Local Search in Smooth Convex Sets

Ravi Kannan
Yale University
Department of Computer Science
New Haven, CT 06510
kannan@cs.yale.edu

Andreas Nolte
Universität zu Köln
Institut für Informatik
50931 Köln
anolte@informatik.uni-koeln.de

Abstract

In this paper we analyse two very simple techniques to minimize a linear function over a convex set. The first is a deterministic algorithm based on gradient descent. The second is a randomized algorithm which makes a small local random change at every step. The second method can be used when the convex set is presented by just a membership oracle whereas the first requires something similar to a separation oracle.

We define a simple notation of smoothness of convex sets and show that both algorithms provide a near optimal solution for smooth convex sets in polynomial time. We describe several application examples from Linear and Stochastic Programming where the relevant sets are indeed smooth and thus our algorithms apply.

The main point of the paper is that such simple algorithms yield good running time bounds for natural problems.

1. Introduction

In this paper we analyse two geometric approaches to minimize a linear function cx over a convex set.

The first algorithm can be used when the set is given as a level set $M_\gamma = \{x : F(x) \leq \gamma\}$ of a convex function $F : \mathbf{R}^n \to \mathbf{R}$. A step of the algorithm is simply described : from the current feasible point x (inside M_γ), we compute $y = x - \lambda \nabla F(x) - \alpha c$ where λ, α are small positive reals and if y is in M_γ, we replace x by y, and repeat, otherwise we terminate. This of course requires the computation of the gradient of F.

The second algorithm applies when the convex set is only given by a membership oracle; in the second algorithm, we generate a new point y in a ball with center x and a certain radius r uniformly at random. If $y \in M_\gamma$ and y has a better objective function value, we go to y; otherwise, we do not. The algorithm terminates after a number of steps determined in advance.

We will develop a framework, namely the notation of smooth sets, in order to analyse these local search techniques and give a number of application examples from Linear and Stochastic Programming. The surprising fact is, that these simple methods work very well in a number of interesting cases.

The first example where our techniques apply is to linear programs

$$\max cx \quad \text{subject to} \quad Ax \geq b$$

which have the special property that (here and below we use the notation that $A^{(i)}$ denotes the i th row of $A \in \mathbf{R}^{m\times n}$)

$$A^{(j)}A^{(i)} \geq (-1+\kappa)|A^{(i)}||A^{(j)}|$$

for a constant $\kappa > 0$. This stipulates that there must be at least a constant angle between any two constraints of A, i.e., that there are no "sharp" corners in the polyhedron.

For such linear programs, we show that our algorithm finds a nearly feasible, nearly optimal point within additive error of ϵ in $O^*(\frac{m^2nD^2}{\epsilon})$ arithmetic operations. D denotes the diameter of the set $\{x|Ax \geq b\} \cap \{x|cx \leq cx_f\}$ and the O^* notation suppresses, as usual, logarithmic factors.

Positive or up-monotone Linear Programs (i.e. where $A_{ij} \geq 0$ for all i, j) is obviously a special case. In this special case, our algorithms can find an approximate solution z that satisfies the inequalities in $O^*\left(\frac{m^2n^2}{\epsilon^2}\log^2\left(\frac{D}{Opt}\right)\right)$ arithmetic operations. Opt is the value of the objective function of the optimal solution and D is here the value of an initially given feasible solution x_f. If we do not have an initially feasible solution, we can find one by setting $(x_f)_i = \max_{i,j}\frac{b_i}{a^i_j}$ - the algorithm is still polynomial.

Our second example is from a general class of problems called Constrained Probabilistic Programming [14] on which there is substantial literature. Here, as in the deterministic case we are given a cost vector c and a matrix A. But, here we have a random vector b with a probability distribution P and the task is to minimize cx over the feasible set $M_\gamma = \{x : P(Ax \geq b) \geq \gamma\}$ for a fixed $\gamma < 1$. Under mild and natural conditions on the density of b, one can see that M_γ is convex and we can prove a running time bound of $O\left(n^{2.5}m\min\left\{\frac{\min_{y\in M_\gamma} cy, cx_f}{\epsilon^2}\right\}\right)$ calls to the membership oracle of M_γ. We will see later that such a membership oracle is naturally available.

These bounds can be improved in the case of independent distribution of the components of b by a factor of $n^{0.5}m$. A similar approach is possible for another problem in Constrained Probabilistic Programming, namely the Component Commonality Problem, where also a fast running time of our method can be proved [10].

Finally, we will describe a method to make a general class of convex sets smooth, that are given by a membership oracle. This implies that our simple optimization algorithm can also be applied to certain, not necessarily smooth sets with good running time bounds. Furthermore, this method improves the local conductance [8] of the convex set considerably without changing the set much. This might have some implications for the running time of the random sampling algorithm, that approximates the volume of a convex set, but this is subject of further research.

2. Comparison

While of course gradient descent methods are very old, the analysis presented in this paper is new. This seems to be one of the first non-asymptotic results of gradient descent methods in a non-trivial geometric setting. Furthermore, we will develop a notation of smooth sets, where these simple methods have provable good running time bounds, that is very generally applicable. This is, additionally to the simplicity of this technique, the reason for the interest in this approach.

An alternative approach to solve the problem of minimizing a linear function over a convex set is the method of centers from NESTEROV and NEMIROVSKY [13]. It can be applied in the case where the convex set is given as a level set $M_\gamma = \{x : F(x) \leq \gamma\}$ of a given convex function $F : \mathbf{R}^n \to \mathbf{R}$. The method of centers is based on a logarithmic barrier function. It proceeds in stages and the Newton method is used to construct a series of feasible points in the interior of M_γ, that follow approximately a certain central path [15]. The barrier function of M_γ is tightly connected to F and has to be self-concordant (i.e. the third derivative is bounded above by a multiple of the second derivative), which implies fast convergence of the Newton method in a certain area, and Θ-self-limiting, which implies a certain lower bound on the size of the area of fast convergence of the Newton method. Given a suitable starting point x_f, this method can provide a feasible point x_K with $|cx_K - cx_{opt}| < \epsilon$ in $K = O(\sqrt{\Theta} \ln \frac{cx_K - cx_{opt}}{\epsilon})$ iterations. Each iteration involves the computation of F as well as the computation of the first and second derivatives and the solution of a linear system in $\mathbf{R}^n$ [15]. The parameter Θ is dependent on the barrier function and is at least k, if there exists an affine subspace in M_γ, that contains a vertex where precisely k linearly independent constraints are active.

The running time of our algorithm is not directly comparable to the running time of the method of centers. Our gradient descent algorithm does not involve the computation of the second derivative (i.e. the Hessian of F) and yields a solution which is approximately optimal in $\frac{D^2 \delta \log^2 D}{\epsilon}$ evaluations of F. D denotes the diameter of M_γ and δ is a parameter, that measures smoothness. Moreover, despite being much simpler, our technique can also be applied in the case where M_γ is given only by a membership oracle. Here, the method of centers cannot be applied directly, since the first and second derivatives have to be computed with a high precision.

In the case of Linear Programming it is of course possible to solve the optimization problem exactly in polynomial time with the ellipsoid method [6] or an interior point method [15]. However, the best known time bound for general linear programming is $O(m^{1.5}nL)$ arithmetic operations (Vaidya[16]). L is a parameter, that is bounded by the size of the problem [15]. Given the assumptions of the LP described above our algorithm outperforms Vaidya's algorithm, if ϵ is large (e.g. a constant) and the diameter is small.

In the special case of Positive Linear Programming our algorithm cannot improve the best known running time $O^*(\frac{mn}{\epsilon^4})$ of Luby and Nisan [11]. But their Lagrangian based algorithm is not as simple and has to cleverly exploit the structure of the problem.

In the case where our convex set is given by an oracle we could, as common in non-linear optimization, also use a gradient descent approach by estimating the gradient. Especially in stochastic optimization, where these problems arise frequently, this is the usual approach in practice to solve these problems ([17]). Because of the difficulty of getting an accurate estimate of the components of the gradient in a reasonable amount of time, apart from asymptotic results, only empirical running time results are known [17]. One could also use the Ellipsoid method to solve the problem, since membership and separation oracle are polynomially equivalent [6]. But the conversion needs a large number of steps and our approach here is more efficient. In the special case of the Component Commonality Problem our approach yields, despite being much simpler, an improved running time in comparison to the best known result so far from Kannan, Mount and Tayur [9].

3. Smoothness

In the following we will describe a suitable condition on our convex sets, that will imply a fast running time of our local improvement algorithms.

Definition 1 *Let $F : \mathbf{R}^n \to \mathbf{R}$ be a partially differentiable function, $\lambda, r, \gamma \in \mathbf{R}_+$. The set $M_\gamma = \{x \in \mathbf{R}^n : F(x) \leq \gamma\}$ is called (λ, r)-smooth, iff M_γ is convex and for every $x \in \{x \in \mathbf{R}^n : F(x) = \gamma\}$ the set*

$$\{y \in \mathbf{R}^n : |y - x|_2 \leq r\} \cap \{y \in \mathbf{R}^n : (x - y)\frac{\nabla F(x)}{|\nabla F(x)|_2} \geq \lambda\}$$

is contained in M_γ.

It is easy to see that this definition reflects the intuitive way to think about smoothness. If a set $M_\gamma = \{x \in \mathbf{R} | F(x) \geq \gamma\}$ is (λ, r)-smooth, then at every border point the cap of a ball with radius r

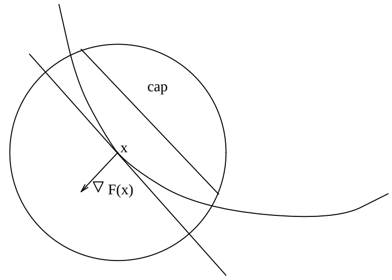

Figure 1. A smooth set

is contained in the set (see Figure 3). To be more precise let H be the hyperplane with normal vector $\nabla F(x)$ and $x - \lambda \nabla F(x) \in H$. Then the part of the ball that is, compared to x, on the other side of the hyperplane should be fully contained in M_γ. This prevents the border of M_γ having sharp corners, if we assume $\lambda < r$. Obviously M_γ is just a halfspace, if $\lambda = 0$ and $r > 0$ and the definition is meaningless if $\lambda \geq r$. This implies that the relation of λ and r is a measure of smoothness. The next lemma is a useful criterion for smoothness.

Lemma 1 *Suppose $F : \mathbf{R}^n \to \mathbf{R}$ is twice differentiable, $\lambda, r \in \mathbf{R}^+$ and x satisfies $F(x) = \gamma$. Let $y \in B(x, r)$ (ball with center x and radius r according to the L_2 norm) and let $G(\zeta) = F(x + \zeta(y - x)) : [0, 1] \to \mathbf{R}$. If*

$$2\lambda |\nabla F(x)|_2 \geq G''(\zeta) \tag{1}$$

for every $\zeta \in [0, 1]$ and every such pair x, y, then M_γ is (λ, r)-smooth.

Proof: Easy application of the second order Taylor theorem. □

Before we describe our approximation algorithm, we want to make an observation, that is used in the analysis several times. We prove a lower bound on the angle between the tangent hyperplane and the isohyperplane (same values of the objective function as x) at an arbitrary point x at the boundary of M_γ, that is still quite far away from the optimum in terms of the objective function. Let x be at the boundary of M_γ (i.e. $F(x) = \gamma$) and $\Delta = cx - \min_{y \in M_\gamma} cy$. We assume c to be a unit vector.

Lemma 2 *Suppose H_1 is the tangent hyperplane of M_γ at x with normal vector $t = \frac{-\nabla F(x)}{|\nabla F(x)|}$ and H_2 is the hyperplane with normal vector c and $x \in H_2$. Let D be the diameter of the set $M_\gamma \cap \{y \in \mathbf{R}^n | cy \leq cx\}$. Thus*

$$\vec{t} \cdot \vec{c} \leq \cos\left(\frac{\Delta}{D}\right).$$

Proof: Elementary geometry. □

Roughly speaking, Lemma 2 implies, that, if we are quite far away from the optimum in terms of the objective function, but not too far away in terms of L_2 distance, we can expect a quite large angle between the tangent hyperplane and the isohyperplane.

4. The gradient descent method

Here, we assume that F itself and the gradient ∇F can be computed efficiently. To avoid messy notation we assume further that both evaluations take the same amount of time which is at least $\Omega(n)$.

Procedure Decrease(x, Δ)
begin
$\alpha = \Delta / D$;
$z' = x - \frac{\alpha^2}{6\delta} \frac{\nabla F(x)}{|\nabla F(x)|}$;
Find next border point z in direction
$-c + \left(\frac{\nabla F(x)}{|\nabla F(x)|} c\right) \frac{\nabla F(x)}{|\nabla F(x)|}$ (line search);
Return z;
end;

Figure 2. The procedure Decrease

We will give a running time bound in terms of the evaluations of F. The following condition is our condition on the smoothness of M_γ in order to optimize over M_γ efficiently:

$$\exists \delta \quad \forall r \leq 1 \quad \forall \lambda \geq \delta r^2 \quad M_\gamma \text{ is } (\lambda, r)\text{-smooth.} \tag{2}$$

It says that if we choose λ to be a certain fixed fraction of r (i.e. we want to have a certain cap inside the convex set), then we have to choose the radius accordingly. Therefore δ is a direct measure of the curvature. The more our level curve is bent, the smaller the chosen radius and the larger δ have to be and vice versa. The following theorem summarizes the running time analysis of our gradient descent algorithm.

Theorem 1 *Let $x_f \in M_\gamma$ and D be the L_2-diameter of the set $M_\gamma \cap \{y \in \mathbf{R}^n | cy \leq cx_f\}$. The gradient descent algorithm will find a $z \in M_\gamma$ with $cz \leq \min_{y \in M_\gamma} cy + \epsilon$ in $O(\frac{D^2 \delta \log^2 D}{\epsilon})$ evaluations of F.*

Let $\min_{y \in M_\gamma} cy \leq C$ (this is only a technical assumption enabling us to do a binary search for the correct value of the minimum later). We assume $x \in M_\gamma$ to be a border point of M_γ with $\Delta = cx - C > 0$ (we have an initially given feasible point $x_f \in M_\gamma$ and finding a border point is easy using line search). In the following we will describe a single step of our algorithm and analyse the improvement. We try to optimize the objective function over that part of the ball with radius $r \in \mathbf{R}^+$, that is, due to the smoothness condition, guaranteed inside of M_γ.

Proposition 1 *Suppose x is a point at the border of M_γ. Let $\alpha = \Delta / D$ and $cx = C + \Delta$. Procedure Decrease (see Figure 4) finds a $z \in M_\gamma$ with $cz \leq cx - \frac{\alpha^2}{12\delta}$ in $O(\log D)$ evaluations of F.*

Proof: (Sketch) By defining $r = \frac{\alpha}{3\delta}$ and $\lambda = \alpha r / 2$ we have (λ, r)-smoothness of M_γ. Using this and the fact about the angle in Lemma 2 it is easy to prove the Lemma. □

By repeated application of Proposition 1 and by doing a binary search on C to find a C near the minimal value $\min_{y \in M_\gamma} cy$ the main result of this section, Theorem 1, follows.

4.1. Application example: Linear Programming

In the following we will describe an application example that does not fit in this framework in the obvious way. We will give an approximation algorithm for certain types of linear programs, which are obviously not smooth in general. But, it turns out that we can consider a slightly bigger, but smooth set and apply our method to this set. In general we are interested in solving the following problem

$$\min cx \quad \text{s.t.} \quad Ax \geq b,$$

where $c \in \mathbf{R}^n$ is a unit cost vector, A is a $m \times n$ matrix with normalized rows $|A^{(i)}| = 1$ and $b \in \mathbf{R}^m$. In order to apply the gradient descent method we have to put some conditions on the instances. We assume

- $A^{(j)}A^{(i)} \geq -1+\kappa$ for a constant $\kappa > 0$ (i.e. there must be a constant angle between the hyperplanes)
- we are given an initially feasible point x_f (i.e. $Ax_f \geq b$)

We will denote the set of feasible points $\{x|Ax \geq b\}$ by P. The basic idea to apply the gradient descent algorithm is, that we have defined a set, that contains P, however, that is not much larger than P, but smooth without sharp corners. We optimize over this larger set using the gradient descent method described above. In order to state our result we define for $x \in \mathbf{R}^n$ $(A^{(i)}x - b_i)^- = -A^{(i)}x + b_i$, if $A^{(i)}x < b_i$ and 0 elsewhere as the distance of x to the hyperplane $A^{(i)}y = b_i$, if x is not on the 'right' side. We consider $F(x) = \sum_{i=1}^m ((A^{(i)}x - b_i)^-)^2$ and define $M_\gamma = \{x \in \mathbf{R}^n | F(x) \leq \gamma\}$ for a given $\gamma > 0$ as the extension of P, which is still convex. Given $\epsilon, \gamma > 0$ our gradient descent algorithm will find a point z s.t. $cz \leq \min_{Ax \geq b} cx + \epsilon$ and $F(x) \leq \gamma$ i.e. the objective function value of z is within a small range of the actual solution and the sum of the violations of the inequalities squared is bounded by γ. The main object of our efforts in this section is to prove the following Proposition.

Proposition 2 *Suppose $x_f \in P$ is the initially feasible solution and D is the (finite) diameter of the set $\{x|Ax \geq b\} \cap \{x|cx \leq cx_f\}$.Let $\epsilon, \gamma > 0$. The gradient descent algorithm described in the last section can find a point z with $\vec{c}z \leq \min_{y \in P} \vec{c}y + \epsilon$ and $\sum_j ((A^{(j)}z - b)^+)^2 \leq \gamma$ in $O(\frac{m^2 n D^2 \log^2 D}{\epsilon\sqrt{\gamma}})$ arithmetic operations.*

As a first step we show, that M_γ is smooth.

Lemma 3 $\forall r \leq 1 \quad \forall \lambda \geq \frac{m}{2\sqrt{\gamma}} r^2 \quad$ *M_γ is (λ, r)-smooth.*

Proof: Let $r \leq 1, \lambda \geq \frac{m}{2\sqrt{\gamma}} r^2$ be given and assume $F(x) = \gamma$ (i.e. x is at the border of M_γ). An easy calculation yields $\nabla F(x) = -\left(\sum_{i=1}^m 2(A^{(i)}x - b_i)^+ a_k^i\right)_{k=1,\ldots,n}$. The assumptions on A allow us us to get a lower bound on the length of the gradient: $|\nabla F(x)|_2^2 \geq 4\kappa\gamma$. Let $y \in B(x, r)$ and $G(\lambda) = F(x + \lambda(y - x)) : [0,1] \to \mathbf{R}$. We get $G''(\zeta) \leq \sum_{i=1}^m 2(A^{(i)}(y - x))^2 \leq 2m|y - x|_2^2 \leq 2mr^2$. Using Lemma 1 the result follows. $\square$

With Theorem 1 the proof of Proposition 2 is complete.

4.2. Positive Linear Programming

In this subsection we are concerned with the special case that each entry of the matrix A is positive and P is contained in the positive orthand. This is a special case of the last section since the scalar product of each row is even bounded below by 0. It is easy to see, that we may assume, that the cost vector c has all strictly positive components. After rescaling, if necessary, we may also assume, that $c = \frac{\vec{1}}{\sqrt{n}}$, the normalized vector of 1's. In order to get hold on the diameter D of the set $\{x|Ax \geq b\} \cap \{x|cx \leq cx_f\}$, we consider the initially given feasible solution $x_f \in P$. By defining $D = \sqrt{2}(\vec{1}x_f)$ it is easy to see that every solution $y \in P$ with $\vec{1}y \leq \vec{1}x_f$ must satisfy $|y - x_f|_2 \leq D$, since P is contained in the positive orthand. The following theorem is the main result of this section.

Theorem 2 *Suppose x_f is an initially feasible point in P and $D = \sqrt{2}(\vec{1}x_f)$. The gradient descent algorithm can find a $z \in P$ with $\vec{c}z \leq (1+\epsilon)\min_{y \in P} \vec{c}y$ in $O(\frac{m^2 n^2}{\epsilon^2} \log^2 \frac{D}{Opt} \log n \log \frac{1}{\epsilon})$ arithmetic operations.*

As a first step we are concerned with a proper choice of γ in order to round a solution $x \in M_\gamma$ to a feasible solution without losing much in terms of the objective function. After the termination of our process analogous to the last section we get an $x \in M_\gamma$ with $\frac{\vec{1}x}{\sqrt{n}} \leq \min_{y\in P} \frac{\vec{1}y}{\sqrt{n}} + \epsilon$. We have to round it to $x' \in P$ with $\frac{\vec{1}x}{\sqrt{n}} \leq \min_{y\in P} \frac{\vec{1}y}{\sqrt{n}} + 2\epsilon$. Before we round we will carry out a preprocessing step to get closer to the set of feasible points P without making our objective function much worse. The idea is to calculate repeatedly the gradient of F at the current point and go in every step a little bit along the negative gradient towards P.

Proposition 3 *Assume $\gamma = \frac{\epsilon^2}{16\log^2 n}$ and let $x \in M_\gamma$ with $F(x) \leq \gamma$. We can round x to a $z \in P$ with $\frac{\vec{1}z}{\sqrt{n}} \leq \frac{\vec{1}x}{\sqrt{n}} + \epsilon$ in $O(m^2 n \log n)$ arithmetic operations.*

Proof: (Sketch) Using the idea of repeatedly calculating the gradient we can find, starting from an $x \in M_\gamma$ with $F(x) = \gamma$, a $y \in M_\gamma$ with $|x-y|_2 = 3\sqrt{\gamma}\log n$ and $F(y) \leq \gamma/4n$ in $O(m^2 n \log n)$ arithmetic steps. Using this and applying the facts, that P is up-monotone (i.e. $x \in P$ and $y \geq x$ componentwise implies $y \in P$) and $F(x) \geq \inf_{y\in P} |x-y|_2^2$ the result follows. □

Finally we use a scaling approach to prove Theorem 2, i.e. we apply our gradient descent methods repeatedly to a rescaled polyhedron. The analysis is technical and is omitted from this abstract.

5. Membership Oracle

In this section we do *not* assume that we can compute F and the gradient of F efficiently. We just assume that we are given a membership oracle for the set M_γ that can decide whether a given point $x \in \mathbf{R}^n$ belongs to M_γ or not. Our aim is, as in the last section, to minimize a linear function over M_γ, while we assume the existence of an initially given point $x_f \in M_\gamma$. Instead of computing the gradient at a feasible point x and going perpendicular to it in the direction of the objective function, we just generate a point in a ball of a certain radius r and center x uniformly at random. If the point is feasible (i.e. in M_γ) and has a better objective function value, we move in the direction of this point (binary search for the next border point) and iterate. Otherwise we repeat the random generation. This means we are just looking for a random direction, in which the objective function might improve. The algorithm is described in detail in Figure 5. The following condition is again our condition on M_γ in order to optimize efficiently.

$$\exists\delta \quad \forall r \leq 1 \quad \forall\lambda \geq \delta r^2 \quad M_\gamma \text{ is } (\lambda, r)\text{-smooth} \tag{3}$$

In the following we are concerned with the proof of the main theorem.

Theorem 3 *Suppose $x_f \in M_\gamma$ and D is the diameter of the set $M_\gamma \cap \{y \in \mathbf{R}^n | cy \leq cx_f\}$. The random greedy algorithm described in Figure 3 will find a $z \in M_\gamma$ with $cz \leq \min_{y\in M_\gamma} cy + \epsilon$ in $O^*\left(\frac{D^3\delta n}{\epsilon^2}\right)$ calls to the membership oracle.*

The simple greedy strategy was outlined above. We will proceed with the analysis of a single step. The aim is that we will indeed get an improvement with a certain probability depending on the value of the objective function of the current point. We assume that our current point x is at the border of M_γ. First, we try to get a lower bound on the probability to get a feasible point in M_γ by randomly picking a point in a ball $B(x,r)$ with center x and radius r. Let $\Delta = \min_{y\in M_\gamma} cy - cx_f > 0$.

Lemma 4 *Suppose $\alpha = \frac{\Delta}{D}$, $r = \frac{\alpha}{50\sqrt{n\delta}}$ and $B_{nf} = \{y \in \mathbf{R}^n || y - x| \leq r \quad \text{and} \quad F(y) > F(x) = \gamma\}$. This implies vol $B_{nf}(x,r) \leq \frac{1}{4}\frac{\alpha}{2\pi}$vol $B(x,r)$.*

Procedure Randomgreedy (x_f, ϵ, membership oracle for M_γ, diameter D)
begin
    $x = x_f$;
    $U = \vec{c}x$;
    $L = U - D$;
    Repeat;
        Repeat $\frac{50D^3\delta n}{(U-L)^2}$ times;
            $r = \frac{U-L}{50D\delta n^{0.5}}$;
            Repeat(basic step)
                Generate a random $x' \in B(x,r)$;
                check whether $x' \in M_\gamma$;
                check whether $cx' < cx$;
            Until $x \in M_\gamma$ and $cx' < cx$;
            Find border point x in direction x';
        End(repeat);
        If $\vec{c}x \leq 2/3U + 1/3L$
            then $U = 2/3U + 1/3L$;
        Else $L = 2/3L + 1/3U$
    Until $U - L \leq \epsilon$;
end;

Figure 3. The algorithm random greedy

Proof: Let $\lambda = \frac{\alpha r}{50\sqrt{n}}$. The tangent hyperplane cuts the generated ball in half. According to the definition of smoothness and (3) the volume of the set $B_{nf}(x,r)$ of points y that are on the right side of the hyperplane and $F(y) < F(x)$ (i.e. $y \notin P$) is bounded above by $\mathrm{vol}_{n-1}B(x,r)\lambda = \frac{r^{n-1}\tilde{c}^{n-1}\alpha r}{(n-1)^{\frac{n-1}{2}}50\sqrt{n}} \leq \frac{1}{4}\frac{\alpha}{2\pi}\mathrm{vol}_n B(x,r)$, where $\tilde{c} \leq 2\pi$ is a certain, ball specific constant. $\square$

We consider now the tangent hyperplane at x and the isohyperplane $\{y|cy = cx\}$. According to Proposition 2 we know that there is at least an angle of $\alpha \geq \frac{\Delta}{D}$ between the corresponding normal vectors. We need the following obvious lemma.

Lemma 5 *Let $B_b = \{y \in \mathbf{R}^n | \nabla F(x)y \geq \nabla F(x)x \quad \text{and} \quad cy \leq cx\} \cap B(x,r)$ be the set of better points in the ball, that are on the right side of the tangent hyperplane. This implies vol $B_b \geq \frac{\alpha}{2\pi}$vol $B(x,r)$.*

In the following we want to estimate how much progress we could expect in each step.

Lemma 6 *Suppose $B_p = \left\{y \in \mathbf{R}^n | cx \geq cy \geq cx - \Theta\left(\frac{\alpha r}{\sqrt{n}}\right)\right\}$. Then we get vol $B_p \leq \frac{1}{4}\frac{\alpha}{2\pi}$vol $B(x,r)$.*

Proof: Analogous to the proof of Lemma 4. $\square$

As a corollary we get a lower bound on the expected step size.

Corollary 1 *Suppose x is a point at the border of M_γ and $\Delta = cx - \min_{y \in M_\gamma} cy$. Let $\alpha = \frac{\Delta}{D}$ and $r = \frac{\Delta}{50D\delta\sqrt{n}}$. The process of choosing a random point in $B(x,r)$ hits a point $y \in M_\gamma$ with $\vec{c}y \leq \vec{c}x - \Theta\left(\frac{\alpha r}{\sqrt{n}}\right)$ with probability of at least $\Omega(\alpha) = \Omega(\frac{\Delta}{D})$.*

Thus, we have an analysis of one basic step of the algorithm. We will perform the algorithm in stages, taking care of the dependence of the radius on α. The proof is technical and omitted from this abstract. In the following subsection we will give an application example from stochastic programming.

5.1. Application Example: Probabilistic Constrained Programming

The following problem is a standard problem in stochastic optimization [17]. We are given a $m \times n$ matrix A with real entries and a random vector $b \in R^m$ with density h. Furthermore, we are given a cost vector $c \in R^n$ and the problem is to minimize cx under the condition

$$P(Ax \geq b) \geq \gamma$$

for a certain fixed $\gamma < 1$. This natural extension of Linear Programming was already considered by DANTZIG [3] and there exists substantial literature on this topic (see [14] for an overview). The general assumption is that membership queries can be answered efficiently. This means that we are given a subroutine which can decide for a given $x \in \mathbf{R}^n$ whether $P(Ax \geq b) \geq \gamma$. In practical settings we could use a sampling approach, if a set of sample vectors b according to the density function is given. Or we could use a Markov chain approach [1], if the density is known. But we do not go into the details here. In order to use our local search approach we have to put certain conditions on the instances to get smooth sets, since obviously a sharp concentrated density would cause our set to be non-smooth. We assume

- $A \geq 0$ and $h \geq 0$
- there is a constant bound for each component of h (i.e. $h_i(x) > 0$, iff $0 \leq x_j \leq \kappa$ for each j and a constant $\kappa \in \mathbf{R}^+$
- h is log-cancave and $\exists c_1, c_2 \in \mathbf{R}^+ \quad \forall x \in \mathbf{R}^m \quad \forall y \in B(x, c_1) \quad \frac{h(x)}{h(y)} \leq 1 + c_2|x - y|$ (this is a Lipschitz condition for $\log h$, which guarantees smoothness of h)
- h is also a density function of every (n-1)-dimensional subspace

But these conditions do not put severe restrictions on the density function, since (with minor technical changes) all the most important distributions like the exponential and the normal distributions meet these requirements. As usual we also assume the existence of a given initial feasible point x_f (i.e. $P(Ax_f \geq b) \geq \gamma$). Due to the unboundedness of M_γ in every positive direction we might assume that the cost vector c has all strictly positive components. After rescaling, if necessary, we may also assume that $c = \vec{1}/\sqrt{n}$. To simplify the notation we may further assume that the row vectors of the matrix A are unit vectors. We define $M_\gamma = \{x \in \mathbf{R}^n | P(Ax \geq b) \geq \gamma\}$. One can show that $F(x) = P(Ax \geq b)$ is a log-concave function [14], so that M_γ is convex and fits into our framework. The following theorem summarizes the running time of the random greedy algorithm.

Theorem 4 *The random greedy algorithm needs at most* $O\left(n^{2.5} m \min\left\{\frac{\min_{y \in M_\gamma} cy}{\epsilon^2}, cx_f\right\}\right)$ *calls of the membership oracle of* M_γ *to get a feasible approximate solution* z *with* $cz \leq (1+\epsilon)\min_{y \in M_\gamma} cy$ *and* $P(Ax \geq b) \geq \gamma$.

First, we prove a bound on the smoothness of M_γ.

Lemma 7 $\exists c \quad \forall r \leq 1 \quad \forall \lambda \geq cn^{1.5}mr^2 \quad M_\gamma$ *is* (λ, r)*-smooth.*

The proof of this proposition requires a sequence of lemmata. Before we start we state a theorem [4], called the isoperimetric inequality, which is useful in the analysis.

Theorem 5 *Let $J \subset \mathbf{R}^n$ be a convex body and h be a log-concave function defined on J and μ the induced measure. Let $S_1, S_2 \subset J$, $t \leq dist(S_1, S_2) = \max_{x \in S_1, y \in S_2} |x-y|_2$ and $d \geq diam\ (J) = \max_{x,y \in J} |x-y|_2$. If $B = J \setminus (S_1 \cup S_2)$, then*

$$\min(\mu(S_1), \mu(S_2)) \leq \frac{1}{2}(d/t)\mu(B).$$

We are going to use the isoperimetric inequality in [4] to get a lower bound on the length of the gradient, since this gradient is related to a $(n-1)$-dimensional surface area inside a convex body.

Lemma 8 $|\nabla F(x)|_2 \geq \frac{c}{m}$ *for a certain constant $c \in \mathbf{R}^+$.*

Proof: Let $F_j = \{y \in \mathbf{R}^m | y_j = (Ax)_j \quad \text{and} \quad y \leq Ax\}$. and $i, 1 \leq i \leq n$ be fixed. We get with $A_c^{(i)} =$ ith-column of A $\frac{\partial F(x)}{\partial x_i} = \sum_j A_{ji} \int_{F_j} h$. Therefore $|\nabla F(x)|_2^2 \geq |(\int_{F_j} h)_j|_2^2$. Let Σ be the $(n-1)$-dimensional surface area of $S_x = \{y | y \leq Ax\}$ (inside the positive orthand). $|(\int_{F_j} h)_j|_1$ is obviously the integral of h over Σ. $\int_\Sigma h \geq \frac{2}{diam(B)} \min\left\{\int_{M_\gamma} h, \int_{B \setminus M_\gamma} h\right\}$ with $B = \{y \in \mathbf{R}^m | h(y) > 0\}$ is a direct implication of the isoperimetric inequality 5. Because there is a constant bound for each component, we obtain $diam(B) \leq \kappa\sqrt{m}$ for the constant $\kappa \in R^+$ (see Assumptions). Therefore, observing $\int_{S_x} h = \gamma < 1$, it follows $|\nabla F(x)|_1 \geq \frac{c}{\sqrt{m}}$ for a constant $c \in \mathbf{R}^+$. Applying the standard relationship between the L_1 and the L_2 norm we get the result of the lemma. □

Now, we prove an upper bound on the norm of the Hessian of F. F is almost everywhere twice differentiable and we want to consider $|HF(x)|_2 = \max_{|z|,|y|=1} |z^T HF(x) y|$ with $x \in M_\gamma$.

Lemma 9 $|HF(x)| = O(n^{1.5})$.

Proof: (Sketch) Let $i, j, 1 \leq i, j \leq n$ be fixed. According to the last lemma we have $H(x)_{ij} = \sum_k a_{ki} \left(\frac{\partial \int_{F_k} h}{\partial x_j}\right)$ and we get with $g(y) = y + \zeta A_c^{(j)}$ $\frac{\partial \int_{F_k} h}{\partial x_j} = O(1)$ because of our Lipschitz-assumptions on the density h. As a consequence we obtain $|HF(x)|_2 = \max_{|z|,|y|=1} |z^T HF(x) y| \leq O(n^{1.5})$. and the lemma follows. □

With the help of the last two lemmata and Lemma 1 the proof of Lemma 7 is complete. Thus, after the verification of smoothness we could use the analysis of the last section. But, since we get again a diameter of the set $M_\gamma \cap \{z | \vec{1}z \leq \vec{1}x\}$ is related to the objective function value $(M_\gamma \cap \{z | \vec{1}z \leq \vec{1}x\} \subset B(0, \sqrt{2}(\vec{1}x)))$, we could use once again a dynamic programming approach. The analysis is tedious and omitted in this abstract.

Remarks:

- If the componets of the vector b are independently distributed and if the density is bounded below by a constant, we can improve the running time of our algorithm to $O\left(n^2 \min\left\{\frac{\min_{y \in M_\gamma} cy, cx_f}{\epsilon^2}\right\}\right)$ by a direct bound on the length of the gradient and the second derivative without using the isoperimetric inequality.

- The method can be applied in a similar fashion to the Component Commonality Problem [17], where the costs of buying raw materials should be minimized, while the customer demands have to be satisfied with a certain fixed probability γ [10]. Our method yields also in this case superior running time bounds to the results known so far [9]. Furthermore, our method and analysis are much simpler than those in [9], that are based on the analysis the conductance of a certain Markov chain. Moreover, our method allows Hit and Run steps (which [9] does not), a feature that is very likely to speed up the algorithm a lot in practice [18].

6. Smoothing of convex sets

In this section we will describe a method to make certain convex sets smooth in order to apply our local search technique. This method does not apply to arbitrary convex sets, therefore we assume two conditions to be true for the given convex set S.

- S is up-monotone (i.e. $x \in S$ and $y \geq x$ (componentwise) implies $y \in S$
- $S \subset \mathbf{R}^n_+$)

The problem is again to optimize a given linear cost function c over S, while we assume to have an initial feasible point x_f. This problem has a lot of interesting applications, in fact all the application examples described earlier are special cases of this general problem.

The idea to get a smooth set from a given up-monotone set is easy to describe. We will consider only a subset S' of points x of S with the following property: the volume of the intersection of a certain cube with center x and S should be at least a constant. This set is provably smooth. Furthermore, the set deviates not very much from S (if we choose the cube to be small enough), so that optimizing over S yields a good solution for S. In the following we will give bounds on the smoothness and tackle the problem of computing the volume of the intersection of the cube and S. The following theorem summarizes the results of this section.

Theorem 6 *Suppose $\epsilon \in \mathbf{R}_+$ and $x_f \in S$. The random greedy algorithm described below applied to a suitable smoothed set $S' \subset S$ yields a $z \in S$ with $cz \leq (1+\epsilon)\min_{y \in S} cy$ in $O^*\left(\frac{n^{4.5}}{\epsilon^2}\min\left\{\frac{\min_{y\in S} cy}{\epsilon^2}, cx_f\right\}\right)$ calls to the membership oracle of S.*

It is again easy to see that we might assume that every component of the cost vector is positive and, after rescaling, we might assume $c = \vec{1}/\sqrt{n}$ (the rescaled set is still up-monotone and contained in the positive orthand). Furthermore, $D = \sqrt{2}(1x_f)$ is an upper bound on the diameter of $S \cap \{x|1x \leq 1x_f\}$.

We will describe our smoothing procedure first. Suppose $e^r_1, \ldots, e^r_n$ is an ONB of $\mathbf{R}^n$ with $e^r_1 = \vec{1}/\sqrt{n}$ and $C = \{x \in \mathbf{R}^n| -\frac{\epsilon}{2n} \leq x_i \leq \frac{\epsilon}{2n}\}$ for a given $\epsilon \in \mathbf{R}^+$ is the cube with center 0 and sidelength ϵ/n. Let $C^r = [e^r_1, \ldots, e^r_n]C$ be the image of C of the linear mapping $L : e_i \to e^r_i$. Thus, C^r is just the rotated cube of side length $d = \frac{\epsilon}{n}$ with center axis $e^r_1, \ldots, e^r_n$. Suppose $x \in \mathbf{R}^n$. We define $C^r(x)$ as the rotated cube with center x and $F(x) = \frac{\mathrm{vol}_n(C^r(x)\cap S)}{\mathrm{vol}_n(C^r(x))}$ with the level sets $M_\gamma = \{x \in \mathbf{R}^n|F(x) \geq \gamma\}$. One can show by an easy application of the BRUNN-MINKOWSKI Theorem [2], that M_γ is convex, so that these definitions fit into our framework of smooth sets. □

We will consider the problem of determining the membership of M_γ of an arbitrary point $x \in \mathbf{R}^n$. Because computing the volume of $M_\gamma \cap C^r(x)$ is in general hard [4], we use a sampling approach instead. We will generate $\log^c n$ points in $C^r(x)$ uniformly at random and say $x \in M_\gamma$, if at least a fraction γ of these points hit S. Using Chernoff bounds [12] it is easy to see that we can decide the membership with an error probability of $O(1/n^c)$. We make a small error here (i.e. we compute membership of $M_{\gamma+\delta}$ with a small constant δ instead of M_γ), but this is not important since we will show the smoothness of every level set M_γ with $0 < \gamma < 1$.

To make the analysis easier we replace the surface of S by a C^∞ curve. We can do this without loss of generality, since this approximation can be made arbitrarily exact. For the smoothness analysis of M_γ we consider the following special case first. Suppose h is the normal vector of a tangent hyperplane of S. We assume the following condition to be true.

$$|(h\vec{1})/\sqrt{n}| \leq 1 - \Omega\left(\frac{\epsilon}{n}\right) \tag{4}$$

This condition implies that there is at least an angle of $\frac{\epsilon}{\sqrt{n}}$ between the normal vector and the cost vector $\vec{1}/\sqrt{n}$. It is a necessary condition for the smoothness of F, but if it does not hold we are near optimal anyway, as we will see later.

Lemma 10 *Suppose (4) holds. This implies that every border plane of $C^r(x)$ and every tangent hyperplane of S include at least an angle of $\frac{\epsilon}{\sqrt{n}}$.*

Proof: Easy calculation using the up-monotonicity of S, i.e. the components of a normal vector of a tangent hyperplane have the same sign. □

To get a suitable bound on the smoothness of M_γ we consider again the length of the gradient at an arbitrary border point $x \in M_\gamma$.

Lemma 11 $|\nabla F(x)|_2 = \Omega\left(\frac{d^{n-1}}{n}\right)$ *(d is the sidelength of the rotated cube).*

Proof: Let Σ be the $n-1$-dimensional surface between $C^r(x) \cap S$ and $C^r(x) \setminus S$. We denote the normalvector of the tangent hyperplane of S (in direction of S) by v_t. This implies $\nabla F = \int_\Sigma v_t$ (as a vector). Because S is up-monotone, every component of v_t is non-negative. We get $|\nabla F(x)|_1 = \mathrm{vol}_{n-1}(\Sigma)$, since v_t is a unit vector according to the L_2 norm. The isoperimetric inequality [4] implies $\mathrm{vol}_{n-1}(\Sigma) \geq \frac{1}{\mathrm{diam}(C^r(x))} \min\{\gamma d^n, (1-\gamma)d^n\}$. Therefore we get $|\nabla F(x)|_2 \geq \frac{1}{\sqrt{n}}|\nabla F(x)|_1 = \Omega\left(\frac{d^{n-1}}{n}\right)$. □

Next, we will give a bound on the Hessian $|HF(x)| = \max_{|z|,|y|=1} |z^T HF(x)y|$.

Lemma 12 $|HF(x)| \leq \frac{d^{n-2}n^{1.5}}{\epsilon}$.

Proof: (Sketch) The entries of the Hessian are related to the $(n-2)$-dimensional border lines of S on the facets of $C^r(x)$. Due to convexity of S we get an upper bound of these by considering the $(n-2)$-dimensional surface area of these facets. Using the lower bound on the angle between tangent hyperplanes and the facets of the cube we can derive the desired result. The details are omitted from this abstract. □

Proof of Theorem 6: (Sketch) Under the assumption that we never encounter a point $y \in C^r(x)$, where the assumption (4) does not hold, our analysis of the random greedy algorithm in the last section runs through analogously. In this case it is easy to see that the result of Theorem 6 is an implication of the general result in section 5. If we encounter a point, where (4) does not hold, we can argue, using a similar proof as in Lemma 2, that we are already near optimal. □

7. Current Research and Open Problems

We have described a very simple approach with good running time bounds in many application examples. One interesting question is of course, whether these easily implementable methods have any value in real world problems. There are some very promising empirical results of Hit and Run Algorithms for Global Optimization [18] and we are currently testing our approach.

On the theoretical side there are still many open questions. The notation of smooth sets seems to be applicable in a lot of different areas and we are currently working on extending our results to optimize general convex functions over convex smooth sets. There are also interesting application examples, like the two stage programming problems in stochastic optimization, that recently gained interest, when applied to problems in Computational Finance.

Moreover, it seems to be natural to use a ball instead of a rotated cube for smoothing convex sets as described in the last section. Intuitively, this procedure should also yield a smooth set. There is, however, the problem of bounding the second derivative that is related to the $(n-2)$-dimensional volume of the intersection of the convex set with the ball, which seems to be hard. At the moment, we do not know of any method to get suitable bounds.

References

[1] D. Applegate and R. Kannan. Sampling and integration of near log-concave functions. *Proc. of 23th ACM STOC*, 1990.

[2] M. Berger. *Geometry I.* Springer Verlag, 1980.

[3] G. Dantzig. Linear programming under uncertainty. *Management Science 1*, 1955.

[4] M. Dyer and A. Frieze. Computing the volume of convex bodies: A case where randomness provably helps. *Proceedings of the Symposium on Applied Math*, 1991.

[5] M. Dyer, A. Frieze, and R. Kannan. A random polynomial time algorithm for approximating the volume of convex bodies. *Journal of the ACM 38*, 1991.

[6] M. Grötschel, L. Lovasz, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization.* Springer Verlag, 1988.

[7] J. Jayaraman, J. Srinivasan, and R. Roundy. Procurement of common components in a stochastic environment. *IBM Technical Report*, 1992.

[8] R. Kannan, L. Lovasz, and M. Simonovits. Random walks and an n^5 volume algorithm. *Random Structures and Algorithms*, 1997.

[9] R. Kannan, J. Mount, and S. Tayur. A randomized algorithm to optimize over certain convex sets. *Mathematics of Operations Research 20*, 1995.

[10] R. Kannan and A. Nolte. A fast random greedy algorithm for the component commonality problem. *Proc. of ESA*, 1998.

[11] M. Luby and R. Nisan. Positive linear programming. *25th ACM STOC*, 1992.

[12] R. Motwani and P. Raghavan. *Randomized algorithms.* Cambridge University Press, 1995.

[13] J. Nesterov and A. Nemirovsky. *Interior Point Polynomial Methods in Conex Programming: Theory and Applications.* SIAM, Philadelphia, 1994.

[14] M. Prekopa. *Numerical Solutions to Probabilistic Constrained Programming Problems in Ermoliev, Yu.; Wets, R.: Numerical Techniques for Stochastic Optimization.* Springer, 1980.

[15] T. Terlaky. *Interior point methods of mathematical programming.* Kluwer Academic Publishers, 1996.

[16] P. Vaidya. Speeding up linear programming using fast matrix multiplication. *Proc. of 30th FOCS*, 1989.

[17] R. Wets. *Stochastic Programming in Handbooks of Operations Research and Management Science Vol. 1.* North Holland, 1989.

[18] Z. Zabinsky and et al. Improving hit-and-run for global optimization. *Journal Global Optimization 3*, 1993.