minimization of a linear function over a convex body. The idea is to generate N random points inside the body, choose the best one and cut the part of the body defined by the linear constraint. We first analyze the convergence properties of the algorithm from a theoretical viewpoint, i.e., under a rather classical assumption that an algorithm for uniform generation of random points in the convex body is available. Under this assumption, the expected rate of convergence for such method is proved to be geometric. Moreover, explicit sample size results on convergence are derived. In particular, we compute the minimum number of random points that should be generated at each step in order to guarantee that, in a probabilistic sense, the method converges at least as fast as the deterministic center-of-gravity algorithm. From a practical viewpoint, the method can be implemented using Hit-and-Run versions of Markov-chain Monte Carlo algorithms, and we show how these convergence results can be extended to a Hit-and-Run implementation. A crucial notion for the Hit-and-Run implementation is that of Boundary Oracle, which is available for most optimization problems including LMIs and many other kinds of constraints. Preliminary numerical results for SDP problems are presented confirming that the randomized approach might be competitive to modern deterministic convex optimization methods.

Key words. Convex Optimization, Randomized Algorithms, Hit and Run, Linear Matrix Inequalities

AMS subject classifications. 90C25, 65K05, 65K10, 90C15

1. Introduction. Recent years exhibited a growing interest into randomized algorithms in computer science, control and optimization; e.g., see [24, 36, 7]. There are numerous reasons for this interest, from philosophical to computational ones. The present paper continues this line of research for convex optimization. In particular, we consider a convex minimization problem of the form

$$\min c^\top x \quad \text{subject to} \quad x \in \mathcal{X}, \tag{1.1}$$

where the set $\mathcal{X} \subset \mathbb{R}^n$ is a *convex body*, i.e., it is full-dimensional and bounded. Specifically, we assume that there exist two full-dimensional Euclidean balls $\mathcal{B}_r$ and $\mathcal{B}_R$, of radii $0 < r < R$, such that $\mathcal{B}_r \subseteq \mathcal{X} \subseteq \mathcal{B}_R$. The linear cost function is taken without loss of generality, since any convex optimization problem can be reduced to this form.

The use of a cutting plane scheme for the solution of convex optimization problems as the one above dates back to 1960 [17]. A center of gravity version of cutting plane, for a slightly different problem formulation, was proposed almost simultaneously by Levin and Newman in 1965 [19, 26]. This methods can be viewed as the natural extension to general convex multi-dimensional functions of the bisection algorithm for one-dimensional unimodal functions, and has been proved to have extremely appealing features: it works also when the set $\mathcal{X}$ is described by nonsmooth functions, and its theoretical convergence is among the best known. Unfortunately however, despite its very attractive theoretical behavior, the center of gravity algorithm has not been used in practice for a very simple reason: *For a generic convex set, computing the center of gravity turns out to be more difficult than solving the original optimization problem.*

As a result, the use of different types of centers, instead of the geometric center, have been proposed in the literature. Most of the methods proposed for circumventing this problem can

*This work was partially supported by a bilateral project between the Italian National Research Council (CNR) and the Russian Academy of Sciences (RAS). The support of RSTL-CNR found is also acknowledged.

†IEIIT-CNR, Politecnico di Torino, Italy; (fabrizio.dabbene@polito.it)

‡Institute of Control Science, RAS, Moscow, Russia; (sherba@ipu.ru)

§Institute of Control Science, RAS, Moscow, Russia; (boris@ipu.ru)

be classified as *exterior-point* methods, in that they use the center of a specific *localization set*, which is updated at each step and is guaranteed to always contain the feasible set. First of these techniques is the famous ellipsoid method developed by Shor, Yudin, Nemirovsky in 1970s [33, 39], and used in 1979 by Khachiyan to show polynomial solvability of LPs [18]. The localization set in this case is represented by an ellipsoid inscribing $\mathcal{X}$. Evolutions of this approach lead to methods based on the analytic center of a polytope inscribing $\mathcal{X}$, see [13] and references therein for an excellent survey.

In recent years, the interest on designing implementable methods based on center of gravity has gained new interest motivated by the randomized scheme proposed in [3]. In this latter work, the authors compute at each step an *approximate* center of gravity of an outer polytope inscribing $\mathcal{X}$, based on points randomly extracted in the polytope. In the present work, we move a step further and consider the possibility of randomly extracting points *directly* in the feasible set $\mathcal{X}$. This idea has been inspired by the recent results in [28].

The paper is structured as follows. In Section 2, we review the classical deterministic center-of-gravity (DCG) method of [19, 26] and analyze its properties. The rate of convergence of the DCG method has been estimated via Grünbaum theorem (see below) and happened to be geometric. We provide another result based on Radon theorem [31] which also guarantees geometric convergence. In Section 3 we introduce a randomized cutting plane (RCP) scheme based on the generation of random points in the feasible set. Namely, at each iteration, N points are generated, the best of them x_k (corresponding to $\min c^\top x$) is chosen and a cut of the half-space $\{x : c^\top x \leq c^\top x_k\}$ is performed.

In the sections to follow, we analyze the probabilistic properties of the algorithm under the quite standard assumption that a mechanism is available for generating points uniformly in a convex body. First, in Section 4.1, we explicitly compute the *expected rate* of convergence of the RCP algorithm; notably, for $N = 1$ this rate coincides with that of the DCG algorithm, while it is shown to improve by a factor of $\ln(N + 1)$ for $N > 1$. Then, in Sections 4.2.1–4.2.2, a novel and different type of probabilistic analysis of the RCP algorithm is provided. In particular, we estimate the *probability* that RCP converges slower than DCG and calculate the number N of points to be generated in RCP to guarantee a better convergence with prescribed (high) probability.

The second part of the paper discusses the practical implementation of the algorithm, which is based on Hit-and-Run (H&R) versions of the Monte Carlo method, — the procedure aimed at *approximately uniform* generation of points in a body via random walks.

The application of H&R to convex optimization has been first proposed in [3]. Note that the method in [3] differs from the implementable version of RCP presented in Section 5 in the following crucial aspects:

i) the problem formulation in [3] differs from (1.1), and it is this difference that allows us to introduce the concept of *best point*;
ii) the method proposed in [3] updates at each step an outer polytope inscribing $\mathcal{X}$, while here we update directly the set $\mathcal{X}$;
iii) in [3], the authors use a so-called Separation Oracle, while we exploit here a Boundary Oracle for the set $\mathcal{X}$.

Notice that in view of i)-ii), our method can be considered an interior-point algorithm, while the method in [3] in fact represents a classical exterior-point localization scheme.

The results of numerical experiments with the proposed algorithm for semidefinite programming problems are described in Section 6.

Preliminary results obtained in the two conference papers [8], [9] serve as a basis for the exposition to follow.

2. Deterministic Center of Gravity Algorithm. For a convex body $\mathcal{X}$, define its *center of gravity* as

$$\mathsf{cg}(\mathcal{X}) \doteq \frac{\int_{\mathcal{X}} x \mathrm{d}x}{\int_{\mathcal{X}} \mathrm{d}x}.$$

Then, the deterministic cutting plane method based on recursive cutting of the feasible set through the center of gravity can be stated as follows, see also [8, 28].

Algorithm 1 (DCG algorithm)

Input: $\mathcal{X}$

Output: x^*

1: $k \Leftarrow 0$, $\mathcal{X}_k \Leftarrow \mathcal{X}$;
2: $x_k \Leftarrow \mathsf{cg}(\mathcal{X}_k)$;
3: $\mathcal{X}_{k+1} \Leftarrow \{x \in \mathcal{X}_k : c^\top(x - x_k) \leq 0\}$;
4: check Stopping Rule; goto 2.

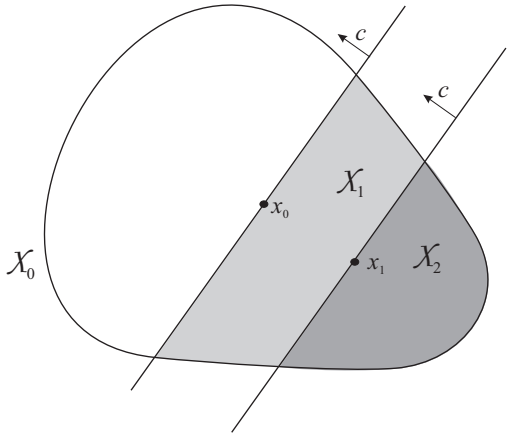

FIG. 2.1. *Illustration of the DCG algorithm.*

The behavior of the deterministic center of gravity cutting plane is illustrated in Figure 2.1. The method works as follows. Let $\mathcal{X}_0 \equiv \mathcal{X}$, and let x_0 be the center of gravity of $\mathcal{X}_0$. Then, proceed by considering the new set

$$\mathcal{X}_1 \doteq \{x \in \mathcal{X}_0 : c^\top(x - x_0) \leq 0\} \subset \mathcal{X}_0$$

obtained by cutting off a portion of $\mathcal{X}_0$ using the hyperplane

$$\mathcal{H}_0 \doteq \{x \in \mathbb{R}^n : c^\top(x - x_0) = 0\}.$$

For this convex set $\mathcal{X}_1$ in turn, compute its center of gravity $x_1 = \mathsf{cg}(\mathcal{X}_1)$ and construct the hyperplane $\mathcal{H}_1$ passing through x_1, which defines the set $\mathcal{X}_2 \subset \mathcal{X}_1$, etc. As a result, we obtain a sequence of embedded sets $\mathcal{X}_k \subset \mathcal{X}_{k-1} \subset \cdots \subset \mathcal{X}_1 \subset \mathcal{X}_0$ and a sequence of points x_k having the property

$$c^\top x_{k+1} < c^\top x_k.$$

The convergence analysis of this algorithm can be based on the following proposition due to Grünbaum [15].

PROPOSITION 2.1 (Grünbaum). *Let $\mathcal{X} \in \mathbb{R}^n$ be a convex body and let $x_{\mathrm{G}} = \mathsf{cg}(\mathcal{X})$ be its center of gravity. Consider any hyperplane $\mathcal{H} = \left\{x \in \mathbb{R}^n \,:\, c^\top(x - x_{\mathrm{G}}) = 0\right\}$ passing through x_{G}. This plane divides the set $\mathcal{X}$ in the two subsets*

$$\begin{aligned}\mathcal{X}_1 &= \{x \in \mathcal{X} \,:\, c^\top x > c^\top x_{\mathrm{G}}\},\\ \mathcal{X}_2 &= \{x \in \mathcal{X} \,:\, c^\top x \leq c^\top x_{\mathrm{G}}\}.\end{aligned}$$

Then, the following relations hold for $i = 1, 2$

$$\mathrm{vol}\,(\mathcal{X}_i) \;\leq\; \left(1 - \left(\frac{n}{n+1}\right)^{1/n}\right) \mathrm{vol}\,(\mathcal{X}) \;\leq\; (1 - 1/\mathrm{e})\,\mathrm{vol}\,(\mathcal{X}). \tag{2.1}$$

From this proposition, it follows that each step of the DCG algorithm guarantees that a given portion of the feasible set is cut out, namely $\mathrm{vol}\,(\mathcal{X}_{k+1}) \leq (1-1/\mathrm{e})\mathrm{vol}\,(\mathcal{X}_k)$. A recursive application of this inequality immediately leads to the volume inequality

$$\mathrm{vol}\,(\mathcal{X}_k) \leq (1 - 1/\mathrm{e})^k \mathrm{vol}\,(\mathcal{X}_0) \approx (0.63)^k \mathrm{vol}\,(\mathcal{X}_0) \tag{2.2}$$

which proves that DCG has guaranteed geometric convergence in terms of *volumes*. Notice that the volume reduction in (2.2) is completely independent of all problem parameters, including the dimension n.

A different type of convergence analysis of the DCG algorithm, which provides an estimated rate of monotone decrease of the *cost-function values* sequence

$$f_k \doteq c^\top x_k,$$

can be deduced from the following result by Radon [31] on the measures of symmetry of convex bodies.

PROPOSITION 2.2 (Radon). *Let $\mathcal{X} \in \mathbb{R}^n$ be a convex body and $x_{\mathrm{G}} = \mathsf{cg}(\mathcal{X})$ be its center of gravity. Denote by $\mathcal{H}$ an arbitrary $(n-1)$-dimensional hyperplane through x_{G}, and let $\mathcal{H}_1$ and $\mathcal{H}_2$ be the two hyperplanes supporting $\mathcal{X}$ and parallel to $\mathcal{H}$. Denote by*

$$r(\mathcal{H}) \;\doteq\; \frac{\min\{\mathrm{dist}(\mathcal{H}, \mathcal{H}_1); \mathrm{dist}(\mathcal{H}, \mathcal{H}_2)\}}{\max\{\mathrm{dist}(\mathcal{H}, \mathcal{H}_1); \mathrm{dist}(\mathcal{H}, \mathcal{H}_2)\}}$$

the ratio of the distances from $\mathcal{H}$ to $\mathcal{H}_1$ and $\mathcal{H}_2$, respectively. Then

$$\min_{\mathcal{H}} r(\mathcal{H}) \geq 1/n.$$

This result immediately applies to the DCG algorithm. Indeed, with $\mathcal{H}_1$ and $\mathcal{H}$ being the two hyperplanes through the two successive points x_k and x_{k+1} as described above, and $\mathcal{H}_2$ being the supporting hyperplane through the optimal point $x^* \doteq \arg\min c^\top x$, we arrive at the following estimate:

$$f_{k+1} - f^* \;\leq\; \frac{n}{n+1}(f_k - f^*), \tag{2.3}$$

where f^* is the optimal value of the objective function. In particular, the following lemma is an immediate consequence of Proposition 2.2.

LEMMA 2.3 (Convergence of DCG). *The DCG algorithm computes an α-optimal solution (i.e. such that $f_k - f^* \leq \alpha$) in at most*

$$k = \left\lceil \frac{\ln \frac{D}{\alpha}}{\ln \frac{n+1}{n}} \right\rceil = \mathcal{O}\left(n \ln D/\alpha\right)$$

steps, where $D \doteq |f_0 - f^*|$.

Proof. The proof immediately follows noting that, from (2.3), we have

$$f_k - f^* \leq \left(\frac{n}{n+1}\right)^k (f_0 - f^*) \leq \left(\frac{n}{n+1}\right)^k D,$$

and imposing $f_k - f^* \leq \alpha$. ∎

In other words, the method is expected to have a guaranteed geometric rate of convergence. Notably, to the best of our knowledge, this result has never been used in optimization, in contrast to similar results on the guaranteed volumetric reduction (in the spirit of Proposition 2.1), which are typical to various modifications of the ellipsoid method.

However, as already mentioned in the Introduction, this nice convergence property has a crucial drawback: The practical implementation of the DCG algorithm is hindered by the fact that computing the center of gravity turns out to be an NP-hard problem, even for the simple case when $\mathcal{X}$ is a polytope, as recently proved by [29]. This motivates the introduction of randomized version of the method, as detailed in the next section.

3. A Randomized Cutting Plane Algorithm. We propose the following randomized modification of the DCG scheme.

Algorithm 2 (RCP algorithm)

Input: $\mathcal{X}$
Output: x^*

1: $k \Leftarrow 0$, $\mathcal{X}_k \Leftarrow \mathcal{X}$;
2: generate N_k uniform random samples in $\mathcal{X}_k$, $\{x^{(1)}, \ldots, x^{(N_k)}\}$;
3: $x_k \Leftarrow \arg\min_{x \in \{x^{(1)}, \ldots, x^{(N_k)}\}} c^\top x$;
4: $\mathcal{X}_{k+1} \Leftarrow \left\{x \in \mathcal{X}_k \,:\, c^\top (x - x_k) \leq 0\right\}$;
5: check Stopping Rule; goto 2.

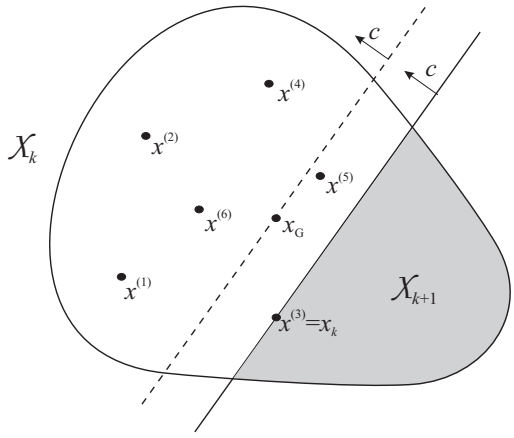

FIG. 3.1. *A sketch of the proposed randomized cutting scheme. The cut at step k is performed at the random point with the smallest cost value (in this case $x_k = x^{(3)}$). The gray area indicates the set $\mathcal{X}_{k+1}$. The dashed line is the hyperplane passing through x_G, the center of gravity of $\mathcal{X}_k$.*

The randomized scheme we propose for computing the query point at step k is very simple and intuitive. At step k, we generate N_k samples in $\mathcal{X}_k$; then, the next query point x_k is selected as the random sample providing the minimum to the objective function. The intuition behind Algorithm 2 is illustrated in Figure 3.1. Clearly, the k-the step of the RCP algorithm will perform better than a corresponding step of a deterministic DCG method as long as at least one random point will fall *below* (in terms of the cost function) the center of gravity x_G of the set $\mathcal{X}_k$. Notice

that the number of sample points drawn at each iteration is allowed to depend on the iteration index k; this is a key feature that we exploit in Section 4.2.2.

In the second part of this paper, we discuss in detail how the proposed scheme can be easily implemented using techniques based on random walk and present specific applications of this methodology. In the next section we instead concentrate on deriving the theoretical properties of the RCP.

It should be noted that a method of a very similar flavor was proposed in [28], with the cuts being performed through the estimated centers of gravity of the $\mathcal{X}_k$ sets; however, no formal theoretical analysis of the overall scheme has been provided in [28].

4. Probabilistic Analysis of RCP. In order to obtain rigorous convergence results, in this section we make the abstract assumption on the availability of a mechanism for generating uniform samples in a convex body. Namely, we assume that the following oracle is available:

ASSUMPTION 1 (Uniform Generating Oracle). *We assume that a uniform generating oracle* UGO *is available such that for any convex set $\mathcal{X} \in \mathbb{R}^n$, a call to* $\mathsf{UGO}(\mathcal{X}, N)$ *returns N random i.i.d. points uniformly distributed in $\mathcal{X}$.*

We stress out that Assumption 1 is made only for theoretical analysis purposes and is quite common in the computational geometry community, see e.g. [30].

First, in the next subsection, we are aimed at analyzing the expected behavior of the RCP scheme.

4.1. Expected convergence rate of RCP. In this section, we explicitly evaluate the expected convergence rate of the RCP algorithm and formally show that it improves by a constant factor upon that of DCG. To this end, we consider $N_k \equiv N$ and let

$$\{x^{(1)}, \dots, x^{(N)}\} = \mathsf{UGO}(\mathcal{X}_k, N)$$

be N uniform random points in $\mathcal{X}$, and define the following random variables:

$$f^{(i)} \doteq c^\top x^{(i)}, \quad i = 1, \dots, N,$$

and

$$f_{[1]} \doteq \min_{i=1,\dots,N} f^{(i)}. \tag{4.1}$$

The random variable $f_{[1]}$ represents the so-called *first order statistics* of $f^{(i)}$, e.g., see [10]. The key theorem below proves that, for every convex body $\mathcal{X}$, the expected value of the relative distance between $f_{[1]}$ and the optimum f^* is bounded from below and from above by constants that depend only on n and N.

THEOREM 4.1. *Let $\mathcal{X} \subseteq \mathbb{R}^n$ be a convex body. Given $c \in \mathbb{R}^n$, define $h \doteq (\max_{\mathcal{X}} c^\top x - \min_{\mathcal{X}} c^\top x)$ and $f^* \doteq \min_{\mathcal{X}} c^\top x$. Then, it holds that*

$$\frac{1}{nN+1} \leq \frac{\mathbb{E}\left[f_{[1]} - f^*\right]}{h} \leq \frac{1}{n} B(N+1, 1/n) \tag{4.2}$$

$$\leq \left(\frac{1}{N+1}\right)^{\frac{1}{n}} \tag{4.3}$$

where $B(\cdot, \cdot)$ is the Euler Beta function.

Proof. Assume, without loss of generality, that $c = [1\, 0 \cdots 0]^\top$ (that is, $c^\top x = x_1$) and that $x^* = \arg\min_{\mathcal{X}} c^\top x = 0$. We begin by proving the upper bound in (4.2). The first step in the proof of Theorem 4.1 is closely related to the proof of Lemma 4 in [3]. Define the function

$$\phi_{\mathcal{X}}(s) = \frac{1}{\mathrm{vol}\,(\mathcal{X})} \int_{x \in \mathcal{X}, c^\top x = s} \mathrm{d}x$$

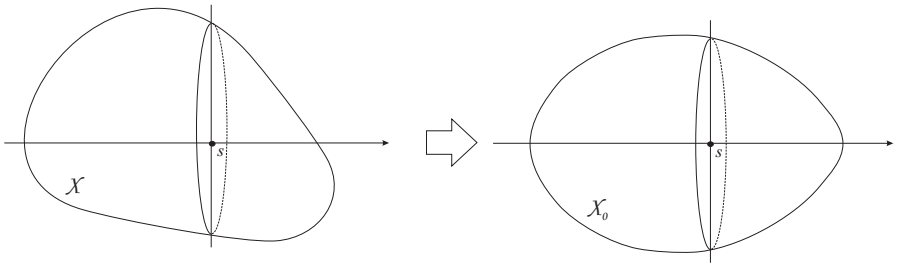

FIG. 4.1. *First step of the proof of Theorem 4.1*

that represents the $(n-1)$-dimensional volume of $\mathcal{X}$ intersected with the hyperplane $c^\top x = s$, as a fraction of the n-dimensional volume of $\mathcal{X}$. Then, define the set $\mathcal{X}_0$ obtained by replacing each cross-section $\mathcal{X} \cap \{x : x_1 = s\}$ by an $(n-1)$-dimensional ball of volume $\phi_{\mathcal{X}}(s)$ and centered at the point $[s\, 0 \cdots 0]^\top$, as shown in Figure 4.1. The set $\mathcal{X}_0$ has the same volume as $\mathcal{X}$, and it holds that $\phi_{\mathcal{X}}(s) = \phi_{\mathcal{X}_0}(s)$. Moreover, if we let S_1 and S_2 be the cross-sections of $\mathcal{X}_0$ at s_1 and s_2, respectively, and let S be its cross-section at a point $s = \lambda s_1 + (1-\lambda)s_2$, $\lambda \in [0,1]$, then, by the Brunn-Minkowski inequality (e.g., see [12]), we have

$$\mathrm{vol}\,(S)^{\frac{1}{n-1}} \;\geq\; \mathrm{vol}\,(\lambda S_1 + (1-\lambda)S_2)^{\frac{1}{n-1}} \;\geq\; \lambda \mathrm{vol}\,(S_1)^{\frac{1}{n-1}} + (1-\lambda)\mathrm{vol}\,(S_2)^{\frac{1}{n-1}}. \tag{4.4}$$

Hence, if we denote by $r(s)$ the radius of the $(n-1)$-dimensional ball at s, then (4.4) implies that $r(\lambda s_1 + (1-\lambda)s_2) \geq \lambda r(s_1) + (1-\lambda) r(s_2)$. This, in turn, implies that $r(s)$ is a concave function, thus $\mathcal{X}_0$ is a convex set.

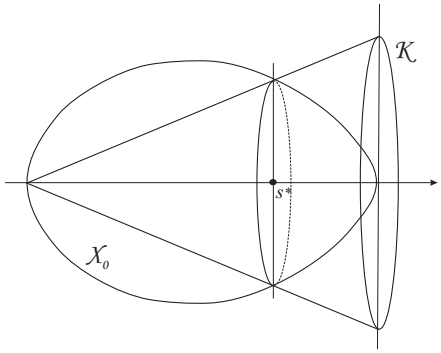

FIG. 4.2. *Second step of the proof of Theorem 4.1*

As a second step, define now the cone $\mathcal{K}$ with base area $S = \frac{n}{h}\mathrm{vol}\,(\mathcal{X})$, the axis directed along c and located as shown in Figure 4.2, with h being the height of $\mathcal{K}$. Then, by construction, we have $\mathrm{vol}\,(\mathcal{K}) = \mathrm{vol}\,(\mathcal{X}) = \mathrm{vol}\,(\mathcal{X}_0)$. Let s^* be the coordinate at which the sets $\mathcal{X}_0$ and $\mathcal{K}$ intersect, see Figure 4.2. Next, for every $s \in [0,h]$, define the sets $\mathcal{X}^+(s) \doteq \{x \in \mathcal{X} : x_1 \geq s\}$, $\mathcal{X}_0^+(s) \doteq \{x \in \mathcal{X}_0 : x_1 \geq s\}$ and $\mathcal{K}^+(s) \doteq \{x \in \mathcal{K} : x_1 \geq s\}$. Then, the following chain of inequalities holds

$$\begin{aligned} \mathbb{P}\{f^{(i)} \geq s\} &= \frac{\mathrm{vol}\,(\mathcal{X}^+(s))}{\mathrm{vol}\,(\mathcal{X})} = \frac{\mathrm{vol}\,\big(\mathcal{X}_0^+(s)\big)}{\mathrm{vol}\,(\mathcal{X}_0)} \\ &\leq \frac{\mathrm{vol}\,(\mathcal{K}^+(s))}{\mathrm{vol}\,(\mathcal{K})} = \frac{h^n - s^n}{h^n} \end{aligned} \tag{4.5}$$

where the last inequality follows from the fact that, for $s \geq s^*$,

$$\frac{\mathrm{vol}\left(\mathcal{X}_0^+(s)\right)}{\mathrm{vol}\left(\mathcal{X}_0\right)} \leq \frac{\mathrm{vol}\left(\mathcal{K}^+(s)\right)}{\mathrm{vol}\left(\mathcal{K}\right)},$$

and, for $s < s^*$,

$$\frac{\mathrm{vol}\left(\mathcal{X}_0^+(s)\right)}{\mathrm{vol}\left(\mathcal{X}_0\right)} = 1 - \frac{\mathrm{vol}\left(\mathcal{X}_0^-(s)\right)}{\mathrm{vol}\left(\mathcal{X}_0\right)} \leq 1 - \frac{\mathrm{vol}\left(\mathcal{K}^-(s)\right)}{\mathrm{vol}\left(\mathcal{X}_0\right)} = \frac{\mathrm{vol}\left(\mathcal{K}^+(s)\right)}{\mathrm{vol}\left(\mathcal{K}\right)},$$

where $\mathcal{X}_0^-(s) \doteq \{x \in \mathcal{X}_0 : x_1 < s\}$ and $\mathcal{K}^-(s) \doteq \{x \in \mathcal{K} : x_1 < s\}$.

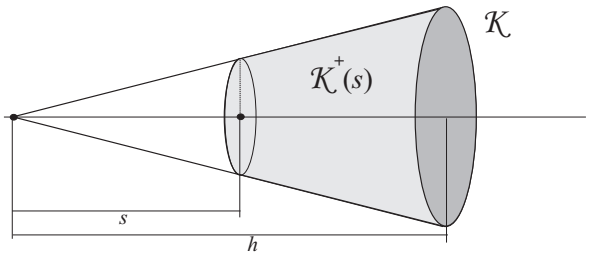

FIG. 4.3. *Third step of the proof of Theorem 4.1: the sets $\mathcal{K}$ and $\mathcal{K}^+(s)$*

As a final step for proving the upper bound in (4.3), notice that $f_{[1]}$ is a positive random variable, and hence, we may write

$$\begin{aligned}
\mathbb{E}\left[f_{[1]}\right] &= \int_0^h \mathbb{P}\{f_{[1]} \geq s\}\mathrm{d}s = \int_0^h \left(\mathbb{P}\{f^{(i)} \geq s\}\right)^N \mathrm{d}s \\
\text{[from (4.5)]} &\leq \int_0^h \left(\frac{h^n - s^n}{h^n}\right)^N \mathrm{d}s \\
[t = s^n/h^n] &= \frac{h}{n}\int_0^1 (1-t)^N t^{\frac{1}{n}-1}\mathrm{d}t = \frac{h}{n}B(N+1, 1/n).
\end{aligned}$$

Finally, applying Theorem 3.4. in [1], we get the following bounds

$$1 - \left(\frac{N}{N+1}\right)^{\frac{1}{n}} \leq \frac{1}{n}B(N+1,1/n) \leq \left(\frac{1}{N+1}\right)^{\frac{1}{n}},$$

thus proving the inequality in (4.3), with the cone $\mathcal{K}$ being the attainable "worst-case" configuration for $\mathcal{X}$.

The lower bound in (4.2) can be proved similarly. Namely, instead of the $\mathcal{K}$ above, consider the "inverted" cone; then with the reasonings identical to those above, it proves to be the "best case" configuration. We hence derive the following inequality

$$\mathbb{P}\{f^{(i)} \geq s\} \geq \frac{s^n}{h^n}. \tag{4.6}$$

Therefore, we obtain

$$\begin{aligned}
\mathbb{E}\left[f_{[1]}\right] &= \int_0^h \mathbb{P}\{f_{[1]} \geq s\}\mathrm{d}s = \int_0^h \left(\mathbb{P}\{f^{(i)} \geq s\}\right)^N \mathrm{d}s \\
\text{[from (4.6)]} &\geq \int_0^h \left(\frac{s^n}{h^n}\right)^N \mathrm{d}s = \frac{h}{nN+1},
\end{aligned} \tag{4.7}$$

which concludes our proof. □

From Figure 4.4, it is seen that the upper and lower bounds derived in Theorem 4.1 are indeed tight: the upper bound is attained when the set $\mathcal{X}$ has the ("worst-case") cone configuration ($\triangleleft$), while the lower bound corresponds to the ("best-case") inverted cone configuration ($\triangleright$). In the figure, we also plot the empirical mean values of $f_{[1]}$ (averaged over 10,000 realizations) computed for these two cones as well as for two intermediate configurations, namely a box and a ball (ℓ_∞- and ℓ_2-norm balls).

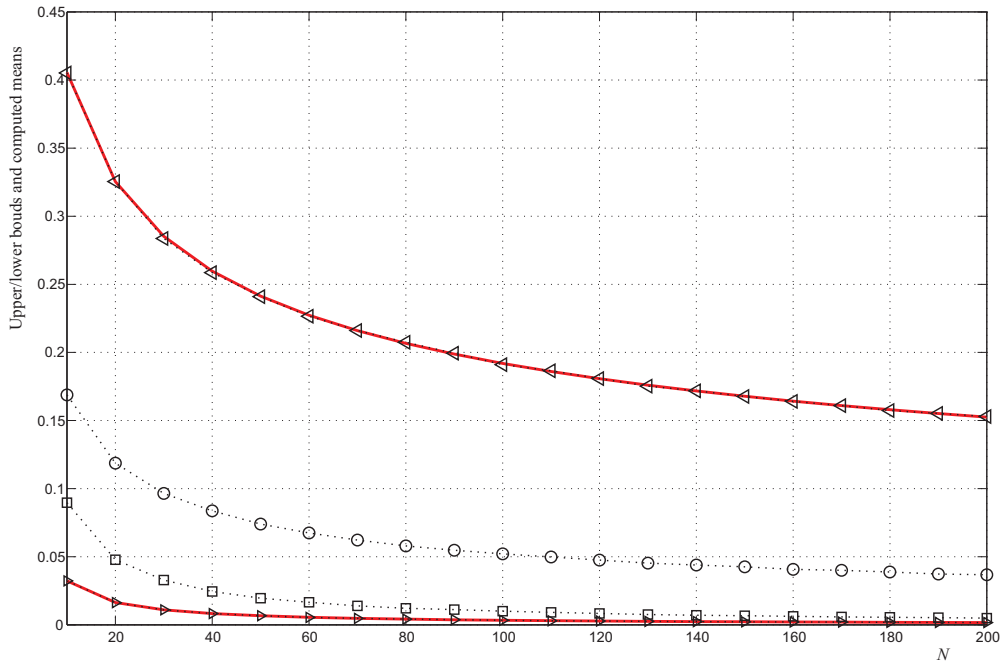

FIG. 4.4. *Illustration of the derived bounds on the mean for $n = 3$: cone configuration, upper bound ($\triangleleft$); inverted cone configuration, lower bound ($\triangleright$); cube configuration ($\square$); ball configuration ($\circ$).*

REMARK 1 (Connections with Radon's result). *Theorem 4.1 constitutes an important and non-trivial extension of Proposition 2.2. Indeed, for $N = 1$, we have $\mathbb{E}\left[f_{[1]}\right] = \mathbb{E}\left[f^{(i)}\right] = \mathsf{cg}(\mathcal{X})$, and*

$$\frac{1}{n}B(2, 1/n) = \frac{n}{n+1}.$$

Analyzing inequality (4.3), we see that for fixed dimension n, as the number of samples N goes to infinity, the expected value of $f_{[1]}$ does converge to f^*, as one may expect. However, we also observe that for fixed N, the distance $(f_{[1]} - f^*)$ tends to h as the dimension n grows. Notice that this fact implies that a simple implementation of a randomized algorithm, where we draw N samples in $\mathcal{X}$ and adopt the best one as a candidate optimizer, is bound to fail as the dimension grows. Indeed, if we were to impose a relative precision

$$\mathbb{E}\left[f_{[1]} - f^*\right]/h \leq \alpha, \quad \alpha < 1,$$

the number of uniform samples to be drawn in $\mathcal{X}$ would grow exponentially as

$$N = \left\lceil \frac{1}{\alpha^n} \right\rceil. \tag{4.8}$$

This is not surprising, since it is well known that in general, an exponential number of samples is needed to approximate with a given precision the minimum of a function, see [2].[1] This latter reasoning further motivates the randomized cutting plane approach we are proposing. In particular, the following corollary proves that the RCP scheme possesses expected geometric convergence.

COROLLARY 4.2 (Expected convergence of RCP). *The expected number of steps required for the RCP algorithm with $N_k \equiv N$ to compute an α–optimal solution (i.e., such that $f_k - f^* \leq \alpha$) is at most*

$$k = \left\lceil \frac{1}{\ln(N+1)}\, n \ln R/\alpha \right\rceil.$$

Proof. A direct application of Theorem 4.1 shows that at step k we can write

$$\mathbb{E}[f_k - f^*] \leq \left(\frac{1}{N+1}\right)^{\frac{1}{n}} (\mathbb{E}[f_{k-1} - f^*]) \leq \left(\frac{1}{N+1}\right)^{\frac{k}{n}} (f_0 - f^*) \leq \left(\frac{1}{N+1}\right)^{\frac{k}{n}} R. \tag{4.9}$$

Thus, the corollary is immediately proved imposing $\mathbb{E}[f_k - f^*] \leq \alpha$. □

This corollary constitutes one of the main contributions of the present work. It is interesting to remark that the result is perfectly consistent with intuition; in particular, it is easily verified that the derived convergence rate reduces to the one in Lemma 2.3 when $N = 1$, while it improves by a factor of $\ln(N+1)$ when $N > 1$.

We complete this section on the expected rate of convergence by extending the results of Theorem 4.1 to the analysis of the variance of the proposed method.

THEOREM 4.3. *Let $\mathcal{X} \subseteq \mathbb{R}^n$ be a convex body. Define h, f^*, and $f_{[1]}$ as in Theorem 4.1. Moreover, without loss of generality assume that $f^* = 0$. Then, it holds that*

$$\frac{1}{h^2}\left(\mathbb{E}\big[f_{[1]}^2\big]\right) \;\leq\; \frac{2}{n}B(N+1,2/n) \;\leq\; \left(\frac{1}{N+1}\right)^{\frac{2}{n}}; \tag{4.10}$$

$$\frac{1}{h^2}\left(\mathbb{V}ar\left[f_{[1]}\right]\right) \;=\; \frac{1}{h^2}\left(\mathbb{E}\big[f_{[1]}^2\big] - \mathbb{E}^2\big[f_{[1]}\big]\right) \;\leq\; \left(\frac{1}{N+1}\right)^{\frac{2}{n}}. \tag{4.11}$$

Proof. Assume again that $c = [1\,0\,\cdots\,0]^\top$ and $x^* = \arg\min_{\mathcal{X}} c^\top x = 0$. Then, $f_{[1]}$ is a positive random variable; hence, we may write

$$\begin{aligned}
\mathbb{E}\left[f_{[1]}^2\right] &= \int_0^h \mathbb{P}\{f_{[1]}^2 \geq s\}\mathrm{d}s \;=\; \int_0^h \mathbb{P}\{f_{[1]} \geq \sqrt{s}\}\mathrm{d}s \;=\; \int_0^h \left(\mathbb{P}\{f^{(i)} \geq \sqrt{s}\}\right)^N \mathrm{d}s \\
\text{[from (4.5)]} \qquad &\leq \int_0^h \left(\frac{h^n - s^{n/2}}{h^n}\right)^N \mathrm{d}s \\
[t = s^{n/2}/h^n] \qquad &= \frac{2h^2}{n}\int_0^1 (1-t)^N t^{\frac{2}{n}-1}\mathrm{d}t \;=\; \frac{2h^2}{n}B(N+1,2/n),
\end{aligned}$$

thus proving the first inequality in (4.10). The second inequality (4.10) can be proved by applying Theorem 3.4 in [1]. Inequality (4.11) directly follows from (4.10) by simply dropping out the second term. □

The theorem above shows that the standard deviation of $f_{[1]}$ decreases as the number of samples N increases. However, it should be noted that, contrary to the expected value result, this bound is not tight; indeed, numerical experiments testify to a faster convergence with respect to N.

---

[1] Interestingly, if we apply inequality (4.8) with $\alpha h = \mathsf{cg}(\mathcal{X})$, we get $N = \left\lceil \left(\frac{n}{n+1}\right)^{\frac{1}{n}} \right\rceil < \mathrm{e}$.

4.2. Sample size results. In this section, we perform a different probabilistic analysis of the proposed randomized scheme. This allows us to derive explicit bounds on the number of samples to be drawn at each step of the algorithm in order to guarantee with arbitrarily high *probability* the desired convergence behavior.

4.2.1. Single step analysis. We first analyze the theoretical behavior of a *single step* of the RCP algorithm. To this end, consider the situation at a generic step k and define by x_{G} the center of gravity of the set $\mathcal{X}_k$ and by x_k the query point returned by line 3 of Algorithm 2. Formally, define Worse_k as the event of RCP returning at step k a query point which is worse (i.e. *above* in terms of cost-function value) than the center of gravity of $\mathcal{X}_k$. The probability of this latter event can be easily bounded in the following way:

$$\begin{aligned}\mathbb{P}\{\mathsf{Worse}_k\} &= \mathbb{P}\{c^\top x_k \geq c^\top x_{\mathrm{G}}\} \\ &= \mathbb{P}\{c^\top x^{(i)} \geq x_{\mathrm{G}}, i = 1, \ldots, N_k\} \\ &= \mathbb{P}\{c^\top x^{(i)} \geq x_{\mathrm{G}}\}^{N_k} \\ &\leq (1 - 1/e)^{N_k},\end{aligned}$$

where the last inequality follows from Proposition 2.1. Hence, we have the following lemma showing that, by appropriately selecting the number of samples N_k, we can guarantee with a prescribed arbitrarily high probability that the k-th step of RCP algorithm performs better than the corresponding DCG step.

LEMMA 4.4 (Single step analysis of RCP). *Given a probability level $\epsilon > 0$, set*

$$N_k \geq 2.2 \ln \frac{1}{\epsilon}.$$

Then, with probability at least $1 - \epsilon$, it holds

$$c^\top x_k \leq c^\top \mathsf{cg}(\mathcal{X}_k). \tag{4.12}$$

Proof. The lemma is easily proved by noticing that

$$N_k \geq \frac{\ln \frac{1}{\epsilon}}{\ln \frac{1}{1-1/\mathrm{e}}}$$

implies $(1 - 1/e)^{N_k} \leq \epsilon$ and $(\ln \frac{1}{1-1/\mathrm{e}})^{-1} \approx 2.1802 < 2.2$. ∎

The previous lemma shows that one can always guarantee *a priori* that a single step of the RCP algorithm will perform at least as good as the corresponding DCG step. In the next section, we concentrate on the overall behavior of the algorithm and derive a specific bound, based on a Bonferroni-like inequality, on the number of samples required to guarantee (with high probability) that the RCP algorithm will expose the same geometric rate of convergence as that of DCG.

4.2.2. Overall analysis of RCP. In the theorem below, the *probabilistic* behavior of the RCP algorithm is characterized. In particular, we bound from above the probability that the algorithm performs worse than DCG.

THEOREM 4.5 (Behavior of RCP). *Given a probability level $\epsilon > 0$, choose*

$$N_k \geq N_{\mathrm{guar}}(\epsilon, k) \doteq 2.2 \ln \frac{1}{\epsilon} + (1.1 + 0.505 \ln k). \tag{4.13}$$

Then, with probability at least $(1 - \epsilon)$, the RCP algorithm will expose a better or equal rate of convergence than DCG.

Proof. The probability of the algorithm performing worse than DCG is the probability of the event

$$\mathsf{Worse}_\infty \doteq \{\text{at some step } k \text{ it holds } \; c^\top x_k \geq c^\top \mathsf{cg}(\mathcal{X}_k)\}.$$

This probability can be bounded as

$$\begin{aligned} \mathbb{P}\{\mathsf{Worse}_\infty\} &\leq \mathbb{P}\{\mathsf{Worse}_1 \cup \mathsf{Worse}_2 \cup \mathsf{Worse}_3 \cdots\} \\ &= \sum_{k=1}^{\infty} \mathbb{P}\{\mathsf{Worse}_k\} \\ &\leq \sum_{k=1}^{\infty} (1-1/e)^{N_k}. \end{aligned}$$

To make this sum finite and equal to the desired probability level ϵ, it is sufficient to select the sequence N_k such that

$$(1-1/e)^{N_k} = \frac{k^{-\alpha}}{\zeta(\alpha)}\,\epsilon, \quad \alpha > 1, \tag{4.14}$$

where $\zeta(\alpha)$ is the Riemann zeta function, for which $\sum_{k=1}^{\infty} k^{-\alpha} = \zeta(\alpha)$. In fact, in this case we have

$$\sum_{k=1}^{\infty} (1-1/e)^{N_k} = \sum_{k=1}^{\infty} \frac{k^{-\alpha}}{\zeta(\alpha)}\,\epsilon \;=\; \epsilon.$$

Solving equation (4.14) with respect to N_k, we obtain the bound

$$N_k \;\geq\; \frac{\ln \zeta(\alpha) + \alpha \ln k + \ln \frac{1}{\epsilon}}{\ln \frac{1}{1-1/\mathrm{e}}}\,.$$

Choosing, by trial and error, $\alpha = 1.1$, we get the bound (4.13). ∎

Table 4.1 shows different values of bound (4.13) for given values of ϵ and k. Note that the number of required random samples is indeed very low, even for very small values of ϵ.

$\lceil N_{\mathrm{guar}}(\epsilon, k) \rceil$	$\epsilon = 0.01$	$\epsilon = 0.005$	$\epsilon = 0.001$	$\epsilon = 0.0005$	$\epsilon = 0.00001$
$k = 1$	12	13	17	18	27
$k = 10$	13	14	18	19	28
$k = 100$	14	16	19	21	29
$k = 1,000$	15	17	20	22	30
$k = 10,000$	16	18	21	23	32

TABLE 4.1
Behavior of the bound $N_{\mathrm{guar}}(\epsilon, k)$ for different values of ϵ and k.

5. A Hit-and-Run Implementation of RCP. In the previous sections, the probabilistic properties of the proposed algorithm have been analyzed in a rather abstract setting which assumed the existence of a mechanism for generating uniform random samples from the current set $\mathcal{X}_k$. We now briefly discuss how this assumption can be relaxed using random walks. To this end, we replace Assumption 1 on UGO with the more realistic one on the availability of a *boundary oracle* as follows.

ASSUMPTION 2 (Boundary Oracle). *We assume that a boundary oracle* (BO) *is available such that, given a direction $v \in \mathbb{R}^n$ and a point z belonging to a bounded convex set $\mathcal{X} \in \mathbb{R}^n$, a call to* $\mathsf{BO}(x, \mathcal{X}, v)$ *returns the two points $\overline{z}$, $\underline{z}$ which are the intersections of the 1D line $x + \lambda v$, $\lambda \in \mathbb{R}$, with the boundary of $\mathcal{X}$.*

In fact, the boundary oracle assumption can be relaxed to the assumption of having a *membership oracle*. Indeed, from the latter one, a boundary oracle can be always constructed by means of a binary search (e.g., see [21]). However, for a wide range of problems, a boundary oracle can be easily formulated in closed form, and in Section 6 we present a particular BO for sets specified by LMIs.

The availability of a boundary oracle is crucial for obtaining implementable versions of the RCP algorithm by means of techniques based on *random walks* in convex bodies (e.g., see [38] for a very detailed survey on these methodologies).

In general, the problem of sampling from a convex body $\mathcal{X}$ has received increasing attention in the last decade. The main reason is that it provides an efficient (polynomial-time) way for estimating the volume of $\mathcal{X}$, see for instance [23, 16, 22]. Specifically, a random walk in $\mathcal{X}$ starts at some point in $\mathcal{X}$ and moves to a neighboring point chosen according to a specific randomized rule that depends on the current point only. Many different techniques have been proposed in the literature to choose the next point to walk in; for instance, the grid or lattice walk [11], the ball walk [16] and the Hit-and-Run algorithm.

We adopt the latter methodology as the most promising one, since it has been proven to possess the best known bounds on the number of steps needed to obtain a random sample. The H&R algorithm has been proposed by Turchin [37] and independently later by Smith [34] and is aimed at generating points with approximately uniform distribution in a body. Its properties have been studied in numerous works by Lovász and co-authors (e.g., see the survey paper [38]). This algorithm in its simplest form is recalled next:

Algorithm 3 (H&R algorithm)

Input: $x_0 \in \mathcal{X}$, N
Output: $x^{(N)}$ random point belonging to $\mathcal{X}$

1: $z \Leftarrow x_0$;
2: **for** $i = 1$ to N **do**
3: generate a uniform random direction $v \in \mathbb{R}^n$,
4: $\{\overline{z}, \underline{z}\} = \mathsf{BO}(\mathcal{X}, z, v)$,
5: generate a uniform point z in the segment $[\overline{z}, \underline{z}]$,
6: **end for**;
7: $x^{(N)} \Leftarrow z$.

The random direction v in line 3 can be easily generated as $\xi/\|\xi\|_2$, where $\xi \in \mathbb{R}^n$ is a Gaussian random variable with zero mean and identity covariance matrix.

Among the attractive features of H&R is the so-called polynomial-time mixing property, which was proved by Smith in [34] and later refined by various authors (e.g., see [23, 22]). Roughly speaking, it formulates as follows: Under certain conditions on the geometry of $\mathcal{X}$ and the initial H&R-point, in order to obtain a point such that the total variation[2] between its distribution and the uniform distribution on $\mathcal{X}$ is bounded by a given constant, the number of steps required (mixing time) is *polynomial in the dimension* n.

Using these results, the paper [9] presents theoretical analysis showing how the RCP sample size bound derived in Section 4 can be immediately extended to the H&R setup. We prefer not to dwell further on this topic, which has been studied in the recent literature. The interested reader is referred to the paper [3], which provides a general exposition on the construction of cutting plane schemes based on H&R and discusses at a theoretical level several implementation issues.

[2]For two probability distributions $\mathbb{P}_1$, $\mathbb{P}_2$ on the same underlying σ-algebra on $\mathcal{X}$, their *total variation* distance is defined as

$$\|\mathbb{P}_1 - \mathbb{P}_2\|_{\mathsf{tv}} \doteq \sup_{A \subseteq \mathcal{X}} |\mathbb{P}_1(A) - \mathbb{P}_2(A)|.$$

Results along this line are of great theoretical importance, showing that H&R schemes can be implemented in polynomial time, but they are still unsatisfactory from a practical viewpoint.

The main reason is that the constants involved in the sample size bounds available so far are usually very large, thus making the method not good in practice. On the other hand, numerical experience shows that H&R usually performs much better than predicted theoretically. This fact is acknowledged by most authors and practitioners. For instance, Bertsimas and Vempala in [3] state that *"In practice, drawing random samples from convex sets might be much faster than the known worst-case bounds; also, sampling convex sets is an active research area and there might well be faster sampling methods in the future that would directly improve the complexity of the random walk algorithm."*

This is also the main reason that convinced us to keep separate the analysis of the pure theoretical setup of Section 4. Equally importantly, in the RCP setup, not only the theoretical bounds on the number of samples are pessimistic in practice, but the above-mentioned conditions on the geometry of the feasible set are also very hard to guarantee in the process of cutting. As a result, implementable H&R-based versions of the algorithm do differ from Algorithm 2 as better described in Section 6.1.

6. Application to Linear Matrix Inequalities. In this section, we focus our attention at standard semidefinite programs (SDP) of the form

$$\min c^\top x \qquad \text{subject to} \qquad F(x) \doteq F_0 + \sum_{i=1}^{n} x_i F_i \preceq 0, \tag{6.1}$$

where $c \in \mathbb{R}^n$ and $F_i \in \mathbb{R}^{m\times m}$, $i = 0, \dots, n$, are known symmetric matrices; the notation $F \preceq 0$ stands for negative semidefiniteness of the matrix F. The constraint inequality in (6.1) is called a linear matrix inequality (LMI), and the convex set

$$\mathcal{X}_{\rm LMI} \doteq \{x \in \mathbb{R}^n : F(x) \preceq 0\}$$

is referred to as the feasible domain of this LMI. To exclude trivialities, we assume that the set $\mathcal{X}_{\rm LMI}$ is nonempty and bounded (this is equivalent to assuming that $\sum_{i=1}^n x_i F_i$ is sign indefinite for all x).

The optimization problem (6.1) is known to be one of the key problems in modern convex optimization [4]. It has numerous applications in various fields of system theory, control and optimization, and at present there exist efficient solution techniques based on interior-point methods; e.g., see [25]. It is because of this generality and availability of "alternative" solvers that we exemplify here the use of our method for this widely adopted convex optimization setup and compare the numerical results with those obtained using one of the "standard" MATLAB-based toolboxes, SeDuMi [35] together with the most popular interfaces YALMIP [20] and `cvx` [14].

To apply our method, we need a semidefinite boundary oracle (SDBO) for the set $\mathcal{X}_{\rm LMI}$, i.e., the intersections of a 1D-line and the boundary of $\mathcal{X}_{\rm LMI}$ are to be computed efficiently. This is accomplished via the following lemma developed in [27], see also the work [5] for a similar result.

LEMMA 6.1 (SDBO). *Let $A \prec 0$ and $B = B^\top$. Then, the minimal and the maximal values of the parameter $\lambda \in \mathbb{R}$ retaining the negative definiteness of the matrix $A + \lambda B$ are given by*

$$\underline{\lambda} = \begin{cases} \max\limits_{\lambda_i<0} \lambda_i, & \\ & \\ -\infty, & \text{if all } \lambda_i > 0; \end{cases}$$

and

$$\overline{\lambda} = \begin{cases} \min\limits_{\lambda_i>0} \lambda_i, & \\ & \\ +\infty, & \text{if all } \lambda_i < 0; \end{cases}$$

where λ_i are the generalized eigenvalues of the pair of matrices $(A, -B)$, i.e., $Ae_i = -\lambda_i B e_i$.

By means of this lemma, the desired endpoints of the segment $[\underline{z}, \overline{z}]$ are computed as

$$\underline{z} = z + \underline{\lambda} v \qquad \text{and} \qquad \overline{z} = z + \overline{\lambda} v.$$

In the setup of this paper, assume that $z \in \mathcal{X}_{\text{LMI}}$, and $v \in \mathbb{R}^n$ is a (random) direction. We then have

$$F(z + \lambda v) = F(z) + \lambda\big(F(v) - F_0\big) \doteq A + \lambda B,$$

and using Lemma 6.1, the desired intersection points of the line and the boundary of $\mathcal{X}_{\text{LMI}}$ are given by $\underline{z} = z + \underline{\lambda} v$ and $\overline{z} = z + \overline{\lambda} v$. To compute the intersection points with the boundary of a current set $\mathcal{X}_k$, the additional linear condition $c^\top(x - x_{k-1}) \le 0$ defining the feasible set at step k is to be taken into account, which is straightforward to implement.

As seen from the aforesaid, this operation should be frequently performed in the process of iterations. The basis of this operation is finding the eigenvalues of symmetric matrices (which is for instance efficiently implemented in MATLAB). Hence, the boundary oracle for LMIs is accurate and "cheap" enough for matrices of quite large dimensions, as confirmed by the numerical simulations discussed in the next section.

6.1. Numerical Experiments. Prior to reporting on the results of experiments, we discuss some of the most important implementation issues, including modifications of the basic scheme that showed to be crucial for the method to perform satisfactorily in practice. We remark again that, in the spirit of the discussion at the end of Section 5, the implemented procedure differs from Algorithm 2; it involves certain semi-heuristic tricks and would require a careful theoretical analysis that we aim to perform in the subsequent papers.

a) Initialization. To implement the H&R procedure, we need to find an initial feasible point $x_0 \in \mathcal{X}_{\text{LMI}}$. This can be done by solving the auxiliary problem

$$\min \gamma \quad \text{subject to} \quad F(x) \preceq \gamma I$$

that admits the couple $\{x = 0, \gamma = \max \text{eig}\big(F(0)\big)\}$ as initial feasible solution. If the optimal solution $\{x^*, \gamma^*\}$ of this problem is such that $\gamma^* > 0$, then the original problem (6.1) is infeasible; otherwise we take $x_0 = x^*$ as initial feasible point. Notice also that, for the specific SDP problem, we can also easily force the initial feasible point to be at a distance d from the boundary of $\mathcal{X}$. The procedure above allows to find a good initial point to pass to the algorithm. However, it should be noted that, after the first step of this algorithm, by construction, the H&R starting point x_k lies on the boundary of the new set $\mathcal{X}_{k+1}$, and therefore we would have $d = 0$. There are different ways to avoid this degeneracy. For instance, the initialization procedure described in Section 4.1.1 of [6] can be directly applied to our setup. In our case, we adopted a method based on boundary points, as better described in point d) below.

b) Number of points. As it follows from the discussion in Section 5, the number N_k of random points suggested by Theorem 4.5 has to be enlarged because of the mixing time required for H&R algorithm. Also, extra points are required to bring the set in near-isotropic position, as described below. For these reasons, in the present implementation, we decided to lean on $N_k \equiv N \gg N_{\text{guar}}(0.01, 10000) = 16$ points at every iteration.

c) Isotropization. The theoretical bounds on the required number of H&R points depend explicitly on the radii of two balls respectively inscribed and inscribing the set $\mathcal{X}$. While these quantities can be assumed to be known at the first steps of the RCP algorithm, this does not hold true anymore as the algorithm proceeds. Indeed, at step $(k-1)$, the set $\mathcal{X}_k$ is obtained from the original set $\mathcal{X}$ by cutting a portion defined by the hyperplane $\mathcal{H}_{k-1} = \big\{x \in \mathbb{R}^n : c^\top(x - x_{k-1}) = 0\big\}$, and there is no guarantee that it still contains a ball of a given radius. A well-accepted technique (see

for instance [3]) to overcome this problem, is to perform an affine transformation of $\mathcal{X}_k$ to bring it in near-isotropic position[3].

The procedure for bringing a set in near-isotropic position is quite straightforward: for a general convex set $\mathcal{X}$, let $y^{(1)}, \ldots, y^{(N)}$ be random samples from $\mathcal{X}$. Compute the quantities

$$\bar{y} \doteq \frac{1}{N}\sum_{i=1}^{N} y^{(i)} \qquad \text{and} \qquad Y \doteq \frac{1}{N}\sum_{i=1}^{N}(y^{(i)}-\bar{y})(y^{(i)}-\bar{y})^\top \tag{6.2}$$

and apply the affine transformation defined by $\bar{y}$ and Y to the set $\mathcal{X}$, obtaining the new set

$$\widetilde{\mathcal{X}} \doteq \left\{ x \in \mathbb{R}^n \,:\, Y^{\frac{1}{2}}x + \bar{y} \in \mathcal{X} \right\}. \tag{6.3}$$

It was shown by Rudelson [32] that one can bring a set in near-isotropic position with high probability in $\mathcal{O}(n \log^2 n)$. In this paper, we adopted the implicit isotropization technique proposed by Kannan *et al.* in [16]. In that case, the set $\mathcal{X}$ is left unchanged and the direction vector for step 2 of Algorithm 3 is taken in the form $v = Y^{1/2}\eta$, where Y is computed as (6.2) and the vector $\eta \in \mathbb{R}^n$ is uniformly distributed on the surface of the unit hypersphere.

d) Heuristics. Various heuristic schemes proved effective in our simulations.

First, as regards the isotropization step, we notice that, as a by-product of H&R algorithm at the kth iteration, we have $2N$ points on the boundary of the set $\mathcal{X}_k$ (endpoints $\underline{z}$, $\overline{z}$). The transformation matrix Y in (6.2) is composed from these boundary points. Based on the assumption that the subsequent sets $\mathcal{X}_k$ and $\mathcal{X}_{k+1}$ have "similar" geometry, we perform isotropization of $\mathcal{X}_{k+1}$ by means of this matrix Y obtained at the previous step.

Moreover, those of the boundary points obtained at the kth step that fall into $\mathcal{X}_{k+1}$ were used to perform the initialization phase of point a). More precisely, the initial point for the H&R algorithm at the $(k+1)$st step was taken as the arithmetic mean of these selected points. If the cut is performed through the "best" H&R-point as described above, then there exist $N_b \geq 1$ such points (at least the endpoint of the chord associated with the "best" H&R-point). For the sake of numerical safety, we performed the cut through the "second best" point to make sure $N_b \geq 3$ so that averaging yields an interior initial point.

Finally, a negative definite matrix A was considered singular if its largest eigenvalue was greater than -10^{-10}, i.e., $\max \mathrm{eig}(F(x_k)) > -10^{-10}$ was adopted as stopping rule.

We do not discuss here the choice of other numerical constants involved in computations and various extra modifications of the basic scheme (e.g., regularization constants, alternative ways for computing the transformation matrix Y, etc); this would be a subject of a separate paper primarily focused on implementation issues.

e) Numerical Experiments. The method was tested over a range of problems whose data were generated randomly as described next. Symmetric matrices F_i were generated so as to guarantee nonemptiness of $\mathcal{X}_{\mathrm{LMI}}$; for simplicity, we chose $F_0 \prec 0$ in the form

$$M = 2\,\mathtt{rand}(m) - 1; \quad F_0 = -M \cdot M^\top - \mathtt{eye}(m),$$

so that $x_0 = 0$ was adopted as the initial point for iterations. The rest of the matrices F_i, $i > 0$, were computed as

$$\begin{array}{ll} \text{(a)} & M = 2\,\mathtt{rand}(m/2) - 1; \\ \text{(b)} & M = M + M^\top; \\ \text{(c)} & F_i = \mathtt{blkdiag}(M; -M), \end{array}$$

[3]A convex set $\mathcal{X} \in \mathbb{R}^n$ is said to be in isotropic position if, given a uniform random point $x \in \mathcal{X}$, it holds $\mathbb{E}[x] = 0$ and $\mathbb{E}\left[xx^\top\right] = I$, i.e., its covariance matrix is identity. We say that a convex set $\mathcal{X}$ is in near-isotropic position if $\mathtt{cg}(\mathcal{X}) = 0$ and the covariance matrix of the uniform distribution over $\mathcal{X}$ has eigenvalues between $\frac{1}{2}$ and $\frac{3}{2}$.

in order to guarantee $\mathcal{X}_{\rm LMI}$ to be bounded and to ensure the existence of a finite solution. In a number of examples, step (b) above was implemented as

$$M = \texttt{triu}\, M + (\texttt{triu}\, M)^\top - \texttt{diag}(\texttt{diag}(M)).$$

Such data were generated for dimensions of the F_i matrices as large as $m = 100$, and the dimensions of the design vector x as high as $n = 1,000$. Without loss of generality, the vector c in the objective function was taken as $c = (1\ 0\ \dots\ 0)$.

We note that the test problems chosen are illustrative, thus giving a first impression on the potentials of our method; of course, a deeper analysis is needed, which will be performed in the subsequent papers. In the experiments, the method demonstrated pretty stable performance; as far as the widely used MATLAB-based realizations of interior-point methods (the `solvesdp` routine in SeDuMi Toolbox) are concerned, it showed comparable performance, sometimes exceeding the classical methods in accuracy. To illustrate, we briefly describe some preliminary results of simulations.

For moderately sized problems with $n = 10$, $m = 10$, we obtain 7 to 9 exact digits after 45 to 55 iterations with $N = 200$ H&R-points at each step. In other words, the observed rate of convergence $(f_k - f^*)/(f_{k-1} - f^*)$ was approximately 0.65 to 0.7 as compared to the expected theoretical rate 0.59 for this dimension $n = 10$ (see (4.9)). The same accuracy is typically observed after 30 to 40 steps if $N = 500$ H&R-points were used.

In the second set of experiments the method was tested on SDP problems having large dimensions of the design vector, $n = 1,000$ and $m = 10$. Typically, the method reproduces 7 to 8 exact decimal digits for the function value after 20 to 30 iterations ($N = 1,500$ points were used). Hence, for such high-dimensional problems, a much faster convergence was observed than the predicted theoretical one 0.992. Also, as a rule, for these high-dimensional problems, the `solvesdp` routine exhibits slightly lower accuracy (6 to 7 digits); moreover, sometimes it yields a formally infeasible point x^*, e.g., $\max \mathrm{eig}\big(F(x^*)\big) \approx 2 \cdot 10^{-7} > 0$ was observed.

The third set of experiments was conducted with problems having worst-case geometry, where the feasible domain is "pyramid-like" (see Section 4.1). For such sets, the bound on the rate of convergence of the RCP algorithm given by (4.9) is known to be attained. In the experiments, the matrices F_i were chosen so as to yield

$$\mathcal{X}_{\rm LMI} \ = \ \{x \in \mathbb{R}^n\colon\ \|x\|_1 \le 1,\ x_1 < 0\},$$

so that the minimum function value is $f^* = -1$ and $m = 2^n$ by construction. Notably, for such a geometry, the isotropization procedure described above very often *does not* bring $\mathcal{X}_{\rm LMI}$ to near-isotropic position, and is therefore omitted. As a result, in practice we lean on a much smaller number N of H&R-points that would be needed to correctly restore the Y matrix.

Specifically, for the worst-case geometry problem with $n = 10$ optimization variables and quite large dimension $m = 1,024$ of the F_i matrices, using only $N = 40$ points yields 6 to 8 exact digits after 75–85 iterations depending on realization. This means that, in accordance with the theory developed here, the observed rate of convergence, approximately 0.8, is comparable to 0.69, which is given by (4.9).

In view of the numerical results above, we mention the paper [28], where a similar randomized algorithm was proposed. At every step, the cut was performed through the estimated center of gravity (averaged H&R-points) rather than through the best point as above, and an additional "projection" step was used that largely accelerated the convergence. The results of numerical tests with the same data were similar to those reported here; however, no formal theoretical claims were made in [28] regarding the convergence properties.

7. Conclusions and Future Research. In this paper, we study a novel randomized scheme for convex optimization. The theoretical behavior of the method is analyzed in the abstract framework in which a Uniform Generating Oracle is assumed to be available. A novel probabilistic analysis is carried out, and our method is shown to outperform the classical DCG methods.

Specifically, we explicitly compute the *expected* rate of convergence of the RCP method and show that the estimates derived are tight in that they are attainable at "worst-case" geometry sets. Moreover, we evaluate the *accuracy* of the estimate provided (in terms of its variance). Next, the bounds on the number N of random samples required to guarantee with prescribed *high probability* the desired convergence behavior of the algorithm are also derived explicitly. It turns out that this number is very small, thus guaranteeing practical implementability of the proposed scheme.

The final part of the paper is devoted to numerical simulations for the special case of SDP optimization and comparison of the method with the standard SDP solvers such as SeDuMi. To the best of our knowledge, this is the first time that a practical implementation of a randomized method of this type is discussed in such detail.

In the numerical illustrations we limited ourselves to a broad class of semidefinite programs. It is believed that future versions and modifications of the randomized algorithm presented here will expose better performance than the presently existing techniques, especially for problems of very high dimensions, in the presence of uncertainty in the F_i matrices, etc.

Specifically, one of the promising applications would be solving SDP problems in the "generic" form

$$\min \operatorname{Tr} CX \quad \text{subject to} \quad AX + XA^\top \preceq 0 \quad \text{and} \quad X \succeq I,$$

where C, A are given matrices, and optimization is performed with respect to the entries of the (symmetric) matrix X. It is known that the currently existing solvers cannot deal with high-dimensional problems, $\dim X \geq 200$ because of the exceedingly large memory requirements and the necessity to convert the problem to the canonical form (6.1). Instead, using our method only requires a boundary oracle, which is straightforward to construct for the feasible set in this problem on the basis of Lemma 6.1.

It also should be stressed that the applicability of the method is not limited solely to SDP problems but rather extends to general (possibly nonsmooth) convex optimization problems. As one of the possibilities, we mention semi-infinite optimization problems; specifically, consider for instance the classical Chebyshev approximation problem: Given a fixed known polynomial $s(t)$ of degree m defined for $t \in [a, b]$, find a polynomial

$$p(x,t) \;=\; \sum_{i=1}^{n} x_i t^{i-1}$$

of degree $n < m$ that minimizes the l_∞-norm distance from $s(t)$, i.e.

$$\min_x \max_{a \leq t \leq b} |s(t) - p(t,x)|. \tag{7.1}$$

Rewrite it in the following form:

$$\min v \quad \text{subject to} \quad -v \;\leq\; \sum_{i=1}^{n} x_i t^{i-1} - s(t) \;\leq\; v \;\; \text{for all} \;\; t \in [a, b] \tag{7.2}$$

in the variables $v, x_1, \ldots, x_n$. This is a convex optimization problem with infinite number of constraints, for which there are no interior-point methods available. However, with our randomized approach, this problem can be solved easily, since a boundary oracle for the feasible set is straightforward to construct. Indeed, having a feasible polynomial $p(t,x)$ with x fixed, and a "direction" polynomial $d(t)$, the problem reduces to finding the minimal and the maximal values of $\lambda \in \mathbb{R}$ retaining the validity of the inequality

$$-v \;\leq\; p(t,x) + \lambda d(t) \;\leq\; v.$$

This problem in turn readily reduces to finding real roots of an appropriate derivative, which is polynomial in t.

We conclude our exposition by noting that preliminary numerical tests conducted with our method testify to its practical vitality, and it looks quite promising to outperform classical deterministic optimization methods.

REFERENCES

[1] H. Alzer, *Some Beta function inequalities*, Proceedings of the Royal Society of Edinburgh, 113A (2003), pp. 731–745.

[2] E.-W. Bai, R. Tempo, and M. Fu, *Worst-case properties of the uniform distribution and randomized algorithms for robustness analysis*, Mathematics of Control, Signals, and Systems, 11 (1998), pp. 183–196.

[3] D. Bertsimas and S. Vempala, *Solving convex programs by random walks*, Journal of the ACM, 51 (2004), pp. 540–556.

[4] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, SIAM, Philadelphia, 1994.

[5] G. Calafiore, *Random walks for probabilistic robustness*, in Proceedings of the IEEE Conference on Decision and Control, 2004, pp. 5316–5321.

[6] G. Calafiore and F. Dabbene, *A probabilistic analytic center cutting plane method for feasibility of uncertain LMIs*, Automatica, 43 (2006), pp. 2022–2033.

[7] G. Calafiore and F. Dabbene (Eds.), *Probabilistic and Randomized Methods for Design under Uncertainty*, Springer-Verlag, London, 2006.

[8] F. Dabbene, *A randomized cutting plane scheme for convex optimization*, in Proc. of IEEE Multiconference on Systems and Control, San Antonio, 2008.

[9] F. Dabbene, P.S. Shcherbakov, and B.T. Polyak, *A randomized cutting plane scheme with geometric convergence: Probabilistic analysis and SDP applications*, in Proc. of the 47th IEE Conference on Decision and Control, 2008.

[10] H.A. David and H.N. Nagaraja, *Order Statistics (3rd Edition)*, Wiley, New Jersey, 2003.

[11] M.E. Dyer, A.M. Frieze, and R. Kannan, *A random polynomial-time algorithm for approximating the volume of convex bodies*, Journal of the ACM, 38 (1991), pp. 1–17.

[12] R.J. Gardner, *The Brunn-Minkowski inequality*, Bull. AMS, 39 (2002), pp. 355–405.

[13] J.-L. Goffin and J.-P. Vial, *Convex non-differentiable optimization: a survey focused on the analytic center cutting plane method*, Optimization Methods & Software, 17 (2002), pp. 805–867.

[14] M. Grant, S. Boyd, and Y. Ye, *Disciplined convex optimization*, in Global Optimization: from Theory to Implementation, L. Liberti and N. Maculan, eds., Nonconvex Optimization and Its Applications, Springer, New York, 2006, pp. 155–210.

[15] B. Grünbaum, *Partitions of mass-distributions and convex bodies by hyperplanes*, Pacific J. Math., 10 (1960), pp. 1257–1261.

[16] R. Kannan, L. Lovász, and M. Simonovits, *Random walks and an $O^*(n^5)$ volume algorithm for convex bodies*, Random Structures and Algorithms, 11 (1997), pp. 1–50.

[17] J.E. Kelley, *The cutting plane method for solving convex programs*, Journal of the Society for Industrial and Applied Mathematics, 8 (1960), pp. 703–712.

[18] L.G. Khachiyan, *A polynomial time algorithm for linear programming*, Soviet Math. Doklady, 20 (1979), pp. 191–194.

[19] A.Y. Levin, *On an algorithm for the minimization of convex functions over convex functions*, Soviet Math. Doklady, 6 (1965), pp. 286–290.

[20] J. Löfberg, *YALMIP: A toolbox for modeling and optimization in MATLAB*, in Proceedings of the CACSD Conference, 2004.

[21] L. Lovász, *Random walks on graphs: A survey*, in Combinatorics, Paul Erdös is Eighty, V. T. Sós D. Miklós and T. Szönyi, eds., Jnos Bolyai Mathematical Society, Budapest, 1996, pp. 353–398.

[22] ———, *Hit-and-Run mixes fast*, Mathematical Programming, 86 (1999), pp. 443–461.

[23] L. Lovász and M. Simonovits, *Random walks in a convex body and an improved volume algorithm*, Random Str. and Alg., 4 (1993), pp. 359–412.

[24] R. Motwani and P. Raghavan, *Randomized Algorithms*, Cambridge University Press, Cambridge, 1995.

[25] Y. Nesterov and A.S. Nemirovski, *Interior Point Polynomial Algorithms in Convex Programming*, SIAM Journal on Applied Mathematics, Philadelphia, 1994.

[26] D.J. Newman, *Location of the maximum on unimodal surfaces*, Journal of the ACM, 12 (1965), pp. 395–398.

[27] B.T. Polyak and P.S. Shcherbakov, *The D-decomposition technique for linear matrix inequalities*, Automat. Remote Control, 67 (2006), pp. 159–174.

[28] ———, *A randomized method for solving semidefinite programs*, in Proc. of the 9th IFAC Workshop – Adaptation and Learning in Control and Signal Processing (ALCOSP'07), 2007. Available from *IPACS Electronic Library* at `http://lib.physcon.ru/`.

[29] L. Rademacher, *Approximating the centroid is hard*, in Proc. of the 24th Annual Symposium on Computationa Geometry (SoCG-07), 2007.

[30] L. Rademacher and S. Vempala, *Testing geometric convexity*, in Proc. of the 23rd Foundations of Software Technology and Theoretical Computer Science (FSTTCS-04), 2004.

[31] J. Radon, *Über eine Erweiterung des Begriffs der konvexen Functionen, mit einer Anwendung auf die Theorei der konvexen Körper*, S. B. Akad. Wiss. Wien, 125 (1916), pp. 241–258.

[32] M. Rudelson, *Random vectors in the isotropic position*, J. Funct. Anal., 164 (1999), pp. 60–72.

[33] N.Z. Shor, *Cut-off method with space dilation in convex programming problems*, Cybernetics, 13 (1977), pp. 94–96.

[34] R.L. Smith, *Efficient Monte-Carlo procedures for generating points uniformly distributed over bounded regions*, Operations Research, 32 (1984), pp. 1296–1308.

[35] J. Sturm, *Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones*, Optimizatuion Methods and Software, 11–12 (1999).
[36] R. Tempo, G. Calafiore, and F. Dabbene, *Randomized Algorithms for Analysis and Control of Uncertain Systems*, Communications and Control Engineering Series, Springer-Verlag, London, 2004.
[37] V. F. Turchin, *On calculation of multi-dimensional integrals via the Monte Carlo method*, Theory of Probability and Appl., 16 (1971), pp. 720–724.
[38] S. Vempala, *Geometric random walks: a survey*, Combinatorial and Computational Geometry, 52 (2005), pp. 573–611.
[39] D.B. Yudin and A.S. Nemirovski, *Informational complexity and efficient methods for solving complex extremal problems*, Matekon, 13 (1977), pp. 25–45.