A simple randomised algorithm for convex optimisation : application to two-stage stochastic programming

technische universiteit eindhoven

SPOR-Report 2002-05

A simple randomised algorithm for convex optimisation

Application to two-stage stochastic programming

M. Dyer
R. Kannan
L. Stougie

/ department of mathematics and computer science

A simple randomised algorithm for convex optimisation

Application to two-stage stochastic programming

M. Dyer, [1] R. Kannan [2] and L. Stougie [3]

Abstract

We consider maximising a concave function over a convex set by a simple randomised algorithm. The strength of the algorithm is that it requires only approximate function evaluations for the concave function and a weak membership oracle for the convex set. Under smoothness conditions on the function and the feasible set, we show that our algorithm computes a near-optimal point in a number of operations which is bounded by a polynomial function of all relevant input parameters and the reciprocal of the desired precision, with high probability.

As an application to which the features of our algorithm are particularly useful we study two-stage stochastic programming problems. These problems have the property that evaluation of the objective function is #P-hard under appropriate assumptions on the models. Therefore, as a tool within our randomised algorithm, we devise a fully polynomial randomised approximation scheme for these function evaluations, under appropriate assumptions on the models. Moreover, we deal with smoothing the feasible set, which in two-stage stochastic programming is a polyhedron.

1 Introduction

In this paper we develop a randomised approximation algorithm for certain convex optimisation problems, defined as

$$\begin{array}{rl} \max & G(x) \\ \text{subject to} & x \in S, \end{array}$$

[1] Department of Computer Science, University of Leeds, Leeds, UK; e-mail: dyer@comp.leeds.ac.uk

[2] Department of Computer Science, Yale University, New Haven, CT; e-mail: kannan@cs.yale.edu

[3] Department of Mathematics and Computer Science, Eindhoven University of Technology and CWI Amsterdam, The Netherlands; e-mail:leen@win.tue.nl, Supported by the TMR Network DONET of the European Community ERB TMRX-CT98-0202

where $G : \mathbb{R}^n \to \mathbb{R}$ is a concave function and $S \subset \mathbb{R}^n$ is a convex set. The *weak optimisation* version of this problem, finding a point $x \in S$ with function value within ϵ of the optimal (cf. [9]), can be solved in a polynomial number of basic computer operations [9, 15]. Generally, known polynomial time algorithms use a separation oracle for S (and level sets of $G(\cdot)$). While this can be simulated by a membership oracle for S (and function evaluations for $G(\cdot)$), in polynomial time, the simulation is very expensive. We wish to avoid altogether the use of separation oracles.

As an alternative, we present a simple randomised algorithm based on local moves. At each iteration, we choose a random point in a small ball centred at the current feasible point. We move to it if it is feasible and the objective function is strictly better. Otherwise, we stay at the current point and repeat the random selection.

The algorithm requires only a membership oracle for S and an approximate evaluation oracle for $G(\cdot)$ (which returns an approximate function value in the queried point). We show that with high probability our algorithm outputs a solution that is within ϵ of the optimal solution value. Under reasonable smoothness conditions on the feasible region and the function to be optimised, the number of oracle calls required is bounded by a polynomial function of the size of various input parameters. In Section 2 we present and analyse our randomised convex optimisation algorithm.

An important application of our result is to stochastic programming. We consider randomised approximations to optimal solutions of *two-stage stochastic programming problems.* Problems of this type have been studied since they were proposed in the 1950's [2], [4], [17]. They model optimisation under *uncertainty.* In Section 3 we give a brief introduction to these problems. In Section 4 we review their complexity.

In sharp contrast to ordinary linear programs, two-stage stochastic programs are hard to solve in a well defined sense. In fact, even a single evaluation of the objective function may be computationally hard for a two-stage program. Thus, simply assuming the existence of (even) an approximate function evaluation oracle for these evaluations conceals the intrinsic complexity of the problem. Assuming higher order function information is also undesirable, since derivatives are numerically unstable with respect to relative approximation. Therefore, an application of the usual solution methods for convex optimisation is problematic.

We design an subroutine for approximate evaluations of the objective function of two-stage stochastic programming problems in Section 5. This subroutine is again a randomised algorithm, which, with high probability, produces a function value that is within any prescribed precision. Under appropriate assumptions on the randomness in the two-stage stochastic programming problem, the number of steps required is bounded by a polynomial function of the size of input parameters of the function to be evaluated and of the logarithm of the reciprocal of the desired precision, making our subroutine a *fully polynomial randomised approximation scheme.* We achieve this result by drawing on known techniques, but to our best knowledge it is a new result.

In Section 6 we combine the subroutine of Section 5 with the randomised convex optimisation algorithm developed in Section 2 to yield an algorithm for solving two-stage stochastic programming problems. It turns out that the conditions we place on the input of the stochastic programming problem, in order to get approximate function values, imply the smoothness requirement on the objective function which we need for our convex optimisation algorithm to converge in a polynomial number of steps.

2 Random local improvement

In this section we consider the general problem of maximizing a twice differentiable concave real-valued function $G : \mathbb{R}^n \to \mathbb{R}$, over a compact convex set $S \subset \mathcal{D}$. We will assume little about the function G and the set S. We assume S is given only by a *membership oracle*, which can decide, for a given $x \in \mathbb{R}^n$, whether or not $x \in S$. We assume G is given by an *approximation oracle*, which for a given error parameter $\epsilon > 0$, and $x \in \mathbb{R}^n$, returns a number in the interval $[G(x) - \epsilon, G(x) + \epsilon]$. In the sequel we denote the optimal solution of the maximisation problem by x^*.

We propose a very simple solution strategy. Starting from a given initial feasible point $x_0 \in S$, we successively generate points in S as follows. At $x \in S$, we generate a point in a ball of a certain radius r and centre x uniformly at random. If this point is feasible (i.e. in S) and has a significantly better objective function value than x, we move to it and iterate. Otherwise, we repeat the random generation. We stop the algorithm if a certain number of successive trials have not given a significantly better point. Thus we look simply for a local random move which improves the objective function. We call this the "Ball Walk algorithm".

This strategy does not lead to an efficient method for general concave functions and convex sets. For example, if $S = \mathbb{R}^n_+$, and our current point is the origin, we have an exponentially small probability of hitting S. This example illustrates one problem - poor local conductance in the terminology of [14]. However, we will show that, under mild smoothness conditions, the method converges rapidly.

In the sequel we denote the volume of a set S by vol(S), and $B(x,r)$ denotes a ball with radius r and centre x. A *cap* of $B(x,r)$ is the subset cut off by a half-space which excludes x. We denote the *unit* n-ball by B_n. We use ∂S to denote the boundary of a set S. We denote the first and second directional derivatives of a function $F : \mathbb{R}^n \to \mathbb{R}$ in direction w by $F'(w;x)$ and $F''(w;x)$, respectively. The gradient of a function F at a point $x \in \mathcal{D}$ will be denoted by $\nabla F(x)$, and its Hessian by $\nabla^2 F(x)$. We denote the Euclidean norm of a vector x by $\|x\|$, and the L_2-norm of a matrix A by $\|A\|$, i.e., $\|A\| = \max_{x:\|x\|=1} \|Ax\|$. Thus $|F'(w;x)| \leq \|\nabla F(x)\|$ and $|F''(w;x)| \leq \|\nabla^2 F(x))\|$ for all x and $w \in \partial B_n$, and equality holds in both cases for every x and some $w(x) \in \partial B_n$. We now list the assumptions that we make.

Assumptions A:

1. G is concave and $S \subset \mathbb{R}^n$ is convex and and has diameter bounded above by a constant D;
2. there exists $\tau > 0$ such that, for all $x \in S$, $\|\nabla G(x)\| \leq \tau$;
3. there exists $\nu > 0$, such that, for all $x \in S$, $\|\nabla^2 G(x)\| \leq \nu$;
4. there exist $\sigma, r_0 > 0$, such that, for all $r \leq r_0, x \in S$,
$$\frac{\text{vol}(B(x,r) \cap S)}{\text{vol}(B(x,r))} \geq \frac{1}{2} - \sigma r;$$
5. Some point $x_0 \in S$ is given.

The Ball Walk algorithm aims at finding a point x for which $G(x) \geq G(x^*) - \epsilon\tau D$ with probability at least $1 - \eta$. It is described in Figure 1, where we choose $r = \min\{r_0, \frac{D}{\sqrt{n}}, \frac{\epsilon\tau}{90\sigma\tau+3\nu\sqrt{n}}\}$. The improvement we get in one step of Ball Walk will be examined below in Theorem 2.1. But first we give some elementary estimates which will prove useful.

Lemma 2.1 *For $n \geq 2$, $\sqrt{n}/3 \leq$ vol$(B_{n-1})/$vol$(B_n) \leq 2\sqrt{n}/3$.*

```
Procedure Ball Walk
begin $x \leftarrow x_0$ and $counter \leftarrow 1$;
        while $counter \leq \frac{120}{\sigma r} \ln\left(\frac{6\sqrt{n}D\ln(1/\epsilon)}{\eta r}\right)$;
                generate $x' \in B(x,r)$ uniformly at random;
                        if $x' \in S$ and $G(x') > G(x) + \frac{\epsilon\tau r}{6\sqrt{n}}$,
                                then $x \leftarrow x'$ and $counter \leftarrow 1$;
                        else set $counter \leftarrow counter + 1$;
        return $x$ as the approximate solution.
end;
```

Figure 1: The Ball Walk Algorithm

PROOF. Since $\text{vol}(B_n) = \pi^{n/2}/(n/2)!$,

$$\frac{\text{vol}(B_{n-1})}{\text{vol}(B_n)} = \frac{(n/2)!}{\sqrt{\pi}((n-1)/2)!}.$$

Stirling's approximation to the factorial (in the inequality form) gives the conclusion after some calculation. □

Lemma 2.2 *For $n \geq 2$, let $U = \{x = (x_1, \ldots, x_n) \in B(0,r) | 0 \leq x_1 \leq cr/\sqrt{n}\}$ be a slice of $B(0,r) \subset \mathbb{R}^n$, where $0 < c < 1$. Then $c/5 \leq vol(U)/vol(B) \leq 2c/3$.*

PROOF. Since $c \leq 1$, elementary estimates give

$$\frac{c}{\sqrt{n}}\left(1 - \frac{1}{n}\right)^{\frac{n-1}{2}} \frac{\text{vol}(B_{n-1})}{\text{vol}(B_n)} \leq \frac{\text{vol}(U)}{\text{vol}(B)} \leq \frac{c}{\sqrt{n}} \frac{\text{vol}(B_{n-1})}{\text{vol}(B_n)}.$$

Using Lemma 2.1 and $(1 - 1/n)^{n-1} \geq e^{-1}$ now gives the conclusion. □

Theorem 2.1 *Let $n \geq 3$, and $x \in S$ be such that $G(x) \leq G(x^*) - \epsilon\tau D$. With probability at least $\sigma r/120$ a point z is found in one step with*

$$G(z) \geq G(x) + \frac{\epsilon\tau r}{2\sqrt{n}}.$$

PROOF. First shrink $B(x,r)$ to $B(x,r')$, with $r' = (1-\beta)r$, and $\beta = \frac{6}{n}$. We write B and B' shortly for $B(x,r)$ and $B(x,r')$, respectively. Notice that

$$\text{vol}(B') = (1-\beta)^n \text{vol}(B).$$

Define $g(x) = \frac{\nabla G(x)}{\|\nabla G(x)\|}$, and $v = \frac{x^* - x}{\|x^* - x\|}$. Consider the set

$$T_1 = B' \cap S \cap \{y | (y - x)g(x) \geq -\frac{5c_1 r'}{\sqrt{n}}\} \cap \{y | (y - x)v \leq \frac{5c_2 r'}{\sqrt{n}}\},$$

with $c_1 = 3\sigma r'$ and $c_2 = \frac{1}{2} - \sigma r'$. Note that the bound on r implies that $c_1 < \epsilon/2 \leq \frac{1}{2}$, and clearly $c_2 < \frac{1}{2}$. Now T_1 is obtained from B' by cutting off the union of two caps and therefore, using Lemma 2.2,

$$\mathrm{vol}(T_1) \geq \mathrm{vol}(B' \cap S) - (\frac{1}{2} - c_2)\mathrm{vol}(B') - (\frac{1}{2} - c_1)\mathrm{vol}(B').$$

Thus, using Assumption A.4 and the definition of c_1 and c_2,

$$\frac{\mathrm{vol}(T_1)}{\mathrm{vol}(B')} \geq \frac{1}{2} - \sigma r' - \sigma r' - \frac{1}{2} + 3\sigma r' = \sigma r'.$$

Let $\alpha = \frac{r}{3D\sqrt{n}}$, and define the set

$$T_2 = \{\alpha x^* + (1 - \alpha)y | y \in T_1\}.$$

We claim that T_2 is a subset of B, each point of which gives the improvement stated in the theorem. Thus, its relative volume is a lower bound on the probability that such an improvement is attained in one step of the Ball Walk algorithm. This relative volume is in turn bounded as follows:

$$\begin{aligned} \frac{\mathrm{vol}(T_2)}{\mathrm{vol}(B)} \geq \frac{(1-\alpha)^n \mathrm{vol}(T_1)}{(1-\beta)^{-n}\mathrm{vol}(B')} &\geq (1-\alpha)^n (1-\beta)^n \sigma r' \\ &= (1-\alpha)^n (1-\beta)^{n+1} \sigma r \\ &\geq (8/9)^3 (1/3)^4 \sigma r \\ &\geq \sigma r / 120, \end{aligned}$$

where the last but one inequality is implied by $r \leq \frac{D}{\sqrt{n}}$, $n \geq 3$, and the choices for α and β.

To show that $T_2 \subset B$, we take $z = \alpha x^* + (1 - \alpha)y$ for some $y \in T_1$. and show that $\|z - x\| \leq r$.

$$\|z - x\|^2 = \alpha^2 \|x^* - x\|^2 + (1-\alpha)^2 \|y - x\|^2 + 2\alpha(1-\alpha)(x^* - x)(y - x). \quad (1)$$

To bound the first term of the right-hand side of (1), we use the definition of α and the fact that $\|x^* - x\| \leq D$, giving

$$\alpha^2 \|x^* - x\|^2 \leq \left(\frac{r}{3D\sqrt{n}}\right)^2 D^2 = \frac{r^2}{9n}. \quad (2)$$

Since $y \in B'$ implies that $\|y - x\| \leq r' = (1-\beta)r$ and $0 \leq \alpha, \beta \leq 1$ implies that $(1-\alpha)^2(1-\beta)^2 \leq (1-\beta)$, the second term of the right-hand side is bounded by

$$(1-\alpha)^2\|y-x\|^2 \leq (1-\beta)r^2 = (1-\frac{2}{n})r^2. \tag{3}$$

Finally, the definitions of α, v, c_2 and T_1 imply

$$2\alpha(1-\alpha)(x^* - x)(y - x) \leq \frac{10}{3}(1-\beta)c_2\frac{r^2}{n} \leq \frac{5r^2}{3n}. \tag{4}$$

(2), (3), (4) inserted in (1) yields

$$\|z - x\|^2 \leq \left(\frac{1}{9n} + 1 - \frac{2}{n} + \frac{5}{3n}\right) r^2 \leq r^2.$$

Next we show that z gives the desired improvement over x. By concavity of G

$$G(z) - G(x) \geq \alpha(G(x^*) - G(x)) + (1-\alpha)(G(y) - G(x)). \tag{5}$$

The second order Taylor expansion of G in y around x yields

$$\begin{aligned} G(y) - G(x) &= \nabla G(x)(y-x) + \frac{1}{2}G''(w;x')\|y-x\|^2 \\ &\geq -\frac{5c_1r'}{\sqrt{n}}\|\nabla G(x)\| - \frac{1}{2}\nu r'^2 \\ &\geq -\frac{5c_1r}{\sqrt{n}}\|\nabla G(x)\| - \frac{1}{2}\nu r^2 \\ &\geq -\frac{15\sigma r^2}{\sqrt{n}}\|\nabla G(x)\| - \frac{1}{2}\nu r^2, \end{aligned} \tag{6}$$

where $x' \in [x, y]$, $w = (y-x)/\|y-x\|$, and we used Assumption A.3 for the first inequality. Using the definition of α, (5) and (6) yield

$$G(z) - G(x) \geq \frac{r}{3D\sqrt{n}}(G(x^*) - G(x)) - \left(\frac{15\sigma}{\sqrt{n}}\|\nabla G(x)\| + \frac{1}{2}\nu\right) r^2.$$

Since we assumed $G(x^*) - G(x) \geq \epsilon\tau D$, using $\|\nabla G(x)\| \leq \tau$ (Assumption A.2), leads to

$$\frac{G(z) - G(x)}{G(x^*) - G(x)} \geq \frac{r}{3D\sqrt{n}} - \left(\frac{15\sigma\tau/\sqrt{n}}{\epsilon\tau D} + \frac{\nu/2}{\epsilon\tau D}\right) r^2.$$

Since $r \leq \frac{\epsilon\tau}{90\sigma\tau + 3\nu\sqrt{n}}$, a simple calculation now gives

$$\frac{G(z) - G(x)}{G(x^*) - G(x)} \geq \frac{r}{6D\sqrt{n}}$$

□

The justification of the Ball Walk algorithm follows as a corollary.

Theorem 2.2 *The number of samplings from a ball that Ball Walk requires to reach, with probability at least $1-\eta$, a point x with $G(x^*) - G(x) \leq \epsilon\tau D$ is bounded from above by*

$$\frac{720\sqrt{n}D\ln(1/\epsilon)}{\sigma r^2}\ln\left(\frac{6\sqrt{n}D\ln(1/\epsilon)}{\eta r}\right).$$

PROOF. The probability of finding an improving point given in Theorem 2.1 implies that, if the inner loop of the Ball Walk algorithm runs $\frac{120}{\sigma r}\ln\left(\frac{6\sqrt{n}D\ln(1/\epsilon)}{\eta r}\right)$ of times, then the probability of incorrect termination is at most $\frac{\eta r}{6\sqrt{n}D\ln(1/\epsilon)}$.

The one step improvement in Theorem 2.1 implies

$$\frac{G(x^*) - G(z)}{G(x^*) - G(x)} \leq 1 - \frac{r}{6D\sqrt{n}}.$$

Moreover, by concavity of G and Assumption A.2, $G(x^*) - G(y) \leq \tau D$ for any feasible point y, hence also for the starting point. Thus, after k improvement steps we obtain a point x^k with

$$G(x^*) - G(x^k) \leq (1 - \frac{r}{6D\sqrt{n}})^k \tau D.$$

Therefore,

$$\frac{\log(1/\epsilon)}{\log\left(6\sqrt{n}D/(6\sqrt{n}D - r)\right)} \leq \frac{6\sqrt{n}D\log(1/\epsilon)}{r}$$

improvement steps are sufficient to obtain the desired precision $\epsilon\tau D$. Multiplying the error probability of incorrect termination by the number of improvement steps yields the overall incorrect termination is bounded by η. □

Note that the bound of Theorem 2.2 is indeed polynomial in the parameters of the problem, since r is a rational function of the problem parameters.

3 Two-stage stochastic programming

In this section we describe briefly stochastic linear programming problems. Problems of this type have been studied since they were proposed in the mid 1950's [2], [4], [17]. They model optimisation under *uncertainty.* Such models are useful in many practical situations. Obtaining exact information about all parameters in a practical optimisation problem is often impossible.

As an example, think of allocating funds to a variety of possible investments so as to maximise profit under a budget restriction. Usually at the moment the investment decision is to be made there is no certainty at all about the future yields of the various investments. Neither might there be exact information about the amounts needed to invest in a certain given project. At best one might hope to have some idea of what these parameter values could be, and to express this in the form of probability distributions. In this way we arrive at stochastic programming problems.

Suppose that we have a linear programming problem in which some parameters are random. The random variables we indicate by putting a tilde over them.

$$\begin{array}{rl} \max & px \\ \text{subject to} & Ax \leq b \\ & \tilde{T}x \leq \tilde{\xi} \end{array}$$

with $b \in \mathbb{R}^m$, $\tilde{\xi} \in \mathbb{R}^d$, and A an $m \times n$ matrix, $\tilde{T}$ an $d \times n$ matrix, and $p \in \mathbb{R}^n$.

We assume that probability distributions are given for the random matrix $\tilde{T}$ and the random vector $\tilde{\xi}$. The above model is clearly ill-defined since a solution x that is optimal for one realisation of $\tilde{T}$ and $\tilde{\xi}$ may even be infeasible for another.

Two main directions have been taken in the literature to arrive at sensible models. In the conceptually easiest, violation of the uncertain constraints is allowed to occur with a probability that does not exceed a prespecified level, giving the so-called *probabilistic constraints problem.* The best comprehensive survey of this field is [16]. The paper of Kannan and Nolte [13] takes a similar approach to the probabilistic constraints problem that we take here for the model described next.

The other approach is the one we consider in this paper and is called the *two-stage stochastic programming problem* or the *stochastic recourse problem.*

Conceptually one should think of the decision process taking place in two *stages.* In the first, values for the *first stage variables* x are chosen. In the second, upon a realisation of the random parameters, a *recourse* action is to be taken in case of infeasibilities. Costs are attached to the various possible recourse actions leading to the *second stage* (or recourse) problem, to choose the optimal action given the infeasibilities. The expected cost of the optimal recourse action is then added to the objective function. For a comprehensive review of the extensive literature we refer to [3], [8], [16].

A generic mathematical programming formulation for this problem is

$$\begin{array}{rl} \max & px + \mathbf{E}[\max\{\tilde{q}y \mid Wy \leq \tilde{T}x - \tilde{\xi},\ y \in \mathbb{R}^{n_1}\}] \\ \text{subject to} & Ax \leq b. \end{array} \tag{7}$$

with $\tilde{q} \in \mathbb{R}^{n_1}$ and W an $d \times n_1$ matrix. In the literature W is sometimes allowed to be a random matrix. However, this may cause the feasible region to be non-convex in terms of x (see [18]). We concentrate on the so-called *fixed recourse model* in which W is fixed. Moreover, we assume that W is such that for any x and any realisation of $\tilde{T}$ and $\tilde{\xi}$ there exists a feasible solution y in the second stage problem. This property of W is called the *complete recourse property*, and the model is accordingly called the *complete recourse model* (see e.g. [3]).

It is well known that the objective function of (7) is *concave* (see [19, 18]). Therefore, the two-stage stochastic programming problem boils down to maximising a concave function over a convex (polyhedral) set. Thus, we can use our Ball Walk algorithm to solve this problem if we know that the objective function and the convex feasible set satisfy our smoothness conditions. It will be clear that this is not true for the feasible set, which is a polyhedron. We will come to this point later, in Section 6. However, another serious obstruction against using Ball Walk is that this algorithm requires an oracle that gives function values on request. As will be clear from the next section, it is exactly the evaluation of the objective function which makes the two-stage stochastic programming problem so excessively hard to solve. Therefore, the assumption that a function evaluation oracle exists for these problems significantly hides their computational difficulty.

Therefore, before adapting the Ball Walk algorithm to solve two-stage stochastic programming problems in Section 6, we first devise a suitable function evaluation oracle in Section 5.

4 Complexity

It is shown in [7] that, even in case there is independence between the coordinates of the random parameters, the complexity of two-stage stochastic optimisation is essentially determined by the complexity of a single evaluation of the objective function, and that this may be a very hard computational problem. Specifically, it is shown that exact solution of (7) is #P-hard in general. Thus exact solution for such problems is apparently intractable for polynomial time computations; it is at least as hard as exact counting.

On the other hand, although *exact* counting is usually hard, there are situations where *randomised approximate* counting is possible. See, for example, [10]. Therefore we might guess that a similar type of approximation would be possible here. We show below that this intuition is justified. We will devise a *fully polynomial randomised approximation scheme (fpras)* for the case when the random parameters have a *log-concave* distribution. A key ingredient is the fact that, in this case, evaluation of the objective function is equivalent in complexity to the computation of the volume of a convex body [1]. An *fpras* for the objective function evaluations follows from this equivalence, as we show below.

5 Computation of the objective function

As we pointed out in the last section, the main difficulty in solving the two-stage stochastic programming problem is the computation of the objective function. We concentrate in the rest of the paper on the version of (7) in which only the right hand side coefficients $\tilde{\xi}$ are random. Thus, q and T are fixed. We also suppress the tilde on ξ. We use the notation G for the objective function, i.e.

$$\max_{x \in S} \quad G(x) = px + Q(x), \tag{8}$$

with

$$Q(x) = \mathbf{E}_{\xi}[\max\{qy | Wy \leq Tx - \xi, y \in \mathbb{R}^{n_1}\}].$$

and

$$S = \{x \in \mathbb{R}^n | Ax \leq b\}$$

In this section we describe a *fpras* for evaluating $Q(x)$ and therefore for evaluating the objective function $G(x)$. It is based on a Markov chain approach, where we sample approximately according to the known density function of ξ, compute the value of the linear program and take the average over the values obtained from the sample.

The only source of randomness is ξ, which we assume is described by a given density function $f : \mathbb{R}^d \to \mathbb{R}$. Thus,

$$Q(x) = \int v(Tx - \xi) f(\xi) d\xi$$

with

$$v(Tx - \xi) = \max\{qy | Wy \leq Tx - \xi, y \in \mathbb{R}^{n_1}\}.$$

We require some mild conditions on f. We cannot expect to approximate Q efficiently for arbitrary f, since it is known that there exist counting functions which are NP-hard to approximate [10]. Therefore we assume the following conditions, borrowed from volume computation [1].

Assumptions B:

1. f is log-concave, i.e. $\log f$ is concave on its support $\operatorname{supp} f$;
2. f has a negligible measure outside $B(0, R)$, i.e. there exists $R \geq 4d$ such that $\int_{\|\xi\| \geq R} f(\xi) d\xi \leq \varphi$, for some φ;
3. $\log f$ is Lipschitz-continuous, i.e. there exists $\theta > 0$ such that $|\log f(\xi) - \log f(\xi')| \leq \theta \|\xi - \xi'\|$ for all $\xi, \xi' \in \operatorname{supp} f$;
4. We are given a $\xi_0 \in B(0, R)$ with $f(\xi_0) \geq R^{-\gamma d}$ for some absolute constant γ. This implies that we know a good starting point for the Markov chain, a so called "warm start";[4]
5. v is uniformly bounded, i.e. there exists L such that for all $\xi \in \operatorname{supp} f$ and $x \in S$, $|v(Tx - \xi)| \leq L$;
6. The rows of T are scaled to have unit norm. Thus $\|T\| \leq \sqrt{d}$.
7. There exist $x_0 \in \mathbb{R}^n$, $R_{\text{in}}, R_{\text{out}} \in \mathbb{R}_+$ such that $B(x_0, R_{\text{in}}) \subseteq S \subseteq B(x_0, R_{\text{out}})$.

[4] If necessary, such a point can be found with a preliminary convex optimisation.

These conditions do not severely restrict the instances, since (with minor technical changes) all the most important distributions, for example those from the exponential family, meet the requirements.

Condition B.5 is a little more restrictive than the usual assumption in two-stage stochastic programming that the function v is bounded. Boundedness only is already assured if $\mathbf{E}[\xi]$ is finite (see e.g. [18]). Condition B.6 is clearly not restrictive, and B.7 is discussed in the next section. We don't use either of B.6 or B.7 until the following section, but give them here to have a complete overview of all our conditions.

To sample according to f, we define a Metropolis random walk that has f restricted to $B(0, R)$ as its steady state density function. Note that, by Assumption B.2, the restricted density $\hat{f}$ satisfies $f \leq \hat{f} \leq f/(1-\varphi)$. If φ is assumed negligible, we can suppose that this simply contributes a negligible amount to the approximation error for $Q(x)$. Thus we will not need to draw a distinction between f and $\hat{f}$ in what follows and we will use the notation f, though it should read $\hat{f}$.

To sample from f we define a random walk that has f as its steady state density. One step of the random walk is defined as follows. Suppose the walk is at $x \in B(0, R)$. We choose uniformly at random a point y in a ball with center x and radius δ, to be specified later. If y is not in $B(0, R)$, we stay at x. Otherwise, if $f(y) \geq f(x)$, we move to y, else we move to y with probability $f(y)/f(x)$. Formally, the transition kernel $p(x, y)$ $(x \neq y)$ for moving from x to y in one step of the random walk is given by

$$p(x,y) \quad = \quad \begin{cases} \dfrac{1}{\mathrm{vol}(B(0,\delta))} \min\left\{1, \dfrac{f(y)}{f(x)}\right\} & \text{if } y \in B(0,R),\ 0 < \|x-y\| \leq \delta, \\ 0 & \text{otherwise.} \end{cases}$$

Note that $p(x, x)$ is an atom $1 - \int_{y \neq x} p(x, y)\, dy$. It is easy to see, by time reversibility, that the steady state distribution of this random walk has density proportional to $f(x)$.

In the following we will use techniques and results from volume estimation [6, 12] to prove that this chain mixes rapidly, i.e. converges fast to the steady state. We first introduce some notation and state the relevant results from the literature.

Theorem 5.1 *Dyer and Frieze [5]*
Let $S \subset \mathbb{R}^d$ be a convex body with diameter D and f be a log-concave function defined on S with π the induced measure. Let $S_1, S_2 \subset S$, $t \leq$

dist $(S_1, S_2) = \min_{x \in S_1, y \in S_2} \|x - y\|$. *If* $S_0 = \mathcal{S} \setminus (S_1 \cup S_2))$, *then*

$$\min\{\pi(S_1), \pi(S_2)\} \leq \frac{D}{2t}\pi(S_0).$$

$2/d$ is called the isoperimetric constant $Iso(\mathcal{S})$ of $\mathcal{S}$. □

Given a random walk with stationary distribution π defined on a set $\mathcal{S}$, its *conductance* is defined as

$$\Phi = \inf_{\{S \subset \mathcal{S} | 0 < \pi(S) \leq 1/2\}} \frac{\int_S P_u(\bar{S}) d\pi(u)}{\pi(S)},$$

where $P_u(\bar{S})$ is the probability of moving in one step from point u in $S \subset \mathcal{S}$ to a point in $\bar{S}$, the complement of S in $\mathcal{S}$.

The *local conductance* of a Markov chain at a point x is defined as the probability of moving to any point $y \neq x$ in one step.

Theorem 5.2 *Kannan [11]*
Consider a Metropolis random walk using balls of radius δ in a convex set $\mathcal{S} \subset \mathbb{R}^d$ which has local conductance at least χ at every point. Then the conductance Φ of the walk is at least

$$\frac{Iso(\mathcal{S})\chi^2\delta}{32\sqrt{d}}.$$

□

Theorem 5.3 *Lovasz and Simonovits [14]*
Let f_0 be the initial density of a Markov chain on $\mathbb{R}^d$ with stationary density f. Define

$$M(f_0, f) = \sup_{\{S \subset \mathbb{R}^d | 0 < \int_S f \leq 1/2\}} \frac{|\int_S f - \int_S f_0|}{\sqrt{\int_S f}}.$$

Let f_k be the density of the Markov chain after k steps. Then

$$\sup_{S \subset \mathbb{R}^d} \left| \int_S f - \int_S f_k \right| < M(f_0, f) \left(1 - \frac{\Phi^2}{2}\right)^k.$$

□

```
Procedure Estimate
begin
        $Q \leftarrow 0$;
        $K \leftarrow \left\lceil \frac{8L^2 \ln(2/\rho)}{\epsilon^2} \right\rceil$;
        for $\ell \leftarrow 1, \ldots, K$ do;
                run the Metropolis walk for $K'$ steps,
                starting from $\xi_0$;
                compute $v(Tx - \xi)$ for the resulting sample $\xi$;
                $Q \leftarrow Q + v(Tx - \xi)$;
        endfor;
        $Q \leftarrow \frac{Q}{K}$;
end;
```

Figure 2: The approximation algorithm for $Q(x)$

Theorem 5.4 *Using the Metropolis random walk, we can sample in* $B(0, R) \subset \mathbb{R}^d$ *according to a density* f' *with* $\sup_{S \subset \mathbb{R}^d} | \int_S f - \int_S f'| < \epsilon$ *in* K' *steps, where*

$$K' = 8 \cdot 10^4 R^2 d^2 \left(\ln(\theta/\epsilon) + (\gamma + 1) d \ln R\right).$$

PROOF. Theorem 5.1 gives an isoperimetric constant of $\frac{1}{R}$ for a random walk on $B(0, R)$. The Lipschitz continuity (Assumption B.3) implies that the acceptance function varies only by a constant factor over $B(x, \delta)$. Thus, choosing $\delta = \frac{1}{\sqrt{d}}$ implies a constant local conductance, since $\mathrm{vol}(B(x, \delta) \cap B(0, R)) \geq 0.4B(x, \delta)$, if $x \in B(0, R)$. To see this note that the intersection contains a cap of $B(x, \delta)$ of distance at most $\delta^2/(2R)$, by elementary geometry. Lemma 2.2 with $c = 1/(2R)$ now implies the volume of the cap is at least $1/2 - 1/(3R) > 0.4$, by Assumption B.2.

Applying these results, Theorem 5.2 implies a lower bound on the conductance Φ of the walk of $\frac{1}{200Rd}$. Theorem 5.3 then yields

$$\sup_{S \subset \mathbb{R}^d} \left| \int_S f - \int_S f' \right| < M(f_0, f) \left(1 - \frac{\Phi^2}{2}\right)^{K'}$$

We choose the uniform distribution on $B(\xi_0, \delta)$ as the initial density f_0, where ξ_0 is the point guaranteed by Assumption B.4. Assumption B.3 again implies that the acceptance function varies over $B(\xi_0, \delta)$ only by a factor $\theta\delta$. These, together with Assumption B.2, imply $M(f_0, f) = \theta R^{(\gamma+1)d}$, after some calculation. □

In Figure 2 we define the procedure *Estimate* based on the Markov chain described above to compute an approximation of the value of $Q(x)$.

Theorem 5.5 *With probability* $1-\rho$, *Procedure* Estimate *computes* $Q_K(x) \in [Q(x)-\epsilon, Q(x)+\epsilon]$ *in* $KK' = O\left(\frac{\gamma R^2 d^3 L^2}{\epsilon^2}\log\frac{\theta L}{\epsilon}\log R \log\frac{1}{\rho}\right)$ *steps.*

Proof. Let $Q_K(x) = \frac{1}{K}\sum_{i=1}^{K} v(Tx-\xi_i)$, with $\xi_1,\cdots,\xi_K$ independent samples generated by the Metropolis random walk. Thus, $\mathbf{E}[Q_K(x)] = \int v(Tx-\xi)f_{K'}(\xi)d\xi$, where $f_{K'}$ is the density produced by the Metropolis random walk when run with error parameter $\epsilon' = \epsilon/4L$. Since v is bounded by L by Assumption B.5, Hoeffding's inequality implies

$$\Pr\{|Q_K(x) - \mathbf{E}[Q_K(x)]| > \tfrac{1}{2}\epsilon\} \leq 2\exp\left(-\frac{\epsilon^2 K}{8L^2}\right),$$

since $K \geq 8L^2\ln(2/\rho)\epsilon^{-2}$. Theorem 5.4 yields

$$\begin{aligned}
|\mathbf{E}[Q_K(x)] - Q(x)| &= \left|\int v(Tx-\xi)f_{K'}(\xi)d\xi - \int v(Tx-\xi)f(\xi)d\xi\right| \\
&\leq L\int |f(\xi) - f_{K'}(\xi)|d\xi \\
&= 2L\int_{f\leq f_{K'}} (f_{K'}(\xi) - f(\xi))d\xi \\
&\leq 2L\epsilon' = \tfrac{1}{2}\epsilon,
\end{aligned}$$

Combining these, we see that $\Pr\{|Q_K(x) - Q(x)| > \epsilon\} \leq \rho$. □

Using the approximation algorithm for the objective function G and applying the ellipsoid algorithm in [9] to our problem, under our assumptions we would have a *fpras* for the stochastic recourse problem. The following theorem follows easily from taking $\rho = \zeta/N$ in Theorem 5.5, where N is the required number of steps of the ellipsoid algorithm.

Theorem 5.6 *Under Assumptions B.1-B.5, with probability at least* $1-\zeta$, *the ellipsoid algorithm, using procedure* Estimate *to approximately evaluate* G, *will solve the two-stage stochastic programming problem (8) to within additive error* ϵ, *in a number of arithmetic operations bounded polynomially in the input parameters,* $1/\epsilon$ *and* $\log 1/\zeta$. □

The ellipsoid algorithm is complicated to apply and can be very slow. Therefore we combine the method for approximate function evaluations with the method of Section 2 to obtain a simple randomized local improvement algorithm. Our earlier results will then imply that, with high probability, a solution is obtained that is close to the optimal.

6 Random directions for the recourse problem

In this section we will extend the Ball Walk approach to solve the two stage stochastic programming problem. Recall the formulation (8) in Section 5. We assume conditions B.1–B.5, which we required previously for the randomized approximate computation of $Q(x)$, and now we also assume that condition B.6 is satisfied.

It turns out that these conditions on the density function also imply smoothness of the objective function, required for the Ball Walk algorithm of Section 2, as we will show.

Lemma 6.1 *Suppose $x \in S$. Under Assumptions B.1-B.6, we have*

$$\|\nabla G(x)\| \leq \|p\| + \theta L\sqrt{d}, \quad \textit{and} \quad \|\nabla^2 G(x)\| \leq \theta^2 Ld.$$

As a result we may choose the smoothness parameters τ and ν of G (Assumptions A.2, A.3) to be $\|p\| + \theta L\sqrt{d}$ and $\theta^2 Ld$, respectively.

PROOF. Let w be a unit vector and $u = Tw/\|Tw\|$. Then

$$\begin{aligned}
Q'(w;x) &= \lim_{t\to 0}\left(\frac{\int v(Tx+tTw-\xi)f(\xi)}{t}d\xi - \frac{\int v(Tx-\xi)f(\xi)}{t}d\xi\right)\\
&= \lim_{t\to 0}\left(\frac{\int v(Tx-\xi)f(\xi+tTw)}{t}d\xi - \frac{\int v(Tx-\xi)f(\xi)}{t}d\xi\right)\\
&= \int v(Tx-\xi)\lim_{t\to 0}\frac{f(\xi+tTw)-f(\xi)}{t}d\xi\\
&= \|Tw\|\int v(Tx-\xi)f'(u;\xi)d\xi.
\end{aligned}$$

Analogously,

$$Q''(w;x) = \|Tw\|^2\int v(Tx-\xi)f''(u;\xi)d\xi.$$

Assumption B.1 ($\log f$ is concave) implies $f \geq 0$ and $f''f \leq (f')^2$ for the first and second derivatives in any direction at any point. Assumption B.3 (the Lipschitz continuity of $\log f$) implies $|(\ln f)'| \leq \theta$ for the first directional derivative, i.e. $|f'| \leq \theta f$, and hence also $f'' \leq \theta^2 f$. Thus, using also B.6,

$$|Q'(w;x)| \leq \theta\|Tw\|\int v(Tx-\xi)f(\xi)d\xi \leq \theta L\sqrt{d},$$

and

$$Q''(w;x) \leq \theta^2 \|Tw\|^2 \int v(Tw - \xi) f(\xi) d\xi \leq \theta^2 L d.$$

Therefore we have, for all x, w,

$$|G'(w;x)| \leq \|p\| + \theta \|T\| L \ \text{ and } \ G''(w;x) = Q''(w;x) \leq \theta^2 \|T\|^2 L.$$

The lemma follows. □

Thus, Assumptions A.2 and A.3 in Section 2 are satisfied. We know that G is concave, satisfying also Assumption A.1. Therefore, we need to be concerned now with smoothing the underlying feasible set $S = \{x \in \mathbb{R}^n | Ax \leq b\}$. Since S is a polyhedron, it is not smooth. Therefore we use the "roundedness" Assumption B.7 on S. This allows us to consider a slightly larger (non-polyhedral) set, which is indeed smooth, and apply the Ball Walk to this larger set.

Let $\kappa = R_{\text{out}}/R_{\text{in}}$, where R_{in} and R_{out} are the constants from Assumption B.7. We call κ the *rounding number* of the polytope. Note that κ resembles a "condition number": it is small if the polytope is "well-rounded" and large if it is not. Thus we will assume κ is not too large. In fact any polytope can be "rounded" to have $\kappa = O(\sqrt{n})$, but here this may be undesirable since it can adversely affect the parameters of G. From now on we assume that S satisfies the above assumption.

To facilitate the exposition, let $A^{(i)}$ denote the ith row of A, and suppose A is normalised so that $\|A^{(i)}\| = 1$ $(i = 1, \ldots, m)$. Then, for $x \in \mathbb{R}^n$, let

$$(A^{(i)}x - b_i)^+ = \begin{cases} A^{(i)}x - b_i & \text{if } A^{(i)}x > b_i, \\ 0 & \text{otherwise,} \end{cases}$$

be the distance from x to the halfspace $A^{(i)}y \leq b_i$. Consider the function $F(x) = \sum_{i=1}^m ((A^{(i)}x - b_i)^+)^2$, and, for a given $\mu > 0$, define the set $S_\mu = \{x \in \mathbb{R}^n | F(x) \leq \mu\}$. Note that $S \subseteq S_\mu$ and S_μ is convex. The following lemmas establish the required smoothness condition for S_μ.

Lemma 6.2 *For all* $x, z \in \mathbb{R}^n$, $\|z\| = 1$, $0 \leq F''(z;x) \leq 2m$.

PROOF. We have

$$\nabla F(x) = 2 \sum_{i \in I(x)} (A^{(i)}x - b_i) A^{(i)}, \tag{9}$$

where $I(x) = \{i : A^{(i)}x > b_i\}$. Hence

$$\nabla^2 F(x) = 2 \sum_{i \in I(x)} A^{(i)} {A^{(i)}}^T, \tag{10}$$

and so $F''(z;x) = 2\sum_{i\in I(x)}(A^{(i)}z)^2$. Clearly $F''(z;x) \geq 0$, and the upper bound follows from the Cauchy-Schwarz inequality and $|I(x)| \leq m$. □

In particular, it follows that F is a convex function.

Lemma 6.3 *Let S have rounding number κ and $0 < \mu \leq 1$. Then, for all $x \in S_\mu$,*

$$\kappa \|\nabla F(x)\| \geq \sqrt{F(x)}.$$

PROOF. Assume without loss that $x_0 = 0$ and $R_{\text{in}} = 1$, so $R_{\text{out}} = \kappa$. Also $b_i \geq 1$ $(i = 1, \ldots, m)$, by considering the points $A^{(i)} \in \partial B(0,1)$. Fix $x \in S_\mu$, let

$$\lambda = \min_{1 \leq i \leq m} \left\{ \frac{\|x\| b_i}{A^{(i)}x} : A^{(i)}x > 0 \right\},$$

and let ℓ be the minimizing i. Let $\tilde{x} = \lambda x/\|x\|$, so $\lambda = \|\tilde{x}\|$. Clearly $\lambda \leq \kappa$, since otherwise $A^{(i)}\tilde{x} \leq b_i$ $(i = 1, \ldots, m)$, but $\tilde{x} \notin B(0,\kappa)$. Now

$$\sqrt{\mu} \geq \sqrt{\mu}/b_\ell \geq (A^{(\ell)}x - b_\ell)/b_\ell = \|x\|/\lambda - 1 = \|x\|/\|\tilde{x}\| - 1.$$

Thus $\|x\| \leq (1+\sqrt{\mu})\|\tilde{x}\| \leq 2\kappa$. Thus, from (9),

$$x\nabla F(x) = 2 \sum_{i \in I(x)} (A^{(i)}x - b_i)A^{(i)}x \geq 2 \sum_{i \in I(x)} (A^{(i)}x - b_i) \geq 2\sqrt{F(x)},$$

since $A^{(i)}x > b_i \geq 1$ for $i \in I(x)$ and $F(x) = \sum_{i \in I(x)}(A^{(i)}x - b_i)^2$. Therefore, using Cauchy-Schwarz,

$$\|\nabla F(x)\| \geq 2\sqrt{F(x)}/\|x\| \geq \sqrt{F(x)}/\kappa,$$

and the result follows. □

Lemma 6.4 *Let $0 < \mu \leq 1$, and $\sigma = (4m\kappa/3)\sqrt{n/\mu}$. Then, for all $x \in S_\mu$,*

$$\textit{vol}(B(x,r) \cap S_\mu)/\textit{vol}(B(x,r)) \geq \tfrac{1}{2} - \sigma r.$$

PROOF. Assume without loss that x is on the boundary of S_μ. Using Taylor's theorem and Lemma 6.2, for any displacement z,

$$F(x+z) \leq F(x) + z\nabla F(x) + 2mz^2 = \mu + z\nabla F(x) + 2mz^2,$$

and the subgradient inequaility gives

$$F(x+z) \geq F(x) + z\nabla F(x) = \mu + z\nabla F(x).$$

Thus, if $F(x+z) = \mu$, $-2mz^2 \leq z\nabla F(x) \leq 0$. Let H denote the half-space $z\nabla F(x) \leq 0$, and ∂H its boundary. Then, using Lemma 6.3,

$$\operatorname{dist}(z, \partial H) = \frac{-z\nabla F(x)}{\|\nabla F(x)\|} \leq \frac{2mz^2}{\|\nabla F(x)\|} \leq \frac{2m\kappa z^2}{\sqrt{F(x)}} = \frac{2m\kappa z^2}{\sqrt{\mu}}.$$

Letting B denote $B(x, r)$, we see that $B \cap S_\mu$ contains a cap of B at distance at most $2m\kappa r^2/\sqrt{\mu}$ from x. Thus, by Lemma 2.2,

$$\frac{\operatorname{vol}(B \cap S_\mu)}{\operatorname{vol}(B)} \geq \frac{1}{2} - \frac{4m\kappa\sqrt{n}}{3\sqrt{\mu}} r.$$

□

Using the last two lemmas we can now apply the Ball Walk to optimize G over S_μ. But, we are of course interested in optimizing G over the set $S \subset S_\mu$. Thus, we will use Procedure Near (see Figure 6), which finds a point arbitrarily close to S after we have optimized over the larger set S_μ. The idea is to go repeatedly along the gradient of F, which is easy to compute, until we are exponentially close to S. We will show that this procedure yields a point that is not much worse than the point resulting from the Ball Walk Algorithm, for an appropriate choice of μ.

Theorem 6.1 *Let $x \in S_\mu$. For any $\beta < \mu$, Procedure Near finds $y \in S_\beta$ with $\|x - y\| \leq 2\sqrt{\mu/\beta}\ln(\mu/\beta)$. The running time is $O(\kappa^2 m^2 n \ln(\mu/\beta))$.*

PROOF. The claim on the running time is easy to verify. Without loss, we will prove the lemma for a point x on the boundary of S_μ, i.e. $F(x) = \mu$. We start the procedure at x with $F(x) = \mu$ and

$$\hat{x} = x - \frac{\sqrt{F(x)}}{2\kappa m} \frac{\nabla F(x)}{\|\nabla F(x)\|}.$$

```
Procedure Near($x, \beta$)
begin
        For $i = 1$ to $\lceil 4\kappa^2 m \ln(\mu/\beta) \rceil$ do;
                Calculate $F(x), \nabla F(x)$
                $\hat{x} = x - \frac{\sqrt{F(x)}}{2\kappa m} \frac{\nabla F(x)}{\|\nabla F(x)\|}$
                $x = \hat{x}$;
        endfor;
        return $x$;
end;
```

Figure 3: The Procedure Near

Using Lemma 6.2, the Taylor expansion yields

$$F(\hat{x}) \leq F(x) - \frac{\sqrt{F(x)}}{2\kappa m}\|\nabla F(x)\| + m\left(\frac{\sqrt{F(x)}}{2\kappa m}\right)^2.$$

Now, using Lemma 6.3, this implies

$$F(\hat{x}) \leq F(x) - \frac{(\sqrt{F(x)})^2}{2\kappa^2 m} + \frac{F(x)}{4\kappa^2 m} = F(x)\left(1 - \frac{1}{4\kappa^2 m}\right).$$

We repeat this process $k = 4\kappa^2 m \log(\mu/\beta)$ times to get a point y with

$$F(y) \leq F(x)\left(1 - \frac{1}{4\kappa^2 m}\right)^k \leq \mu\left(1 - \frac{1}{4\kappa^2 m}\right)^k \leq \beta.$$

Furthermore, letting y be the final point returned by Near,

$$\|y - x\| \leq k\frac{\sqrt{\mu}}{2\kappa m} \leq 2\kappa\sqrt{\mu}\,\ln(\mu/\beta).$$

Hence, using the subgradient inequality,

$$G(y) \geq G(x) - \tau\|y - x\| \geq G(x) - 2\tau\kappa\sqrt{\mu}\,\ln(\mu/\beta).$$

Thus, since $\mu \leq 1$, we will have $G(y) \geq G(x) - \epsilon$ provided

$$\mu \leq \left(\frac{\epsilon}{2\tau\kappa \ln(1/\beta)} \right)^2 . \tag{11}$$

Note that this is polynomial in the relevant parameters, in particular the number of bits of accuracy required. □

Collecting the last results and inserting the right parameters in the general time bounds of the Ball Walk we get our final result.

Theorem 6.2 *Under the Assumptions B.1–B.7, with probability at least* $(1 - \zeta)$*, an application of the* Ball Walk *combined with the procedures* Estimate *and* Near *will find* y *with*

$$G(y) \geq G(x^*) - \epsilon$$

and

$$Ay \leq b + \sqrt{\beta},$$

in time polynomial in the parameters of the problem, $1/\epsilon$, $\log(1/\beta)$ *and* $\log(1/\zeta)$.

PROOF. By using a small enough error probability at each step, the probability of making an error can be made at most ζ over any polynomial number of steps (cf. the discussion before Theorem 5.6). Choose $\mu = \left(\frac{\epsilon}{2\tau\kappa \ln(1/\beta)} \right)^2$. Lemmas 6.1, 6.4 and Theorems 5.5, 6.1 then imply the theorem. □

7 Postlude

We have described a simple randomized approximation scheme for convex optimisation problems, with two-stage stochastic programming problems as the main application. Whether the method that we propose here is a practically efficient method for solving two-stage stochastic programming problems remains to be seen, but in any case it may provide a starting point for a more practical method. For example, it is likely that function evaluations do not have to be so precise if we are still far away from the optimum. Indeed. stochastic programmers have proposed methods that work with more and more accurate function evaluations as their methods proceed. It remains a challenge to incorporate these and other ideas that have been developed in stochastic programming research into our algorithmic framework.

References

[1] Applegate, D., Kannan, R. Sampling and integration of near log-concave functions. *Proceedings of the 23th ACM Symposium on the Theory of Computing*, 1990, pp. 156–163,

[2] Beale, E.M. On minimizing a convex function subject to linear inequalities. *Journal of the Royal Statistical Society* **17**, 1955, pp. 173–184.

[3] Birge J., Louveaux F. *Introduction to stochastic programming*, Springer-Verlag, New York, 1997.

[4] Dantzig, G. Linear Programming under uncertainty. *Management Science* **1**, 1955, pp. 197–206.

[5] Dyer, M. and Frieze, A. Computing the volume of convex bodies: a case where randomness provably helps. *Probabilistic Combinatorics and its Applications*, Proceedings of AMS Symposia in Applied Mathematics Vol. 44, 1991, pp. 123-169.

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

[7] Dyer, M., Stougie, L. A note on the complexity of some stochastic optimization problems, preprint.

[8] Ermoliev Y., Wets R. (eds.) *Numerical techniques for stochastic optimization*, Springer-Verlag, Berlin, 1988.

[9] Grötschel, M., Lovasz, L., Schrijver, A. *Geometric algorithms and combinatorial optimization.* Springer Verlag, New York, 1988.

[10] Jerrum, M., Sinclair, A. The Markov chain Monte Carlo method: an approach to approximate counting and integration. In *Approximation algorithms for NP-hard problems*, D. S. Hochbaum (ed.), PWS Publishing, Boston, 1996, pp. 482–520.

[11] Kannan, R. Markov chains and polynomial time algorithms. *Proceedings of the 35th Symposium on Foundations of Computer Science*, 1994, pp. 656–671.

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

[13] Kannan, R., Nolte, A. Local search in smooth convex sets. *Proceedings of the 39th Symposium on Foundations of Computer Science*, 1998, pp. 218–226.

[14] Lovasz, L., Simonovits, M. Random walks in a convex body and an improved volume algorithm. *Random Structures and Algorithms* **4**, 1993, pp. 359–412.

[15] Nemirovski, A., Nesterov. Y. *Interior point polynomial methods in convex programming*, SIAM Press, 1994.

[16] Prekopa, A., *Stochastic programming*, Mathematics and its applications Vol. 324, Kluwer, Dordrecht, 1995.

[17] Tintner, G. Stochastic linear programming with applications to agricultural economics. In Antosiewicz, H.A. (ed.) *Proceedings of the Second Symposium on Linear Programming*, Washington, 1955, pp. 197–207.

[18] Wets, R. Stochastic programming: solution techniques and approximation. In Bachem, A., Groetschel M., Korte, B. (eds.) *Mathematical programming: the state of the art (Bonn, 1982)*, Springer, Berlin, 1983, pp. 566–603.

[19] Wets, R. Stochastic programming. In *Optimization*, Handbooks of Operations Research and Management Science Vol. 1, North-Holland, Amsterdam, 1989, pp. 573–629.