Math. Program., Ser. A (2014) 147:207–229
DOI 10.1007/s10107-013-0718-0

FULL LENGTH PAPER

A simple randomised algorithm for convex optimisation

Application to two-stage stochastic programming

M. Dyer · R. Kannan · L. Stougie

Received: 8 July 2010 / Accepted: 17 September 2013 / Published online: 13 October 2013

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

L. Stougie: Supported by the Tinbergen Institute and ABRI.

M. Dyer
Department of Computer Science, University of Leeds, Leeds, UK
e-mail: dyer@comp.leeds.ac.uk

R. Kannan
Microsoft Research Labs, Bangalore, India
e-mail: kannan@microsoft.com

L. Stougie (✉)
Division of Econometrics and Operations Research, Department of Economics and Business Adminstration, VU University, Amsterdam, The Netherlands
e-mail: lstougie@feweb.vu.nl

L. Stougie
CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
e-mail: stougie@cwi.nl

on the models. Moreover, we deal with smoothing the feasible set, which in two-stage stochastic programming is a polyhedron.

Keywords Convex optimisation · Stochastic programming · Randomised algorithms · Polynomial time randomised approximation scheme

Mathematics Subject Classification 90C25 · 90C15 · 68W20 · 68Q25

1 Introduction

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

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

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. [13]), can be solved in a polynomial number of basic computer operations [13,24]. 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 centered 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 Sect. 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 1950s [2,5,32]. They model optimisation under *uncertainty*. In Sect. 3 we give a brief introduction to these problems.

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 is #-P hard already for a subclass of rather simple two-stage stochastic linear programs [10]. Thus, simply assuming the existence of (even) an approximate function evaluation oracle for these evaluations conceals the intrinsic complexity of the problem: it is at least as hard as exact counting. 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.

On the other hand, although *exact* counting is usually hard, there are situations where *randomised approximate* counting is possible. See, for example [15]. Therefore, we might guess that a similar type of approximation would be possible here. We show in Sect. 4 that this intuition is justified. In particular, we design a subroutine for approximate evaluations of the objective function of two-stage stochastic programming problems. 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 and under a given uniform bound on the directional derivatives of the second stage random value function, 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* in the absolute sense. The uniform bound assumption is a strong assumption but seems hard to avoid. Other studies with the same aim [26,31] required bounds of some kind, as we will discuss in Sect. 6. We achieve our result by drawing on known techniques, but to our best knowledge it is a new result.

In Sect. 5 we combine the subroutine of Sect. 4 with the randomised convex optimisation algorithm developed in Sect. 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.

We conclude in Sect. 6 with a comparison of our paper with similar approaches that have appeared in the literature, notably the papers [31] and [26], giving the advantages and disadvantages of our approach. We postpone this comparison to the end for a better appreciation by the reader. Next to a historical note, we also review relevant literature which has appeared subsequent to the completion of the work described in this paper.

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 $S \subset \mathbb{R}^n$. 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 $x \in \mathbb{R}^n$ and a given error parameter $\epsilon_0 > 0$, returns a number $\widehat{G}(x)$ in the interval $[G(x) - \epsilon_0, G(x) + \epsilon_0]$. In the sequel we denote by x^* an optimal solution of the maximisation problem.

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 z uniformly at random in a ball of a certain radius r and centre x. 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 [22]. 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 $\mathrm{vol}(S)$, 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 $x \in \mathbb{R}^n$ 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.

Assumption A 1. G is concave and $S \subset \mathbb{R}^n$ is convex 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 $r \leq r_0$, $x \in S$,

$$\frac{\mathrm{vol}(B(x, r) \cap S)}{\mathrm{vol}(B(x, r))} \geq \frac{1}{2} - \sigma r;$$

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 $1 - \eta$. The choices of the various parameters that we make now, may seem magic at this point of the text, but their justification will become clear later. Because of the uncertainty in the estimation of G, we must choose $\epsilon \geq \frac{24\sqrt{n}\epsilon_0}{\tau r}$, so $\epsilon_0 \leq \frac{\epsilon\tau r}{24\sqrt{n}}$. The algorithm is described in Fig. 1, where we choose $r = \min\{r_0, \frac{D}{\sqrt{n}}, \frac{\epsilon\tau}{90\sigma\tau + 3\nu\sqrt{n}}\}$. Note that it might appear at first sight that Assumption A.4 could be satisfied by simply choosing a large value of σ. However, the value of r must then be very small, and so the running time of the algorithm will be large.

The improvement we get in one step of Ball Walk will be examined below in Theorem 2.1. But first we give some preliminary elementary estimates which will proveuseful.

Procedure Ball Walk
begin $x \leftarrow x_0$ and *counter* $\leftarrow 1$;
  while *counter* $\leq \frac{4200\sqrt{n}D\ln(1/\epsilon)\ln(1/\eta)}{\sigma r^2}$;
    generate $z \in B(x,r)$ uniformly at random;
      if $z \in S$ and $\widehat{G}(z) > \widehat{G}(x) + \frac{\epsilon\tau r}{12\sqrt{n}}$ **then** $x \leftarrow z$;
      set *counter* $\leftarrow$ *counter* $+ 1$;
  return x as the approximate solution.
end;

Fig. 1 The Ball Walk algorithm

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

Proof Since $\mathrm{vol}(B_n) = \pi^{n/2}/\Gamma(n/2+1)$,

$$\frac{\mathrm{vol}(B_{n-1})}{\mathrm{vol}(B_n)} = \frac{\Gamma(n/2+1)}{\sqrt{\pi}(\Gamma((n-1)/2+1)}.$$

Stirling's approximation to the Gamma function 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 $c/5 \leq vol(U)/vol(B(0,r)) \leq 2c/3$.*

Proof Since $c \leq 1$, elementary estimates give

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

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*

$$G(z) \;\geq\; G(x) + \frac{\epsilon\tau r}{6\sqrt{n}},$$

and the Ball Walk will accept this step.

Proof First shrink $B(x,r)$ to $B(x,r')$, with $r' = (1-\beta)r$, and $\beta = \frac{2}{n}$. We write B and B' shortly for $B(x,r)$ and $B(x,r')$, respectively. Notice that

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

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

$$T_1 = B' \cap S \cap \left\{y | (y-x)^T g(x) \geq -\frac{5c_1 r'}{\sqrt{n}}\right\} \cap \left\{y | (y-x)^T v \leq \frac{5c_2 r'}{\sqrt{n}}\right\},$$

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,

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

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 $\|z - x\| \leq r$.

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

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

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

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

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

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

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

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

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

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

$$\begin{aligned} G(y)-G(x) &= \nabla G(x)^T(y-x) + \frac{1}{2}G''(w;x')\|y-x\|^2 \\ &\ge -\frac{5c_1r'}{\sqrt{n}}\|\nabla G(x)\| - \frac{1}{2}vr'^2 \\ &\ge -\frac{5c_1r}{\sqrt{n}}\|\nabla G(x)\| - \frac{1}{2}vr^2 \\ &\ge -\frac{15\sigma r^2}{\sqrt{n}}\|\nabla G(x)\| - \frac{1}{2}vr^2, \end{aligned} \tag{6}$$

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

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

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

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

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

$$\frac{G(z)-G(x)}{G(x^*)-G(x)} \ge \frac{r}{6D\sqrt{n}} \tag{7}$$

Hence

$$G(z)-G(x) \ge \frac{r}{6D\sqrt{n}}\big(G(x^*)-G(x)\big) \ge \frac{\epsilon\tau r}{6\sqrt{n}},$$

as required. In this event,

$$\widehat{G}(z) - \widehat{G}(x) \;\geq\; G(z) - G(x) - 2\epsilon_0 \;\geq\; \frac{\epsilon\tau r}{6\sqrt{n}} - \frac{\epsilon\tau r}{12\sqrt{n}} \;=\; \frac{\epsilon\tau r}{12\sqrt{n}},$$

so the Ball Walk will accept. □

If the Ball Walk accepts, then we know that

$$G(z) - G(x) \;\geq\; \widehat{G}(z) - \widehat{G}(x) - 2\epsilon_0 \;\geq\; \frac{\epsilon\tau r}{12\sqrt{n}} - \frac{\epsilon\tau r}{12\sqrt{n}} \;=\; 0, \tag{8}$$

thus the Ball Walk algorithm never causes the value of $G(x)$ to decrease.

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

$$\frac{4{,}200\sqrt{n}D\ln(1/\epsilon)\ln(1/\eta)}{\sigma r^2}.$$

Proof From (7) in the proof of Theorem 2.1, with probability at least $\frac{120}{\sigma r}$ we have

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

and the Ball Walk accepts the step. Moreover, the concavity of G and Assumption A.2 imply that $G(x^*) - G(y) \leq \tau D$ for all feasible y, hence also for the starting point x_0. Let us call a *good step* one in which the improvement is as in (7), and note that no step results in a decrease in $G(x)$. Then, after k good steps, we obtain a point x^k with

$$G(x^*) - G(x^k) \;\leq\; \left(1 - \frac{r}{6D\sqrt{n}}\right)^k \tau D \leq\; \tau D \exp\left(-\frac{kr}{6D\sqrt{n}}\right) \;\leq\; \epsilon\tau D.$$

when

$$k \;\geq\; \frac{6\sqrt{n}D\log(1/\epsilon)}{r}.$$

Let $K = 700\,\mathrm{k}\ln(1/\eta)/(\sigma r)$. Then, using Chernoff's inequality, the probability that in K steps there are fewer than k good steps is at most

$$\exp\left(-\frac{1}{3}\left(\frac{5k\ln(1/\eta) - k}{5k\ln(1/\eta)}\right)^2 5k\ln(1/\eta)\right) \;=\; \exp\left(-\frac{k\big(5\ln(1/\eta) - 1\big)^2}{15\ln(1/\eta)}\right)$$
$$\leq\; \exp\big(-k\ln(1/\eta)\big) \;=\; \eta^k \;\leq\; \eta,$$

provided $\eta \leq e^{-1}$ and $k \geq 1$. □

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 1950s [2,5,32]. 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.

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

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 [27]. The paper of Kannan and Nolte [18] takes a similar approach to the probabilistic constraints problem that we take here for the model described next.

The other direction 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 [4,12,27].

A generic mathematical programming formulation for this problem is

$$\begin{array}{rl} \max & px + \mathbf{E}[\max\{\tilde{q}y \mid Wy \leq \tilde{T}x - \tilde{\xi},\ y \in \mathbb{R}^{n_1}\}] \\ \text{subject to} & Ax \leq b. \end{array} \tag{9}$$

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 [33]). 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. [4]).

It is well known that the objective function of (9) is *concave* (see [33,34]). 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 Sect. 5. 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 Sect. 5, we first devise a suitable function evaluation oracle in Sect. 4.

Our Ball Walk algorithm is certainly not among the first randomised approaches to solving two-stage stochastic programming problems. There is an extensive literature on sampling based methods, see a.o. [6,14,19,21,28]. In all these papers statistical convergence of the methods is analysed, but none considers complexity issues. The first one in that sense was the work by Kleiwegt, Shapiro and Honem-de-Mello, [20] in a version of the so-called sample average method, in which scenarios of the random parameters are sampled and then an approximate problem is solved, the so-called deterministic equivalent problem, with discrete distributions estimated by the samples. They give a bound on the number of samples needed to find a near-optimal solution that is polynomial in the dimension of the problem. The running time is however also a function of a parameter based on the distribution of the scenarios. It was in [29] and the aforementioned papers of [31] and [25] that running times became independent of distribution functions.

4 Computation of the objective function

As we pointed out in the Introduction, 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 (9) 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.

$$\max_{x \in S} \quad G(x) = px + Q(x), \tag{10}$$

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, even in the relative sense [15]. Therefore we assume the following conditions, borrowed from volume computation [1].

Assumption B 1. f is log-concave, i.e. $\log f$ is concave on its support supp f;
2. f has a negligible measure outside $B(0, R)$, i.e. for all $\varphi > 0$, there exists $R \geq 4d$ such that $\int_{\|\xi\| \geq R} f(\xi) d\xi \leq \varphi$;
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 $\xi, \xi' \in$ 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, where R is as chosen in B.2 above. This implies that we know a good starting point for the Markov chain, a so called "warm start";[1]
5. the directional derivative $v'(w; Tx - \xi)$ is uniformly bounded for all unit vectors $w \in \mathbb{R}^d$, i.e. there exists $\lambda > 0$ such that for all $\xi \in$ supp f and $x \in S$, $|v'(w; Tx - \xi)| \leq \lambda$;

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

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 $B(x_0, R_{\text{in}}) \subseteq S \subseteq B(x_0, R_{\text{out}})$. Let $\kappa = R_{\text{out}}/R_{\text{in}}$ be the *rounding number* or *aspect ratio* of the polytope S.

Conditions B.1–B.4 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 requires that the function $v(\cdot)$ does not vary too rapidly. 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, then by B.1 and B.5 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

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

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 [8,16] 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 4.1 *Dyer and Frieze* [7] *Let $\mathcal{S} \subset \mathbb{R}^d$ be a convex body with diameter D and f be a log-concave function defined on $\mathcal{S}$ with π the induced probability measure on $\mathcal{S}$. Let $S_1, S_2 \subset \mathcal{S}$, $\text{dist}(S_1, S_2) = \min_{x \in S_1, y \in S_2} \|x - y\|$. If $S_0 = \mathcal{S} \setminus (S_1 \cup S_2))$, then*

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

□

The quantity $2/D$ 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 4.2 [22] *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 4.3 [22] *Let f_0 be the initial density of a Metropolis random walk 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

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

□

Theorem 4.4 *Using the Metropolis random walk, we can sample in $B(0, R) \subset \mathbb{R}^d$ according to a density f' satisfying $\sup_{S \subset \mathbb{R}^d} |\int_S f - \int_S f'| < \epsilon$ in K' steps, where*

$$K' = 8 \cdot 10^4 R^2 d^3 e^{4\theta} \left(\ln \frac{1}{\epsilon} + (\gamma + 1) d \ln R \right).$$

Proof Theorem 4.1 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 factor $e^{-\theta\delta} \geq e^{-\theta}\delta$, for $\delta \leq 1$, over $B(x, \delta)$. Moreover, $\text{vol}(B(x, \delta) \cap B(0, R)) \geq 0.4 B(x, \delta)$, if $x \in B(0, R)$. To see this note that the intersection contains a cap of $B(x, \delta)$ at distance at most $\delta^2/(2R)$ from x, 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. Thus, choosing $\delta = \frac{1}{\sqrt{d}}$ implies a constant local conductance $\chi \geq \frac{0.4 e^{-\theta}}{\sqrt{d}}$.

Applying these results, Theorem 4.2 implies a lower bound on the conductance Φ of the walk of $\frac{1}{200 R d \sqrt{d} e^{2\theta}}$. Theorem 4.3 then yields

```
Procedure Estimate
begin
        Q ← 0;
        K ← ⌈8λ²R² ln(2/ρ) / ε²⌉;
        for ℓ ← 1, ..., K do;
                run the Metropolis walk for K′ steps,
                starting from ξ0;
                compute v(Tx − ξ) for the resulting sample ξ;
                Q ← Q + v(Tx − ξ);
        endfor;
        Q ← Q/K;
end;
```

Fig. 2 The approximation algorithm for $Q(x)$

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

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 $M(f_0,f)=e^{-\theta}R^{(\gamma+1)d}$, after some calculation. □

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

Theorem 4.5 *With probability* $1-\rho$, *procedure* Estimate *computes* $Q_K(x)\in[Q(x)-\epsilon, Q(x)+\epsilon]$ *in* $K=\left\lceil\frac{8L^2\ln(2/\rho)}{\epsilon^2}\right\rceil$ *samples.*

Proof Let $Q_K(x)=\frac{1}{K}\sum_{i=1}^K v(Tx-\xi_i)$, with $\xi_1,\ldots,\xi_K$ 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 $\epsilon'=\epsilon/4\lambda$. Since we assume that the directional derivative of v is uniformly bounded by λ by Assumption B.5, Hoeffding's inequality implies

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

since $K\ge 8\lambda^2R^2\ln(2/\rho)\epsilon^{-2}$. Theorem 4.4 yields

$$\begin{aligned}|\mathbf{E}[Q_K(x)]-Q(x)| &= \left|\int v(Tx-\xi)f_{K'}(\xi)d\xi-\int v(Tx-\xi)f(\xi)d\xi\right|\\ &\le \lambda R\int|f(\xi)-f_{K'}(\xi)|d\xi\end{aligned}$$

$$= 2\lambda R \int\limits_{f \le f_{K'}} (f_{K'}(\xi) - f(\xi))d\xi$$
$$\le 2\lambda R\epsilon' = \tfrac{1}{2}\epsilon,$$

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

Corollary 4.1 *In* $K \cdot K' = O\left(\frac{\gamma R^4 d^3 \lambda^2 e^{4\theta}}{\epsilon^2} \log \frac{1}{\epsilon} \log R \log \frac{1}{\rho}\right)$ *steps, procedure* Estimate *computes* $Q_K(x) \in [Q(x) - \epsilon, Q(x) + \epsilon]$. □

Using the approximation algorithm for the objective function G and applying the ellipsoid algorithm in [13] 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 4.5, where N is the required number of steps of the ellipsoid algorithm.

Theorem 4.6 *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* (10) *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 Sect. 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 optimal.

5 Random directions for two-stage stochastic programming

In this section we will extend the Ball Walk approach to solve the two stage stochastic programming problem. Recall the formulation (10) in Sect. 4. 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 Sect. 2, as we will show.

Lemma 5.1 *Suppose* $x \in S$*. Under Assumptions B.1–B.6, we have*

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

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

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

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

where we have used B.6 to give $\|Tw\| \le \sqrt{d}$. Also

$$\begin{aligned}
|Q''(w;x)| &= \|Tw\| \left| \lim_{t\to 0} \int \frac{v'(u; Tx+tTw-\xi)f(\xi) - v'(u; Tx-\xi)f(\xi)}{t} d\xi \right| \\
&= \|Tw\| \left| \lim_{t\to 0} \int \frac{v'(u; Tx-\xi)f(\xi+tTw) - v'(u; Tx-\xi)f(\xi)}{t} d\xi \right| \\
&= \|Tw\|^2 \left| \int v'(u; Tx-\xi) f'(u; \xi) d\xi \right| \\
&\le \|Tw\|^2 \int |v'(u; Tx-\xi)| \cdot |f'(u; \xi)| d\xi \\
&\le \theta \|Tw\|^2 \int |v'(u; Tx-\xi)| f(\xi) d\xi \\
&\le \lambda\theta \|Tw\|^2 \int f(\xi) d\xi \\
&\le \lambda\theta d,
\end{aligned}$$

where we have used that Assumption B.3 implies $|f'| \le \theta f$. The lemma follows. □

Thus, Assumptions A.2 and A.3 in Sect. 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 $S = \{x \in \mathbb{R}^n | Ax \le b\}$. 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.

Recall $\kappa = R_{\text{out}}/R_{\text{in}}$. 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 5.2 *For all $x, z \in \mathbb{R}^n$, $\|z\| = 1$, $0 \leq F''(z; x) \leq 2m$.*

Proof We have

$$\nabla F(x) = 2 \sum_{i \in I(x)} \left(A^{(i)}x - b_i\right) A^{(i)}, \tag{11}$$

where $I(x) = \{i : A^{(i)}x > b_i\}$. Since $F'(z; x) = \nabla F(x).z$,

$$F''(z; x) = 2 \sum_{i \in I(x)} \left(A^{(i)}z\right)^2. \tag{12}$$

Clearly $F''(z; x) \geq 0$, and the upper bound follows from $|I(x)| \leq m$ and the Cauchy-Schwarz inequality. □

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

Lemma 5.3 *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

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

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

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

Thus $\|x\| \leq (1 + \sqrt{\mu})\|\tilde{x}\| \leq 2\kappa$. Thus, from (11),

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

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 5.4 *Let $0 < \mu \leq 1$, and $\sigma = (2m\kappa/3)\sqrt{n/\mu}$. Then, for all $x \in S_\mu$,*

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

Proof Assume without loss that x is on the boundary of S_μ. Using Taylor's theorem and Lemma 5.2, for any displacement z,

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

and the subgradient inequality gives

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

Thus, if $F(x + z) = \mu$, $-mz^2 \leq \nabla F(x) \cdot z \leq 0$. Let H denote the half-space $\nabla F(x) \cdot z \leq 0$, and ∂H its boundary. Then, using Lemma 5.3,

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

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

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

□

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 Fig. 3), 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 5.1 *Let $x \in S_\mu$. Then, for any $\beta < \mu$, procedure Near finds a $y \in S_\beta$ such that $\|x - y\| \leq 2\sqrt{\mu/\beta}\ln(\mu/\beta)$. The running time is $O(\kappa^2 m^2 n\ \ln(\mu/\beta))$.*

```
Procedure Near(x, β)
begin
        For i = 1 to ⌈4κ²m ln(μ/β)⌉ do;
                Calculate F(x), ∇F(x)
                x̂ = x − (√F(x)/2κm) (∇F(x)/‖∇F(x)‖)
                x = x̂;
        endfor;
        return x;
end;
```

Fig. 3 The procedure near

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

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

Using Lemma 5.2, 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 5.3, 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

$$\mu \leq \left(\frac{\epsilon}{2\tau\kappa \ln(1/\beta)} \right)^2. \tag{13}$$

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 5.2 *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 4.6). Choose $\mu = \left(\frac{\epsilon}{2\tau\kappa \ln(1/\beta)} \right)^2$. Lemmas 5.1, 5.4 and Theorems 4.5, 5.1 then imply the theorem. □

6 Postlude

We have described a simple randomized approximation scheme for convex optimisation problems, with two-stage stochastic programming problems as the main application.

There have been several proposals to use random walks for convex optimisation since this paper was first written. The algorithm of Bertsimas and Vempala [3] uses random walks to generate hyperplanes which "bisect" the solution space. Repeating this sufficiently many times locates the solution to any desired precision. This is a very different approach from the algorithm given here. The computational complexity of Bertsimas and Vempala's algorithm is polynomial in terms of a standard measure of the size of the input, whereas we have worked with a different, less standard, measure of input size.

In [23], Lovász and Vempala adapt their "hit-and-run" algorithm for sampling from convex sets to optimise a logconcave function. The algorithm proceeds in phases, by modifying the objective function and sampling from the associated logconcave distribution, until an almost-optimal point is identified. The algorithm is shown to have polynomial time complexity in the standard model.

The paper of Kannan and Narayanan [17] uses random walks, similarly to the way they are used here, to optimize over polytopes. However, their steps are adaptive—they depend on the shape of the "Dikin ellipsoid" at the current point. These ellipsoids depend on knowing all the linear constraints and so the method only works for polytopes. It is an open question whether such an adaptive method exists for general convex sets, and hence to improve the algorithm presented here.

With respect to the application, it was as early as 1968 that Ermoliev and Shor proposed a random walk, based on stochastic gradients, to solve the two-stage stochastic programming model and analyzed global convergence [11]. As for the analysis of randomised algorithms for two-stage stochastic programming problems from a complexity theoretic point of view, some other papers close to our approach have appeared in the literature more recently [25,29,31]. We compare our paper to the ones by Shmoys and Swamy [31] and by Nesterov and Vial [25]. We refer to [25] for a comparison to the results on the sample average method by Shapiro and Nemirovski [29].

In fact, the results by Nesterov and Vial were obtained already before, [26], but as they notice, in the introduction of [25], "at that time worst-case complexity analysis in Stochastic Programming was not a common practice." Similarly, the work done on our paper dates back to the turn of the millenium [9]. "It was the attention received by the work of Shapiro and Nemirovski [29] (and Shmoys and Swamy [30]) that confirmed that the community became ready to accept these new notions".

Both [31] and [25] use stochastic subgradients. In [25] this is employed in a subgradient-descent approach for optimization of the convex objective function, whereas in [31] it is employed in an ellipsoid method. In this way expensive function evaluations can be avoided to find the near-optimal solution. Indeed in [31] the optimal solution value does not belong to the output of their method. In [25] only in the final solution the function evaluation is made, by sampling from the distribution of the stochastic parameters. Of course, this could be done as well in [31] but it would reduce the strength of their complexity result (that we discuss below), which in fact it does also in [25]. We use the simple Ball Walk algorithm, which is more stable than the ellipsoid algorithm, and outputs both solution and its value. At this stage of their development all three algorithms are only of academic interest.

A limitation of [31] is that their result applies only to problems in which the first and the second stage decisions are of exactly the same quality. The second stage cost coefficients are higher than the first stage ones. The algorithm of [25] and our algorithm apply to any setting, including hierarchical planning models, where the first stage concerns a longer term strategic decision and the second stage short term operational decisions.

All three approaches yield an fpras under more or less severe assumptions. The advantage of the limitation in [31] is that it allows a very elegant complexity result: it suffices to assume that a bound on the maximum ratio between the second and first stage cost coefficients is given as a fixed parameter; the running time of their algorithm is polynomial in this parameter. They even show that essentially this cannot be improved. In our case the fixed parameter is the uniform bound on directional derivatives of the value function. Similarly, in [25] a uniform bound on the Euclidean norm of the stochastic subgradients is the fixed parameter if, as in [31], only the solution is output. In order to compute also the value of the solution the running time

of the algorithm becomes also a polynomial function of a bound on the variation of the objective value over its feasible domain and over the probability space of the stochastic parameters.

All three approaches give an fpras (under the various above restrictions) under the "black box model" as it was called in [31]. Under this model the size of the input needed to describe the probability distributions of the random problem parameters is not taken into account in the running time. In [31] and in [25] an oracle is assumed to give realisations of the random parameters. We specify how to sample form the distributions. To avoid computational problems we restrict to specific classes of probability distributions.

As it holds for the other methods, it remains to be seen whether the method that we propose here is a practically efficient method for solving two-stage stochastic programming problems. 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.

Acknowledgments We thank David Shmoys for stimulation to update and resubmit our technical report [9] to a journal (the first submission in 2002 was apparently too early). We also thank anonymous reviewers for valuable comments.

References

1. Applegate, D., Kannan, R.: Sampling and integration of near log-concave functions. In: Proceedings of the 23th ACM Symposium on the Theory of Computing, pp. 156–163 (1990)
2. Beale, E.M.: On minimizing a convex function subject to linear inequalities. J. R. Stat. Soc. **17**, 173–184 (1955)
3. Bertsimas, D., Vempala, S.: Solving convex programs by random walks. JACM **51**, 540–556 (2004)
4. Birge, J., Louveaux, F.: Introduction to Stochastic Programming. Springer, New York (1997)
5. Dantzig, G.: Linear programming under uncertainty. Manag. Sci. **1**, 197–206 (1955)
6. Dupačovà, J., Wets, R.J.-B.: Asymptotic behavior of statistical estimators and of optimal solutions of stochastic optimization problems. Ann. Stat. **16**, 1517–1549 (1988)
7. Dyer, M., Frieze, A.: Computing the volume of convex bodies: a case where randomness provably helps. In: Probabilistic Combinatorics and its Applications. Proceedings of AMS Symposia in Applied Mathematics, vol. **44**, pp. 123–169 (1991)
8. Dyer, M., Frieze, A., Kannan, R.: A random polynomial time algorithm for approximating the volume of convex bodies. JACM **38**, 1–17 (1991)
9. Dyer, M., Kannan, R., Stougie, L.A.: Simple algorithm randomised algorithm for convex optimis; Application to two-stage stochastic programming. Technical Report SPOR-Report 2002–2005, Department of Mathematics and Computer Science, Eindhoven Tecnical Universtiy, Eindhoven (2002)
10. Dyer, M., Stougie, L.: A note on the complexity of some stochastic optimization problems. Math. Program. **106**, 423–432 (2006)
11. Ermoliev, Y., Shor, N.Z.: Method of random walk for the two-stage problem of stochastic programming and its generalization. Kibernetica **4**, 59–60 (1968)
12. Ermoliev, Y., Wets, R.J.-B. (eds.): Numerical Techniques for Stochastic Optimization. Springer, Berlin (1988)
13. Grötschel, M., Lovasz, L., Schrijver, A.: Geometric Algorithms and Combinatorial Optimization. Springer, New York (1988)
14. Higle, J.L., Sen, S.: Stochastic decomposition: an algorithm for two stage linear programs with recourse. Math. Oper. Res. **16**, 650–699 (1991)

15. Jerrum, M., Sinclair, A.: The Markov chain Monte Carlo method: an approach to approximate counting and integration. In: Hochbaum, D.S. (ed.) Approximation Algorithms for NP-Hard Problems, pp. 482–520. PWS Publishing, Boston (1996)
16. Kannan, R., Lovasz, L., Simonovits, M.: Random walks and an $O^*(n^5)$ volume algorithm for convex bodies. Random Struct. Algorithms **11**, 1–50 (1997)
17. Kannan, R., Narayanan, H.: Random walks on polytopes and an affine interior point method for linear programming. Math. Oper. Res. **37**, 1–20 (2012)
18. Kannan, R., Nolte, A.: Local search in smooth convex sets. In: Proceedings of the 39th Symposium on Foundations of Computer Science, pp. 218–226 (1998)
19. King, A.J., Rockafellar, R.T.: Asymptotic theory for solutions in statistical estimation and stochastic optimization. Math. Oper. Res. **18**, 148–162 (1993)
20. Kleywegt, A.J., Shapiro, A., Homem-De-Mello, T.: The sample average approximation method for stochastic discrete optimization. SIAM J. Optim. **12**, 479–502 (2001)
21. Mak, W.-K., Morton, D.P., Wood, R.K.: Monte-Carlo bounding techniques for determining solution quality in stochastic programs. Oper. Res. Lett. **24**, 47–56 (1999)
22. Lovasz, L., Simonovits, M.: Random walks in a convex body and an improved volume algorithm. Random Struct. Algorithms **4**, 359–412 (1993)
23. Lovasz, L., Vempala, S.: Fast algorithms for logconcave functions: sampling, rounding, integration and optimization. In: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science, pp. 57–68 (2006)
24. Nemirovski, A., Nesterov, Y.: Interior Point Polynomial Methods in Convex Programming. SIAM Press, Philadelphia, PA (1994)
25. Nesterov, Y., Vial, J.-P.: Confidence level solutions for stochastic programming. Automatica **6**, 1559–1568 (2008)
26. Nesterov, Y., Vial, J.-P.: Confidence level solutions for stochastic programming. Technical Report Core Discussion Papers. http://www.core.ucl.ac.be/services/psfiles/dp00/dp2000-13.pdf (2000)
27. Prekopa, A.: Stochastic Programming, Mathematics and its Applications, vol. 324. Kluwer, Dordrecht (1995)
28. Shapiro, A., Homem-de-Mello, T.A.: simulation-based approach to two-stage stochastic programming with recourse. Math. Program. **81**, 301–325 (1998)
29. Shapiro, A., Nemerovski, A.: On complexity of stochastic programming problems. In: Jeyakumar, V., Rubinov, A.M. (eds.) Continuous Optimization: Current Trends and Applications, pp. 111–144. Springer, Berlin (2005)
30. Shmoys, D.B., Swamy, C.: Stochastic optimization is (almost) as easy as deterministic optimization. In: Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science, pp. 228–237 (2004)
31. Shmoys, D.B., Swamy, C.: An approximation scheme for stochastic linear programming and its applications to stochastic integer programs. JACM **53**, 978–1012 (2006)
32. Tintner, G.: Stochastic linear programming with applications to agricultural economics. In: Antosiewicz, H.A. (ed.) Proceedings of the Second Symposium on Linear Programming, Washington, pp. 197–207 (1955)
33. 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), pp. 566–603. Springer, Berlin (1983)
34. Wets, R.: Stochastic programming. In: Optimization, Handbooks of Operations Research and Management Science, vol. 1, pp. 573–629. Amsterdam, North-Holland (1989)