Computational Issues and Related Mathematics of an Exponential Annealing Homotopy for Conic Optimization

by

Jeremy Chen

B.Eng, Mechanical Engineering, National University of Singapore (2006)

Submitted to the Computation for Design and Optimization Program in partial fulfillment of the requirements for the degree of

Master of Science in Computation for Design and Optimization

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

August 2007
[September 2007]

Author ....
Computation for Design and Optimization Program
August 24, 2007

Certified by......... ....
Robert M. Freund
Theresa Seley Professor in Management Sciences
Sloan School of Management
Thesis Supervisor

Accepted by........ ....
Robert M. Freund
Theresa Seley Professor in Management Sciences
Sloan School of Management

Computational Issues and Related Mathematics of an Exponential Annealing Homotopy for Conic Optimization

by

Jeremy Chen

Submitted to the Computation for Design and Optimization Program
on August 24, 2007, in partial fulfillment of the
requirements for the degree of
Master of Science in Computation for Design and Optimization

Abstract

We present a further study and analysis of an exponential annealing based algorithm for convex optimization. We begin by developing a general framework for applying exponential annealing to conic optimization. We analyze the hit-and-run random walk from the perspective of convergence and develop (partially) an intuitive picture that views it as the limit of a sequence of finite state Markov chains. We then establish useful results that guide our sampling.

Modifications are proposed that seek to raise the computational practicality of exponential annealing for convex optimization. In particular, inspired by interior-point methods, we propose modifying the hit-and-run random walk to bias iterates away from the boundary of the feasible region and show that this approach yields a substantial reduction in computational cost.

We perform computational experiments for linear and semidefinite optimization problems. For linear optimization problems, we verify the correlation of phase count with the Renegar condition measure (described in [13]); for semidefinite optimization, we verify the correlation of phase count with a geometry measure (presented in [4]).

Thesis Supervisor: Robert M. Freund
Title: Theresa Seley Professor in Management Sciences
Sloan School of Management

Acknowledgments

I would like to first thank my advisor Professor Robert Freund for being a good, kind and concerned advisor. His expertise and intuition in convex optimization as well as knowledge of the field helped me greatly in this work, and made for a number of interesting and enlightening segways.

Needless to say, I would like to thank my friends and family back home who have not forgotten me even in my extended absence. In particular, my parents, who have continued to nag me from thousands of miles away, and my awesome, awesome brother who is and has continued to be awesome.

Finally, I would also like to thank the folks at Epsilon Theta who have debunked the rumors of MIT being a "dark depressing place". Staying at $\epsilon\theta$ has been a joy that has far outweighed the annoyance of living 1.5 miles from campus. (*Love, Truth, Honor, Ducks!*)

Contents

1 Introduction **15**

1.1 Prior Work 16

1.2 A Framework for Conic Optimization 17

1.3 Computing the Interval of Allowed Step Lengths 18

1.4 Exponential Annealing in Brief 20

1.5 Structure of the Thesis 22

2 Random Walks and Sampling **23**

2.1 The Hit-and-Run Random Walk 23

2.1.1 A Finite Approximation 24

2.1.2 The Convergence of Approximations 25

2.2 Sampling 31

2.2.1 Distances Between Distributions 33

2.2.2 Large Deviation Bounds 38

3 Computational Methods **45**

3.1 Components of an Algorithm: Improvements and Heuristics 45

3.1.1 The Next Phase: Where to Begin 46

3.1.2 Rescaling towards Isotropicity 53

3.1.3 Truncation 55

3.2 An Algorithm 67

3.2.1 Exponential Annealing for Conic Optimization 67

3.2.2 On the Operation Count for Semidefinite Optimization 68

4 Computational Results and Concluding Remarks 71

4.1 Correlation of Required Phases to Measures of Optimization Problems 71

4.1.1 Condition Measures 72

4.1.2 Geometry Measures 73

4.1.3 Correlation for Linear Optimization Problems 74

4.1.4 Correlation for Semidefinite Optimization Problems 77

4.2 Possible Extensions 84

4.2.1 Stopping Rules 84

4.2.2 A Primal-Dual Reformulation for a Guarantee of ϵ-Optimality 85

4.2.3 Adaptive Cooling 86

4.3 Conclusions 88

A Addendum for Semidefinite Optimization 89

A.1 Maximum Step Lengths in Semidefinite Optimization 89

A.2 On the Computational Cost of a Single IPM Newton Step for Semidefinite Optimization 91

B Computational Results for NETLIB and SDPLIB Test Problems 93

B.1 Linear Optimization Problems from the NETLIB Suite 93

B.2 Semidefinite Optimization Problems from the SDPLIB Suite 97

List of Figures

2-1 The Cone Construction 32

2-2 Total Variation Distance for Exponential Distributions on a Cone . . 35

2-3 Total Variation Distance between Exponential Distributions and the Distributions of Sample Means on a Cone 37

3-1 $\text{Cone}_{20,10}$ (Single Trajectory, No Truncation) 48

3-2 $\text{Cone}_{20,10}$ (Mean Used, No Truncation) 49

3-3 $\text{Cone}_{20,10}$ (Single Trajectory, Truncation Used) 49

3-4 $\text{Cone}_{20,10}$ (Mean Used, Truncation Used) 50

3-5 AFIRO (Single Trajectory, No Truncation) 50

3-6 AFIRO (Mean Used, No Truncation) 51

3-7 AFIRO (Single Trajectory, Truncation Used) 51

3-8 AFIRO (Mean Used, Truncation Used) 52

3-9 Rescaling Illustrated 55

3-10 The Effect of the Boundary 56

3-11 Truncation Illustrated 57

3-12 Phase Count on a Cone versus p ($n = 10$) 58

3-13 Phase Count on a Cone versus p ($n = 20$) 59

3-14 Phase Count on a Cone versus p ($n = 40$) 59

3-15 Phase Count on a Cone versus p ($n = 100$) 60

3-16 No Truncation: NETLIB Problem AFIRO ($n = 32$, $n_{\text{affine}} = 24$) . . . 62

3-17 No Truncation: NETLIB Problem SHARE2B ($n = 79$, $n_{\text{affine}} = 66$) . 63

3-18 No Truncation: NETLIB Problem STOCFOR1 ($n = 111$, $n_{\text{affine}} = 48$) 63

3-19 Schedule 1: NETLIB Problem AFIRO ($n = 32$, $n_{\text{affine}} = 24$) 64

3-20 Schedule 1: NETLIB Problem SHARE2B ($n = 79$, $n_{\text{affine}} = 66$) 64

3-21 Schedule 1: NETLIB Problem STOCFOR1 ($n = 111$, $n_{\text{affine}} = 48$) . . 65

3-22 Schedule 2: NETLIB Problem AFIRO ($n = 32$, $n_{\text{affine}} = 24$) 65

3-23 Schedule 2: NETLIB Problem SHARE2B ($n = 79$, $n_{\text{affine}} = 66$) 66

3-24 Schedule 2: NETLIB Problem STOCFOR1 ($n = 111$, $n_{\text{affine}} = 48$) . . 66

4-1 Required Iterations versus Dimension for 5 NETLIB Problems 75

4-2 Required Iterations versus $C(d)$ for 5 NETLIB Problems 75

4-3 Required Iterations versus $C(d)$ for 5 NETLIB Problems 76

4-4 Required Iterations versus Dimension for 13 SDPLIB Problems 77

4-5 Required Iterations versus Order of Matrices in the Semidefinite Cone for 13 SDPLIB Problems 78

4-6 Required Iterations versus $C(d)$ for 5 SDPLIB Problems 79

4-7 Required Iterations versus $C(d)$ for 5 SDPLIB Problems 80

4-8 Required Iterations versus g^m for 5 SDPLIB Problems 80

4-9 Required Iterations versus g^m for 5 SDPLIB Problems 81

4-10 Required Iterations versus $C_D(d)$ for 13 SDPLIB Problems 81

4-11 Required Iterations versus $C_D(d)$ for 13 SDPLIB Problems 82

4-12 Required Iterations versus g_d for 13 SDPLIB Problems 82

4-13 Required Iterations versus g_d for 13 SDPLIB Problems 83

B-1 NETLIB Problem AFIRO ($n = 32$, $n_{\text{affine}} = 24$) 94

B-2 NETLIB Problem ISRAEL ($n = 142$, $n_{\text{affine}} = 142$) 95

B-3 NETLIB Problem SCAGR7 ($n = 140$, $n_{\text{affine}} = 56$) 95

B-4 NETLIB Problem SHARE2B ($n = 79$, $n_{\text{affine}} = 66$) 96

B-5 NETLIB Problem STOCFOR1 ($n = 111$, $n_{\text{affine}} = 48$) 96

B-6 SDPLIB Problem control1 ($n = 21$, $s = 15$) 98

B-7 SDPLIB Problem hinf1 ($n = 13$, $s = 14$) 99

B-8 SDPLIB Problem hinf2 ($n = 13$, $s = 16$) 99

B-9 SDPLIB Problem hinf3 ($n = 13$, $s = 16$) 100

B-10 SDPLIB Problem hinf4 ($n = 13$, $s = 16$) 100

B-11 SDPLIB Problem hinf5 ($n = 13$, $s = 16$) 101

B-12 SDPLIB Problem hinf6 ($n = 13$, $s = 16$) 101

B-13 SDPLIB Problem hinf7 ($n = 13$, $s = 16$) 102

B-14 SDPLIB Problem hinf8 ($n = 13$, $s = 16$) 102

B-15 SDPLIB Problem truss1 ($n = 6$, $s = 13$) 103

B-16 SDPLIB Problem truss4 ($n = 12$, $s = 19$) 103

List of Tables

3.1 Number of Phases Required with Various Schedules 61

B.1 Computational Results for 5 NETLIB Suite Problems 94

B.2 Computational Results for 13 SDPLIB Suite Problems 98

Chapter 1

Introduction

We consider randomized methods for convex optimization, where "randomized" refers to a process of random sampling on the feasible region. Randomized methods operate on the paradigm of sampling from a distribution or a sequence of distributions on the feasible region such that some function of the samples converge in some sense to the solution of the problem.

Unlike interior point methods (IPMs), the randomized methods we consider are insensitive to how the feasible set is defined. One is only concerned with how a random line through a feasible point intersects the feasible region. Hence, redundant constraints or "massive degeneracy" near the optimal solution set (that pushes the analytic center away from the optimal solution set) have absolutely no effect on the performance of randomized methods.

The method we are studying works on the basis of sampling from a sequence of exponential distributions on the feasible set (under the assumption of bounded level sets) such that in the limit, samples converge to the set of optimal solutions with probability 1. As with all randomized methods, the most important issue is that of generating samples, and much of the computational work and algorithmic analysis deals with this issue.

It is the hope (far-fetched as it might presently seem) that randomized methods might be improved to the point of being competitive with IPMs for solving large-scale optimization problems. For very large-scale problems, IPMs can be impractical, since

Newton steps become prohibitively expensive for large dimension and/or large dense Newton equation systems. While it seems that they are unlikely to be competitive for smaller problems, randomized methods have the inherent benefit of being efficient to implement on a parallel computer insofar as sampling is concerned, and it is conceivable that randomized methods may hold the key to efficient solution of very large-scale optimization problems for this reason.

This thesis studies exponential annealing applied to conic optimization as was presented in [7] by Kalai and Vempala. (It is described in that paper as "simulated annealing" but we prefer using "exponential annealing" as the method is better described as a homotopy method rather than as a sequence of candidate solutions.)

1.1 Prior Work

Recent work explicitly directed at the complexity of solving convex optimization problems by a randomized algorithm began with Bertsimas and Vempala [1], which was directed at solving feasibility problem. Their algorithm makes use of constant factor volume reduction (with high probability) of the subset of $\mathbb{R}^n$ containing the feasible region by appropriate half-space intersection. Given an algorithm that solves the feasibility problem by returning a feasible point or reporting failure, one needs to call it $O(\log 1/\epsilon)$ times to solve the optimization problem to an optimality tolerance of ϵ using binary search.

This was followed by the work on simulated annealing presented in [7] by Kalai and Vempala. In that work, a sequence of probability distributions on the feasible region are generated approximately and sampled from. These distributions are chosen such that their means converge to a point in the set of optimal solutions. (These distributions on the feasible set are sampled from by simulating a Markov chain known as the hit-and-run random walk.) It is their work that we hope to develop further herein.

1.2 A Framework for Conic Optimization

For the purposes of this work, however, we will only consider linear objective functions of the form $c^T x$, where $c \in \mathbb{R}^n$ is the cost vector and $x \in \mathbb{R}^n$ is the decision variable. We assume the existence of an optimal solution and that the level sets of the objective function are bounded, whereby a distribution on the feasible region with density proportional to $e^{-c^T x/T}$ is well defined for $T > 0$.

Given a linear objective function $c^T x$ and a regular convex cone C, we are interested in solving the following optimization problem:

$$(P) \qquad \begin{array}{lrcl} \min\limits_{x} & c^T x & & \\ s.t. & b^{(e)} - A^{(e)}x & = & 0 \\ & b^{(c)} - A^{(c)}x & \in & C, \end{array}$$

where $b^{(e)} \in \mathbb{R}^{m_e}$, $A^{(e)} \in \mathbb{R}^{m_e \times n}$, $b^{(c)} \in \mathbb{R}^{m_c}, A^{(c)} \in \mathbb{R}^{m_c \times n}$. The constraints $b^{(e)} - A^{(e)}x = 0$ are the equality constraints, while the constraint $b^{(c)} - A^{(c)}x \in C$ describes the cone constraints.

In order to begin, we require a point in the feasible set. To obtain that, we may solve the following problem:

$$(P') \qquad \begin{array}{lrcl} \min\limits_{x,\delta,\theta} & \theta & & \\ s.t. & \delta b_e - A_e x - \theta b_e & = & 0 \\ & \delta b_c - A_c x + \theta(y - b_c) & \in & C \\ & \delta & \geq & 0 \\ & \|(x, \delta)\|_2 & \leq & 2 \end{array}$$

for which $(x, \delta, \theta) = (0, 1, 1)$ is strictly feasible for some given $y \in int\, C$.

To briefly digress, we begin with a strictly feasible point because the Markov chain of the hit-and-run random walk converges rapidly to its stationary distribution to the extent that the starting point is in the interior of the feasible region. A step of the hit-and-run random walk can be informally described as follows: (i) starting from a point in the feasible region, (ii) pick a random direction, which defines a line whose intersection with the feasible region gives a line segment or half-line, (iii) then sample from the the desired probability distribution on the feasible region as restricted to that line segment/half-line. (The hit-and-run random walk will be described in greater detail in Section 2.1.)

Note also that the level sets of the feasible region of (P') are bounded, guaranteeing the existence of an exponential density on the feasible region of (P').

If (P) is feasible, one will be able to find a solution to (P'), (x, δ, θ), with $\bar{\delta} > 0$ and $\bar{\theta} \leq 0$, and $\frac{1}{\bar{\delta}-\bar{\theta}}\bar{x}$ is a feasible solution to (P). In addition, if $\bar{\theta} < 0$, $\frac{1}{\bar{\delta}-\bar{\theta}}\bar{x}$ is a strictly feasible solution of (P).

1.3 Computing the Interval of Allowed Step Lengths

Given a strictly interior point, successive iterates of the hit-and-run random walk are strictly feasible with probability 1 (this is because we are sampling from a distribution with a density). It is relatively simple to sample on an affine subspace, by either generating a basis or using projections. Hence, the key to performing the hit-and-run random walk is the ability to determine the end points of a line intersecting the feasible region. Given a strictly interior point and a direction, this is the computation of the interval of allowed step lengths.

We assume that the only cones we will deal with are product cones arising from the non-negative orthant, the second-order cone, and the semidefinite cone. In order to perform hit-and-run, it is necessary to determine, for some $x \in C$, the maximum and minimum values of α such that $x + \alpha d \in C$. We choose d such that $\|d\| = 1$ and $-d$ is a descent direction.

The Non-Negative Orthant

The non-negative orthant is relatively easy to address. We have:

$$\{\alpha : x + \alpha d \geq 0\} = [\alpha_{-d}, \alpha_d]$$

where

$$\alpha_d = \min_i \left(\{-x_i/d_i : d_i < 0\} \cup \{\infty\}\right)$$

and

$$\alpha_{-d} = \max_i \left(\{-x_i/d_i \geq 0 : d_i > 0\} \cup \{-\infty\}\right).$$

The Second Order Cone

For the second order cone, given a point in the cone $x = (\bar{x}, t)$ and a direction $d = (d_{\bar{x}}, d_t)$, we want the maximum and minimum values of α such that $(\bar{x}, t) + \alpha(d_{\bar{x}}, d_t)$ lies in the cone. That is, $\|\bar{x} + \alpha d_{\bar{x}}\|_2 \leq t + \alpha d_t$. We require that:

$$(d_{\bar{x}}^T d_{\bar{x}} - d_t^2)\alpha^2 + 2(\bar{x}^T d_{\bar{x}} - t d_t)\alpha + (\bar{x}^T \bar{x} - t^2) \leq 0$$

and

$$t + \alpha d_t \geq 0.$$

By assumption, there exists a minimum value of α. By finding the region for which the first (quadratic) inequality holds and intersecting it with the region defined by the second (linear) inequality, we obtain the interval of values α may take.

The Semidefinite Cone

For the semidefinite cone, given an initial point X which we assume to be positive definite (since we begin in the strict interior of the cone) and a direction D, we require the positive semidefiniteness of $I + \alpha R^{-T} D R^{-1} = I + \alpha Q T Q^T$ where $X = R^T R$, Q is orthonormal, and T is symmetric and tridiagonal (as obtained from the Hessenberg form of $R^{-T} D R^{-1}$). We thus need to find the largest and smallest eigenvalues of T.

We may find the eigenvalue λ with the largest magnitude using power iteration. We then do the same for $T - \lambda I$ (which shifts all eigenvalues of T by $-\lambda$) to obtain the largest and smallest eigenvalues.

The addendum in A.1 of Appendix A describes an iterative method that helps achieve this end more efficiently.

1.4 Exponential Annealing in Brief

As previously mentioned, exponential annealing works on the basis of sampling from a sequence of exponential distributions on the feasible set such that samples converge to points in the set of optimal solutions.

The key to making exponential annealing work efficiently on arbitrary convex sets is the ability to sample on appropriate linear transformations of the $n-1$ dimensional unit sphere. This is achieved by sampling from multivariate normal distributions with covariances that accurately approximate the "shape" of the feasible region. For instance, sampling on the unit sphere when the feasible region is "thin" would result in very slow convergence of the distribution of iterates to the desired distribution. Although it is known that the hit-and-run random walk converges (see [9]), its worst-case complexity is very bad from a computational standpoint. The algorithm presented by Kalai and Vempala makes use of information from samples to estimate the shape of the set by computing an appropriate covariance matrix.

Before going further, we reproduce the algorithm of [7]:

The `UniformSample` routine picks a uniformly distributed random point from K. Additionally, it estimates the covariance matrix V_0 of the uniform distribution over the set K. This subroutine is the "Rounding the body algorithm" of Lovász and Vempala in [8], which uses X_{init}, $\mathcal{O}_K$, R, and r, and returns a nearly uniformly distributed point while making $O^*(n^4)$ membership queries. (The O^* notation supresses polylogarithmic factors.)

We next describe the hit-and-run random walk precisely. The routine (with its parameters shown), $\texttt{hit-and-run}(f, \mathcal{O}_K, V, x, k)$, takes as input a non-negative function f, a membership oracle $\mathcal{O}_K$, a positive definite matrix V , a starting point $x \in K$, and a number of steps k. It then performs the following procedure k times:

- Pick a random vector v according to the n-dimensional normal distribution with mean 0 and covariance matrix V (in order to sample on the $n-1$ unit sphere under the linear transformation R^T where $V = R^T R$). Let l be the line through the current point in the direction v.

```
Input  : $n \in \mathbb{N}$ (dimension)
       : $\mathcal{O}_K : \mathbb{R}^n \to \{0,1\}$ (membership oracle for convex set $K$)
       : $c \in \mathbb{R}^n$ (direction of minimization, $\|c\| = 1$)
       : $X_{init} \in K$ (starting point)
       : $R \in \mathbb{R}_+$ (radius of ball containing $K$ centered at $X_{init}$)
       : $r \in \mathbb{R}_+$ (radius of ball contained in $K$ centered at $X_{init}$)
       : $I \in \mathbb{N}$ (number of phases)
       : $k \in \mathbb{N}$ (number of steps per walk)
       : $N \in \mathbb{N}$ (number of samples for rounding)
Output: $X_I$ (candidate optimal solution for $(P)$)

$(X_0, V_0) \leftarrow$ UniformSample($X_{init}, \mathcal{O}_K, R, r$);
for $i \leftarrow 1$ to $I$ do
    $T_i \leftarrow R(1 - 1/\sqrt{n})^i$;
    $X_i \leftarrow$ hit-and-run($e^{-c^T x/T_i}, \mathcal{O}_K, V_{i-1}, X_{i-1}, k$);
    Update Covariance:
    for $j \leftarrow 1$ to $N$ do
        $X_j^i \leftarrow$ hit-and-run($e^{-c^T x/T_i}, \mathcal{O}_K, V_{i-1}, X_{i-1}, k$)
    end
    $\mu_i \leftarrow (1/N) \sum_{j=1}^N X_j^i$
    $V_i \leftarrow (1/N) \sum_{j=1}^N X_j^i (X_j^i)^T - \mu_i \mu_i^T$
end
Return $X_I$;
```

Algorithm 1: Simulated Annealing for Convex Optimization [7]

- Move to a random point on the intersection of l and K (this is a one-dimensional chord in K), where the point is chosen with density proportional to the function f restricted to the chord.

1.5 Structure of the Thesis

In Chapter 2, we study the hit-and-run random walk and talk about the issues relating to sampling. In Chapter 3, we propose improvements aimed at enhancing the practical performance of the algorithm. In Chapter 4, we present computational results, describe possible extensions and outline our conclusions.

Chapter 2

Random Walks and Sampling

2.1 The Hit-and-Run Random Walk

For our purposes, the hit-and-run random walk is a Markov chain defined on a state space consisting of points from a convex set $K \subset \mathbb{R}^n$. Let $\mathcal{H}$ be an arbitrary probability distribution on K. (The hit-and-run random walk is not an honest random walk in the sense of having i.i.d. increments, but we use this terminology to maintain consistency with the literature.)

Given X_n, X_{n+1} is defined by sampling uniformly on the unit sphere (possibly by taking a vector $v = (v_1, v_2, \ldots, v_n)$, where the v_i's are independent normal random variables, and returning $v/\|v\|_2$) or a non-singular linear transformation of the unit sphere in order to define a line $l_n = \{X_n + \alpha v : \alpha \in \mathbb{R}\}$. One then samples from the distribution of $\mathcal{H}$ restricted to $K \cap l_n$ to obtain X_{n+1}.

Under mild assumptions, the probability distribution of X_n, as n gets large, approaches $\mathcal{H}$.

Suppose that X is a random variable on K with an underlying σ-algebra $\mathcal{A}$. (A σ-algebra on K is a set of subsets of K closed under countable unions, countable intersections and complements, containing K and the empty set.) We define the Total Variation distance:

$$d_{TV}(\mu_F, \mu_G) = \sup_{A \in \mathcal{A}} |\mu_F(A) - \mu_G(A)|$$

where μ_F is the measure induced by the distribution function F and $\mu_F(A) := \int_A dF(x)$ (where we use the Riemann-Stieltjes integral here). For a density f, the corresponding distribution function is $F(x) := \int_{\substack{\bar{x}\in K \\ \bar{x}\leq x}} f(\bar{x})\, d\bar{x}$. Note that every density has a distribution function but the converse is not true. Note that this distance satisfies the triangle inequality. For the purposes of this work, we will work only with distributions that arise from density functions.

In [9], Lovász and Vempala show the convergence of the hit-and-run random walk for the uniform and exponential distributions in the Total Variation distance. (They experience technical difficulties in extending their results to arbitrary densities). The proof of their result is complicated.

Instead, here we take an elementary (and consequently more intuitive) approach to describing the hit-and-run random walk. We describe how one may approximate the hit-and-run random walk (for a distribution $\mathcal{H}$ with a positive density on all of K) with a finite state Markov chain. These approximations converge to the hit-and-run random walk, which is a Markov chain on a continuous state space. We also note that for each finite approximation, from any distribution on the states, the Markov chain converges to a steady state distribution. We also find that the steady state distributions converge to $\mathcal{H}$. The gap in this analysis, however, is our inability to show convergence to a steady state distribution from any initial distribution in the limit of a Markov chain with a continuous state space. Hence, this section is in some respects more general and in others more limited than the results of [9].

2.1.1 A Finite Approximation

Pick an arbitrary point in $\mathbb{R}^n$ as the origin for a Cartesian coordinate system on which a Cartesian grid will lie. We align the first coordinate direction with some arbitrary vector.

To each approximation, we associate a "mesh-size" ϵ, which describes how refined each grid is. We may index each grid cell with n integers $a_1, a_2, \ldots, a_n$. Let $(a_1, a_2, \ldots, a_n)$ refer to $[a_1\epsilon, (a_1+1)\epsilon] \times \ldots \times [a_n\epsilon, (a_n+1)\epsilon]$ (which we will refer to as an interval in $\mathbb{R}^n$). The union of these almost disjoint cells make up $\mathbb{R}^n$. (Note that for any probability measure with an underlying density, the intersection of any two cells has measure zero.)

Now, only a finite number of the aforementioned cells will intersect K if it is a convex body. If K is not bounded, by the assumption of bounded level sets of $c^T x$, we may consider instead $K \cap \{x : c^T(x - x_0) \leq 0\}$ for some $x_0 \in int\, K$, which is a convex body containing the optimal solution set. For each cell C such that $\mu_{\mathcal{H}}(C \cap K) > 0$, we associate a state. We have now defined the state space of the finite approximation.

We now describe the Markov chain by outlining a single step. Given that the current state ω_1 is associated with a cell C_1, randomly choose a point in $C_1 \cap K$ from the restriction of $\mathcal{H}$ to $C_1 \cap K$. One then performs a step of the hit-and-run random walk. The resulting point lies in some cell C_2, which is associated with some state ω_2. The step is then the transition from state ω_1 to state ω_2.

Clearly as $\epsilon \rightarrow 0$, we recover the hit-and-run random walk.

2.1.2 The Convergence of Approximations

Herein, we describe in detail the results outlined previously.

Let a finite state approximation with fineness ϵ, as defined above, have states $\omega_i^{(\epsilon)}$ $(i = 1, 2, \ldots, m_\epsilon)$ associated with cells $C_i^{(\epsilon)}$ respectively. Denote by $\pi_{\omega_i}^{(\epsilon)}$ the steady state probability of being in state $\omega_i^{(\epsilon)}$. Let $B_i^{(\epsilon)} = C_i^{(\epsilon)} \cap K$.

Proposition 2.1. *If $\mathcal{H}$ has a positive density on all of K, for the finite state Markov chain defined above, the conditional probability of entering any state, given the current state, is positive.*

Proof. Suppose the current state is associated with a cell $C_i^{(\epsilon)}$ and consider any state (possibly the same one) associated with a cell $C_j^{(\epsilon)}$. We know that $B_i^{(\epsilon)}$ and $B_j^{(\epsilon)}$ are convex bodies.

Since the density of the restriction of $\mathcal{H}$ to $B_i^{(\epsilon)}$ is positive everywhere, and $B_i^{(\epsilon)}$ is compact, the said density attains a minimum $a > 0$ on $B_i^{(\epsilon)}$.

Given any point $x \in B_i^{(\epsilon)}$, the set of lines through x that have an intersection with $B_j^{(\epsilon)}$ of at least δ (for $\delta > 0$ small enough) define a positive $n - 1$ dimensional volume on the unit sphere (or linear transformation thereof) since $B_j^{(\epsilon)}$ is a convex body; that volume takes up a fraction of the volume of the unit sphere $b_x > 0$. The minimum of the density function of the restriction of $\mathcal{H}$ to each such line through x exists, and the infimum over all lines through x, c_x, is positive since the density of $\mathcal{H}$ is positive. Let the minimum value of $b_x c_x$ over $B_i^{(\epsilon)}$ be d (which is attained by compactness).

Then the conditional probability of entering state j, given the current state is i is at least $ad\delta > 0$ and the result follows. ∎

Corollary 2.2. *If $\mathcal{H}$ has a positive density on all of K, for the finite state Markov chain defined above, starting from any initial distribution on its states, the Markov chain converges to a unique stationary distribution.*

Proof. By Proposition 2.1, the Markov chain can be represented by a stochastic matrix with positive entries. The result then follows from Perron-Fröbenius theory (see, for instance, [11]). ∎

Proposition 2.3. *$\mathcal{H}$ is stationary under the hit-and-run random walk.*

Proof. Starting from $\mathcal{H}$ and conditioned on picking a fixed unit vector v (when sampling uniformly on the unit sphere), if one were to perform the final part of the hit-and-run random walk (sampling on the set defined by the intersection of lines $\{x + \alpha v : \alpha \in \mathbb{R}\}$ with K where $x \in K$), the resulting density restricted to that line is necessarily preserved since the resulting distribution on that set is the density due to the restriction of $\mathcal{H}$ to that same line. Since this holds for each point v on the unit sphere or any non-singular linear transformation of it, the distribution $\mathcal{H}$ is stationary. ∎

Corollary 2.4. *The unique steady state probabilities are given by $\pi_{\omega_i}^{(\epsilon)} = \mu_{\mathcal{H}}\left(B_i^{(\epsilon)}\right)$.*

Proof. Let X_k and Y_k denote the iterates of the hit-and-run random walk and the finite approximation respectively.

By Proposition 2.3, we have

$$\begin{aligned}\mu_{\mathcal{H}}(B_i^{(\epsilon)}) &= \sum_j \mathbb{P}\left(X_{k+1} \in B_i^{(\epsilon)} | X_k \in B_j^{(\epsilon)}\right) \mu_{\mathcal{H}}(B_j^{(\epsilon)}) \\ &= \sum_j \mathbb{P}\left(Y_{k+1} \in \omega_i^{(\epsilon)} | Y_k = \omega_j^{(\epsilon)}\right) \mu_{\mathcal{H}}(B_j^{(\epsilon)}).\end{aligned}$$

Also, since $\sum_j \mu_{\mathcal{H}}(B_j^{(\epsilon)}) = 1$, $\pi_{\omega_i}^{(\epsilon)} = \mu_{\mathcal{H}}(B_i^{(\epsilon)})$ solves

$$\pi_{\omega_i}^{(\epsilon)} = \sum_j \mathbb{P}\left(X_{k+1} \in B_i^{(\epsilon)} | X_k \in B_j^{(\epsilon)}\right) \pi_{B_j^{(\epsilon)}}, \quad \sum_{i=1}^{m_\epsilon} \pi_{\omega_i}^{(\epsilon)} = 1, \quad \pi_{\omega_i}^{(\epsilon)} \geq 0.$$

The uniqueness of the solution follows from Proposition 2.1 and Perron-Fröbenius Theory (see, for instance, [11]). ■

Proposition 2.5. *If $\mathcal{H}$ has a continuous density function, then for a sequence of finite approximations with decreasing mesh-size $\epsilon_k > 0$ and $\lim_{k\to\infty} \epsilon_k = 0$, the steady state distributions converge to $\mathcal{H}$ in Total Variation.*

Proof. Consider a measurable subset A of K (i.e., one such that $\mu_{\mathcal{H}}(A)$ is well defined). For a measurable subset A, we can approximate A from above and below by the union of almost disjoint intervals in $\mathbb{R}^n$ (see [17]).

Let a sequence of upper and lower approximations be denoted by $\{\overline{A}_k\}$ and $\{\underline{A}_k\}$ (associated with mesh-size ϵ_k). Let

$$\overline{A}_k = \bigcup_{\mu_{\mathcal{H}}\left(B_i^{(\epsilon_k)} \cap A\right) > 0} B_i^{(\epsilon_k)}$$

$$\overline{A}_k = \bigcup_{B_i^{(\epsilon_k)} \subseteq A} B_i^{(\epsilon_k)}$$

Also, for some approximation A_k of a set A, let $\pi_{A_k} = \sum_{B_j^{(\epsilon_k)} \subseteq A_k} \pi_{\omega_j}^{(\epsilon_k)}$ (the lower and upper approximations give the inner and outer measures respectively).

Note that by Proposition 2.4, any subset $B_j^{(\epsilon)}$ of K has $\mu_{\mathcal{H}}(A) = \pi_{\omega_j}^{(\epsilon)}$. Note also that the measure of a measurable set is equal to the (countable) sum of the measure of a partition of the set into measurable subsets. In particular, intervals in $\mathbb{R}^n$ are measurable.

Now, since K is a convex body (hence closed and bounded), there exists a constant $C_{\mathcal{H}}$, the maximum of the density of $\mathcal{H}$ over K, such that for an interval R, $\mu_{\mathcal{H}}(R \bigcap K) \leq C_{\mathcal{H}} Vol(R)$ (where the volume of an interval is nothing but the product of its side lengths).

Clearly, we have $Vol(\underline{A}_k) \leq Vol(A) \leq Vol(\overline{A}_k)$ and $\pi_{\underline{A}_k} \leq \mu_{\mathcal{H}}(A) \leq \pi_{\overline{A}_k}$.

Also, both $Vol(A) - Vol(\underline{A}_k)$ and $Vol(\overline{A}_k) - Vol(A)$ converge to zero as $k \to \infty$ (remembering that $\underline{A}_k \subseteq A \subseteq \overline{A}_k$). We have that both

$$|\mu_{\mathcal{H}}(A) - \pi_{\underline{A}_k}| \leq C_{\mathcal{H}} |Vol(A) - Vol(\underline{A}_k)|$$

and

$$|\pi_{\overline{A}_k} - \mu_{\mathcal{H}}(A)| \leq C_{\mathcal{H}} |Vol(\overline{A}_k) - Vol(A)|.$$

Since this holds for every measurable subset of K, the steady state distributions generated by the finite state approximations converge to $\mathcal{H}$ in Total Variation. ■

To describe the rate of convergence of general irreducible Markov chains in Total Variation, we have a lemma from Chapter 11 of [12] whose proof requires the notion of "coupling" of Markov chains. This lemma is useful to us because Proposition 2.1 states that the conditional probability of entering any state given the current state is positive.

Lemma 2.6. *Given an irreducible finite state Markov chain with state space S, let $m_j = \min_{i \in S} \mathbb{P}(X_{k+1} = j | X_k = i)$ and $m = \sum_{j \in S} m_j$. Then*

$$d_{TV}(\pi^k, \bar{\pi}) \leq (1 - m)^k$$

where $\bar{\pi}$ is the stationary distribution of the chain and π^k is the distribution at the k-th step.

It is difficult to make statements about m in Lemma 2.6 as this value depends very much upon the shape of the set K (and what linear transformation is applied to the unit sphere). However, it is intuitively clear that m would approach 0 as the approximation becomes finer.

Now, suppose that the first coordinate direction of the underlying grid in the finite approximations is aligned with some vector c and we were to compute the "worst case" difference in the means of $c^T X^{(\epsilon)}$ and $c^T X$ with respect to the steady state distribution of the finite approximation with fineness ϵ and $\mathcal{H}$ respectively. We define the worst case mean with respect to vector c and fineness f to be:

$$W_c^{(\epsilon)} = \sum_{j=1}^{m_\epsilon} \pi_{\omega_j}^{(\epsilon)} \max\{\|c\|_2 x_1 : x \in C_j\}$$

Proposition 2.7.

$$W_c^{(\epsilon)} - \mathbb{E}_{\mathcal{H}}\left[c^T X\right] \leq \|c\|_2 \epsilon$$

Proof. Denote the restriction of $\mathcal{H}$ to $A \subseteq K$ to be $\mathcal{H}(A)$.

$$\begin{aligned}
W_c^{(\epsilon)} - \mathbb{E}_{\mathcal{H}}\left[c^T X\right] &= \sum_{j=1}^{m_\epsilon} \pi_{\omega_j}^{(\epsilon)} \max\{\|c\|_2 x_1 : x \in C_j\} - \sum_{j=1}^{m_\epsilon} \mu_{\mathcal{H}}\left(B_i^{(\epsilon)}\right) \mathbb{E}_{\mathcal{H}(B_j^{(\epsilon)})}\left[c^T X\right] \\
&= \sum_{j=1}^{m_\epsilon} \pi_{\omega_j}^{(\epsilon)} \max\{\|c\|_2 x_1 : x \in C_j\} - \sum_{j=1}^{m_\epsilon} \pi_{\omega_j}^{(\epsilon)} \mathbb{E}_{\mathcal{H}(B_j^{(\epsilon)})}\left[c^T X\right] \\
&= \|c\|_2 \sum_{j=1}^{m_\epsilon} \pi_{\omega_j}^{(\epsilon)} \left(\max\{x_1 : x \in C_j\} - \mathbb{E}_{\mathcal{H}(B_j^{(\epsilon)})}\left[X_1\right]\right) \\
&\leq \|c\|_2 \sum_{j=1}^{m_\epsilon} \pi_{\omega_j}^{(\epsilon)} \epsilon \\
&= \|c\|_2 \epsilon
\end{aligned}$$

■

Knowing the "worst case" mean is relevant since one is often unable to find the mean of the restriction of $\mathcal{H}$ to some cell $B_j^{(\epsilon)}$, but the "worst case" mean is readily computable given the underlying grid (and is, in fact, sometimes not attained).

The above result gives us an elementary proof for the convergence of a simplified exponential annealing type algorithm for convex optimization problems with linear objectives which may be informally described as follows:

1. Choose a sequence of distributions $\{\mathcal{D}_k\}$ with means that converge to some optimal solution, and an initial state corresponding to some $x \in K$.

2. Choose mesh-size ϵ.

3. Pick the next distribution to be used and simulate the Markov chain of the finite approximation until "suitably close" to convergence. If the mean for the distribution picked is "suitably close" to an optimal solution, terminate.

4. Repeat 3.

Naturally, since it was shown in [9] that hit-and-run converges for an exponential distribution, we will not be considering the finite approximations in computation, but the above analysis gives a different perspective on the hit-and-run random walk.

To conclude this section, we quote a result by Lovász and Vempala from [9] that describes the convergence of the hit-and-run random walk.

Theorem 2.8 (Lovász and Vempala [9]). *Let $K \in \mathbb{R}^n$ be a convex body and let f be a density supported on K which is proportional to $e^{a^T x}$ for some vector $a \in \mathbb{R}^n$. Assume that the level set of f of probability $1/8$ contains a ball of radius r and that $\mathbb{E}_f\left[|x - z_f|^2\right] \leq R^2$, where z_f is the mean of f. Let σ be a starting distribution and let σ^m be the distribution of the current point after m steps of hit-and-run applied to f. Let $\epsilon > 0$, and suppose that the density function $d\sigma/d\pi_f$ is bounded by M except on a set S with $\sigma(S) \leq \epsilon/2$. Then for*

$$m > 10^{30}\frac{n^2R^2}{r^2}\ln^5\frac{MnR}{r\epsilon},$$

the total variation distance of σ^m and π_f is less than ϵ.

Note that the density $d\sigma/d\pi_f$ can be readily interpreted as $d\sigma/d\pi_f(x) = \lim\limits_{\epsilon\to 0} \frac{\sigma(B_\epsilon(x)\cap K)}{\pi_f(B_\epsilon(x)\cap K)}$ where $B_r(x)$ is a ball of radius r centered at x.

2.2 Sampling

In performing the hit-and-run random walk on a convex set, it is our desire that we have convergence of a sequence of distributions $\{\mathcal{D}_k\}$ to some distribution where the probability measure of the optimal solution set is 1.

Noting that any convex optimization problem can be cast as a convex optimization problem with a linear objective, given a linear objective function (with cost vector c) on a convex set K, using a sequence of exponential distributions with a density proportional to $e^{-c^T x/T_k}$ (where $T_k \downarrow 0$), the mean of the distribution converges to a point in the optimal solution set. This is described in Lemma 4.1 in [7], reproduced here for completeness:

Lemma 2.9 (Kalai and Vempala [7]). *For any unit vector $c \in \mathbb{R}^n$, temperature $T > 0$, and X chosen according to a distribution with density proportional to* $\exp(-c^T x/T)$*,*

$$\mathbb{E}\left[c^T X\right] - \min_{x \in K} c^T x \leq nT$$

(Note that for an exponential distribution on K to exist, we require the level sets of $c^T x$ on K to be bounded.)

We are interested in accurately estimating the probability that certain functions of a random variable, drawn from a certain distribution, lie in some set. The function of the sequence of random variables described in the previous paragraph is nothing but $f(X) := \mathbb{E}[X]$. On a parallel track, we might also wish to find bounds on the probability that our estimates are off track.

Lemma 2.9 gives a bound on the difference between the mean of $c^T X$ and the global minimum. In estimating the mean of X, we perform m hit-and-run random walks starting from the same point, hoping that the distribution of each is sufficiently close to an exponential distribution on K. We then compute the mean of the m sample points. We would like to obtain a bound on the probability that $c^T \bar{X}$ exceeds $\mathbb{E}\left[c^T X\right]$ by a constant factor (where $\bar{X} = \frac{1}{m}\sum_{k=1}^{m} X^{(k)}$, and the $X^{(k)}$'s are drawn i.i.d. from the distribution resulting from performing the hit-and-run random walk).

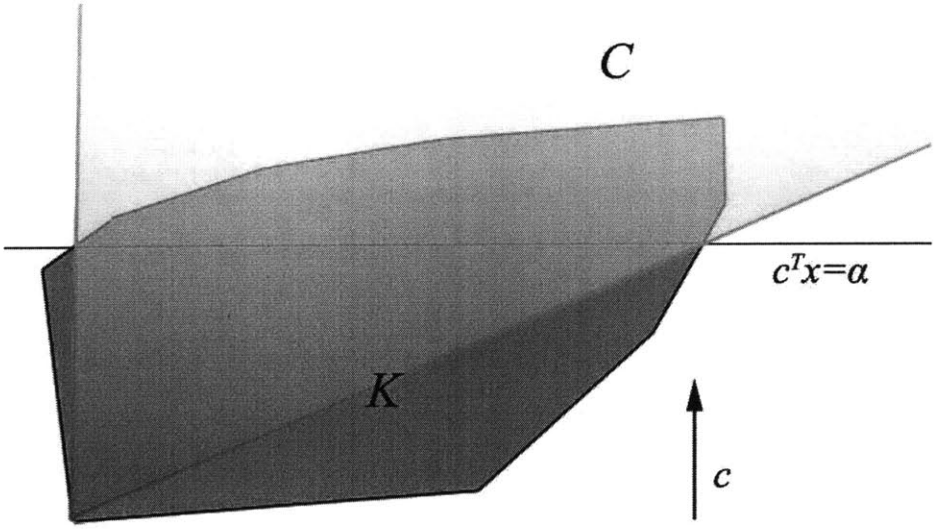

Figure 2-1: The Cone Construction

Given a convex set K and a point $\bar{x}$ in the set of optimal solutions, let $\alpha = \mathbb{E}\left[c^T X\right]$ (given a density proportional to $\exp(-c^T x/T)$). Consider the cone C with its vertex at $\bar{x}$ with the same α level set as K ($\{x \in K : c^T x = \alpha\} = \{x \in C : c^T x = \alpha\}$). It is obvious that the expectation of $c^T X$ over K is less than that over C given a density proportional to $\exp(-c^T x/T)$ on each set. This motivates the results given for cones below and is illustrated in Figure 2-1.

Lemma 2.10. *Let $c = (1, 0, 0, \ldots)$, the temperature be T, and $K \subseteq \mathbb{R}^n$ be a cone where $c^T x = x_1$ is minimized at its vertex, then the marginal of a distribution on K with density proportional to $e^{-c^T x/T}$ with respect to x_1 is an Erlang-n distribution with parameter $\lambda = 1/T$, and hence has a mean of nT, a variance nT^2, and moment generating function $g(r) = \left(\frac{\lambda}{\lambda - r}\right)^n$.*

Proof. Without loss of generality, by the memoryless property of the exponential distribution assume that the vertex of the cone is at a point where $x_1 = 0$.

The marginal density is given by

$$f(x_1) = \frac{C x_1^{n-1} e^{-\lambda x_1}}{\int_0^\infty C x_1^{n-1} e^{-\lambda x_1}\, dx_1}$$

where C is the $n-1$ dimensional volume of the cone at $x_1 = 1$.

Now,

$$\begin{aligned}
\int_0^\infty x^{n-1} \exp(-\lambda x)\, dx &= \frac{1}{\lambda^n} \int_0^\infty x^{n-1} e^{-x}\, dx \\
&= \frac{1}{\lambda^n} \left(\left[-x^{n-1} e^{-x} \right]_0^\infty + (n-1) \int_0^\infty x^{n-2} e^{-x}\, dx \right) \\
&= \frac{1}{\lambda^n} (n-1) \int_0^\infty x^{n-2} e^{-x}\, dx \\
&= \frac{(n-1)!}{\lambda^n}.
\end{aligned}$$

Hence,

$$f(x_1) = \frac{\lambda^n x_1^{n-1} e^{-\lambda x_1}}{(n-1)!}$$

which is the density of an Erlang-n distribution.

The rest follow from known properties of the Erlang-n distribution (see, for instance, [5]). ■

2.2.1 Distances Between Distributions

The next result describes the Total Variation distance between two exponential distributions on a cone with densities proportional to $e^{-c^T x/T_1}$ and $e^{-c^T x/T_2}$.

Corollary 2.11. *Under the assumptions of Lemma 2.10, given two distributions σ_1 and σ_2 on K with densities proportional to $e^{-c^T x/T_1}$ and $e^{-c^T x/T_2}$ respectively ($T_2 > T_1$),*

$$d_{TV}(\sigma_1, \sigma_2) = \left[1 + \sum_{k=1}^{n-1} \frac{(\bar{x}/T_2)^k}{k!} \right] \exp(-\bar{x}/T_2) - \left[1 + \sum_{k=1}^{n-1} \frac{(\bar{x}/T_1)^k}{k!} \right] \exp(-\bar{x}/T_1)$$

where

$$\bar{x} = \frac{n}{\frac{1}{T_1} - \frac{1}{T_2}} \ln \frac{T_2}{T_1}$$

Proof. Consider the marginals with respect to x_1 (referred to as x in this proof for brevity), f_1 and f_2 .

Referring to the proof of Lemma 2.10, one can verify that the density of σ_1 is greater than or equal to that of σ_2 for all $x_1 \in [0, \bar{x}]$ and the converse is true for all $x_1 \in [\bar{x}, \infty)$.

It follows that

$$d_{TV}(\sigma_1, \sigma_2) = \int_{\bar{x}}^{\infty} f_2(x) - f_1(x)\, dx.$$

Now, since

$$f_1(x) = \frac{(1/T_1)^n x^{n-1} e^{-(1/T_1)x}}{(n-1)!},$$
$$f_2(x) = \frac{(1/T_2)^n x^{n-1} e^{-(1/T_2)x}}{(n-1)!}$$

and

$$\begin{aligned}
\int_a^{\infty} \frac{\lambda^n x^{n-1} e^{-\lambda x}}{(n-1)!}\, dx &= \int_{a\lambda}^{\infty} \frac{x^{n-1} e^{-x}}{(n-1)!}\, dx \\
&= \left[\frac{-x^{n-1} e^{-x}}{(n-1)!}\right]_{a\lambda}^{\infty} + \int_{a\lambda}^{\infty} \frac{x^{n-2} e^{-x}}{(n-2)!}\, dx \\
&= \frac{(a\lambda)^{n-1} e^{-(1-\beta)n}}{(n-1)!} + \int_{a\lambda}^{\infty} \frac{x^{n-2} e^{-x}}{(n-2)!}\, dx \\
&= \left[1 + \sum_{k=1}^{n-1} \frac{(a\lambda)^k}{k!}\right] e^{-a\lambda},
\end{aligned}$$

the result follows. ■

Noting that the Total Variation distance between disjoint distributions is 1, consider $T_1 = 1/2T_2$. For $n = 2$, $d_{TV}(\sigma_1, \sigma_2) = 0.360787$; for $n = 10$, $d_{TV}(\sigma_1, \sigma_2) = 0.721328$; for $n = 50$, $d_{TV}(\sigma_1, \sigma_2) = 0.985251$; for $n = 100$, $d_{TV}(\sigma_1, \sigma_2) = 0.999442$; for $n = 1000$, $d_{TV}(\sigma_1, \sigma_2) = 1$ (to 20 digits of precision). This illustrates the dependence on dimension of the Total Variation distance.

On the other hand, one finds that if we consider $T_1 = \left(1 - \frac{1}{\sqrt{n}}\right) T_2$, the Total Variation distance varies with dimension as shown in Figure 2-2. (This is the temperature schedule suggested by Kalai and Vempala in [7].)

Now, if one would prefer to have the Total Variation Distance of the distribution of the sample mean and samples from the next exponential distribution after the

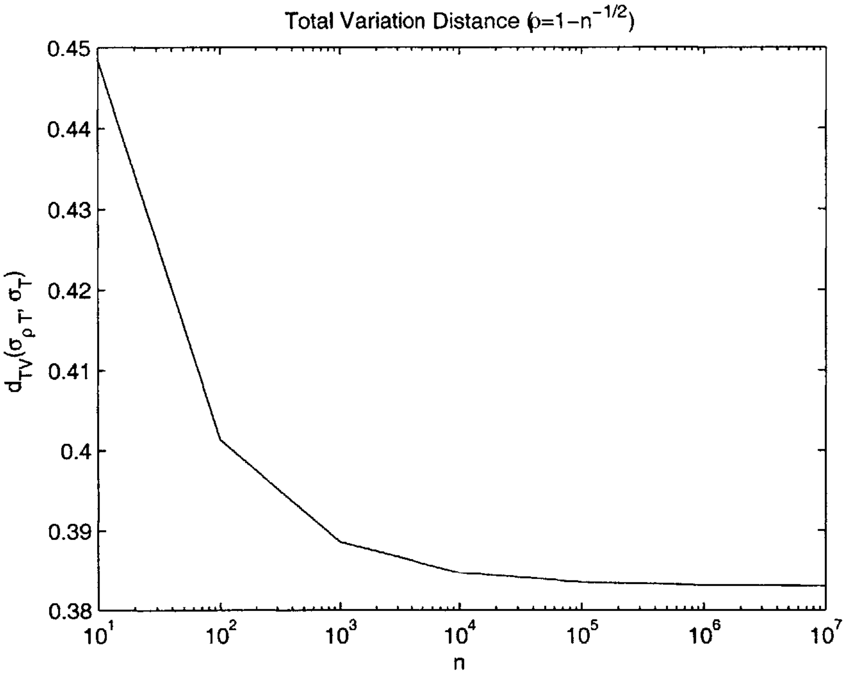

Figure 2-2: Total Variation Distance for Exponential Distributions on a Cone

temperature is decreased (later in this thesis, sample means are used to start the next random walk), the problem becomes far more difficult. The distribution of the sample mean on K no longer has a constant density on the level sets $\{x : c^T x = \alpha\}$. If one were to consider only the distance between marginal distributions of the sample mean and a sample point at the same temperature (which some thought would show gives an upper bound), this is not easy to find since the marginal distribution of the sample mean (using m sample points) is an Erlang-mn distribution with parameter m/T_1 (this can be established by considering the moment generating function). The Total Variation distance between the distribution of the sample mean at temperature T_1 and the distribution of points at temperature T_2 can then be bounded above by using the triangle inequality. It is thus unclear how to find the Total Variation distance between the two.

At a given temperature T, with respect to the direction of the cost vector, the marginal density of the distribution of sample means dominates the marginal density of sample points only on an interval, $[x_-, x_+]$. This can be verified by considering the densities of an Erlang-mn distribution with parameter m/T and that of an Erlang-n distribution with parameter $1/T$. To determine the values of x_- and x_+, we proceed by performing the same procedure in the proof of Corollary 2.12, and find that x_- and x_+ are the roots of

$$xe^{-x/nT} = \frac{T}{m}\left[\frac{1}{m^n}\frac{(mn-1)!}{(n-1)!}\right]^{1/(mn-n)} .$$

To show that real distinct positive roots exist for $m > 1$, observe that at the mean, the marginal density of the sample mean distribution takes a greater value. The marginal density of the sample mean distribution takes a smaller value than the distribution of points as x approaches infinity (note the order of the exponential). At sufficiently small values of x, where the polynomial and factorial terms dominate, the marginal density of the sample mean distribution takes a smaller value. (Note that in the above equation, after factoring x out n times, 0 is not a root.) We omit to show that there are only two positive roots.

Applying a similar procedure as the proof of Corollary 2.12, we have given a sketch of the proof of:

Corollary 2.12. *Under the assumptions of Lemma 2.10, given two distributions σ_1 and σ_2 on K with densities proportional to $e^{-c^T x/T_1}$ and $e^{-c^T x/T_2}$ respectively ($T_2 > T_1$), let $\tilde{\sigma}_1^{(m)}$ be the distribution of the mean of $m > 1$ samples from the distribution σ_1, then*

$$d_{TV}(\tilde{\sigma}_1^{(m)}, \sigma_2) \leq d_{TV}(\sigma_1, \sigma_2) + D_{1,m}^{(n)}$$

where

$$\begin{aligned} D_{1,m}^{(n)} &= \left[1 + \sum_{k=1}^{mn-1} \frac{(mx_-/T_1)^k}{k!}\right] \exp(-mx_-/T_1) - \left[1 + \sum_{k=1}^{n-1} \frac{(x_-/T_1)^k}{k!}\right] \exp(-x_-/T_1) \\ &\quad - \left[1 + \sum_{k=1}^{mn-1} \frac{(mx_+/T_1)^k}{k!}\right] \exp(-mx_+/T_1) + \left[1 + \sum_{k=1}^{n-1} \frac{(x_+/T_1)^k}{k!}\right] \exp(-x_+/T_1) \end{aligned}$$

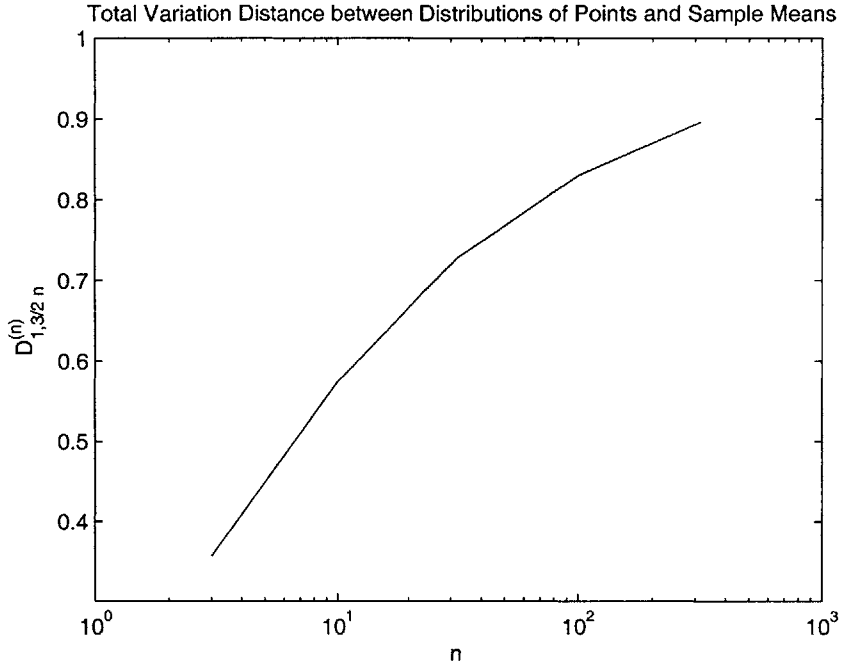

Figure 2-3: Total Variation Distance between Exponential Distributions and the Distributions of Sample Means on a Cone

and x_-, x_+ (with $x_- < x_+$) are the roots of

$$xe^{-x/nT_1} = \frac{T_1}{m}\left[\frac{1}{m^n}\frac{(mn-1)!}{(n-1)!}\right]^{1/(mn-n)}.$$

This Corollary may be used to yield insights like those from Figure 2-2 since before finding the required roots, we can make the transformation $x = T_1\bar{x}$ to remove T_1 from the system, find the two new roots $\bar{x}_-$ and $\bar{x}_+$, and have $x_- = T_1\bar{x}_-$, $x_+ = T_1\bar{x}_+$. This can be seen to remove the dependence on T_1 in the term $D^{(n)}_{1,m}$.

In Figure 2-3, with $m = 3/2\ n$, we observe that $D^{(n)}_{1,m}$ increases rapidly, apparently towards 1, making the bound meaningless. (Figure 2-3 is given for a limited range of n as approximately evaluating the right hand side of the equation for finding the roots was expensive for large n.) Hence, we are unable to establish any *theoretical* justification from using the sample mean as a starting point for the next random walk. This outcome, however, is not entirely unexpected.

2.2.2 Large Deviation Bounds

In sampling from a distribution, it is desirable to know how many sample points to choose in order to ensure that large deviations of the sample mean (cost) from the true mean (cost) occur with low probability. We continue with a proposition that addresses this.

Proposition 2.13. *Suppose $K \subseteq \mathbb{R}^n$ is a cone where $c^T x = x_1$ is minimized at its vertex. Let $c = (1, 0, 0, \ldots)$, the temperature be $T > 0$, $\bar{X} = \frac{1}{m}\sum_{k=1}^{m} X^{(k)}$ and $\mu = \mathbb{E}\left[c^T X\right] = \mathbb{E}\left[X_1\right]$. If the $X^{(k)}$'s are drawn i.i.d. from an exponential distribution on K proportional to $e^{-c^T x/T}$, then*

$$\mathbb{P}\left(c^T \bar{X} \geq (1+\beta)\mu\right) \leq \exp[-(\beta - \ln(1+\beta))nm]$$

for $\beta > 0$, and

$$\mathbb{P}\left(c^T \bar{X} \leq (1-\beta)\mu\right) \leq \exp[(\beta + \ln(1-\beta))nm]$$

for $\beta \in [0, 1)$.

Proof. We show the proof of the first and more important bound, the other result may be proven similarly. For convenience, let $\lambda = 1/T$. Using Lemma 2.10, we obtain the marginal distribution of $X_1 = c^T X$ and its associated properties.

For $r > 0$,

$$\mathbb{P}\left(\bar{X} \geq (1+\beta)\mu\right) = \mathbb{P}\left(\exp\left(r\sum_{k=1}^{m} X^{(k)}\right) \geq \exp\left(rm(1+\beta)\mu\right)\right).$$

Using the Markov inequality,

$$\begin{aligned}\mathbb{P}\left(\bar{X} \geq (1+\beta)\mu\right) &\leq \mathbb{E}\left[\exp\left(r\sum_{k=1}^{m} X^{(k)}\right)\right]\exp\left(-rm(1+\beta)\mu\right)\\ &= g(r)^m \exp\left(-rm(1+\beta)\mu\right)\\ &= \left(\frac{\lambda}{\lambda - r}\right)^{mn}\exp\left(-rm(1+\beta)\mu\right)\\ &=: h(r,m,n,\beta).\end{aligned}$$

Letting $r = \lambda p$, (and noting that $\mu = n/\lambda$), we have

$$h(\lambda p, m, n, \beta) = \left(\frac{1}{1-p}\right)^{mn}\exp[-(1+\beta)mnp].$$

We also find that the first term (due to the moment generating function has an asymptote at $p = 1$, and we note further that $h(\lambda p, m, n, r)$ is strictly convex in p over $(0,1)$ by computing its second derivative and finding it to be

$$\frac{d^2}{dp^2}h(\lambda p, m, n, \beta) = \frac{mn\exp(-(1+\beta)mnp)\left(\frac{1}{1-p}\right)^{mn}(1+mn(p-\beta(1-p))^2)}{(1-p)^2}.$$

We proceed to compute the first derivative to the end of finding a minimizer:

$$\frac{d}{dp}h(\lambda p, m, n, \beta) = \frac{mn\exp(-(1+\beta)mnp)\left(\frac{1}{1-p}\right)^{mn}(p-\beta(1-p))}{1-p}.$$

Giving the minimizer at $p^* = \frac{\beta}{1+\beta}$.

Hence,

$$\begin{aligned}\mathbb{P}\left(c^T\bar{X} \geq (1+\beta)\mu\right) &\leq (1+\beta)^{mn}\exp(-\beta mn)\\ &= \exp[-(\beta - \ln(1+\beta))mn]\end{aligned}$$

The other result is derived similarly, taking care to pay attention to signs (in that case $r < 0$). ■

Note the exponential dependence on both the number of samples m and the dimension n. The improvement of the bound as the dimension increases can be explained by the fact that the deviation from the mean is described in the relative sense.

While we may be tempted to take advantage of this and use $O(1)$ samples, it should be noted that at least n points are needed in order to obtain a positive definite approximation of the covariance matrix. (If K lies in an affine subspace, we may always find a basis for the relative interior of K.) In our algorithm, we use at least `1.5n` points, which at $\beta = 0.05$ gives a probability of deviation of at most `0.0107` for $n = 50$ and increases rapidly (it is at most `1.45e-3` for $n = 60$, and at most `1.37e-4` for $n = 70$).

Supposing that the optimal solution set is a singleton, as the temperature gets lower, the feasible set is "practically" a cone as most of the probability mass is concentrated near that point (the "vertex"). In that case, as we converge to an optimal solution, Proposition 2.13 gives us a good basis for selecting the number of sample points m.

As a minor extension of these ideas, we might wish to consider different functions of our sample points. A possibly useful quantity would be the means of the $q \leq m$ points with the best objective values.

Proposition 2.14. *Under the assumptions of Proposition 2.13, let* $\mu^{[q]} = \frac{1}{q}\sum_{j=1}^{q} X_1^{(k_j)}$ *where the list* $X_1^{(k_1)}, X_1^{(k_2)}, \ldots, X_1^{(k_m)}$ *is arranged in ascending order. Then*

$$\mathbb{P}\left(\mu^{[q]} \leq (1-\beta)\mu\right) \geq \sum_{r=q}^{m} \binom{m}{r} (1-\bar{p}_\beta)^r (\bar{p}_\beta)^{m-r}$$

where

$$\begin{aligned} \bar{p}_\beta &= \mathbb{P}\left(c^T X \geq (1-\beta)\mu\right) \\ &= \left[1 + \sum_{k=1}^{n-1} \frac{((1-\beta)n)^k}{k!}\right] \exp(-(1-\beta)n) \end{aligned}$$

and equality holds at $q = 1$.

Proof. Firstly, the probability that the q points with the smallest objective value all have objective value less than $(1-\beta)\mu$ is $\sum_{r=q}^{m}\binom{m}{r}(1-p_\beta)^r(p_\beta)^{m-r}$. This is thus a lower bound on the probability in question.

To complete the proof,

$$\begin{aligned}
\mathbb{P}\left(c^T X \geq (1-\beta)\mu\right) &= \int_{(1-\beta)\mu}^{\infty} \frac{\lambda^n x^{n-1} e^{-\lambda x}}{(n-1)!}\,dx \\
&= \int_{(1-\beta)n}^{\infty} \frac{x^{n-1}e^{-x}}{(n-1)!}\,dx \\
&= \left[\frac{-x^{n-1}e^{-x}}{(n-1)!}\right]_{(1-\beta)n}^{\infty} + \int_{(1-\beta)n}^{\infty} \frac{x^{n-2}e^{-x}}{(n-2)!}\,dx \\
&= \frac{((1-\beta)n)^{n-1}e^{-(1-\beta)n}}{(n-1)!} + \int_{(1-\beta)n}^{\infty} \frac{x^{n-2}e^{-x}}{(n-2)!}\,dx \\
&= \left[1 + \sum_{k=1}^{n-1} \frac{((1-\beta)n)^k}{k!}\right] e^{-(1-\beta)n}.
\end{aligned}$$

The assertion that equality holds at $q=1$ is obvious. ■

Admittedly, Proposition 2.14 appears to be a weak bound, but the values involved are easily computable. If one were to consider p_β, one would note that it is nothing but the truncated power series expansion of $e^{(1-\beta)n}$ as a fraction of its actual value. Clearly as the dimension n increases, for a given value of $\beta \leq 1$, that fraction increases towards 1.

Consider the following improvement on Proposition 2.14. In Proposition 2.14, we considered only two possible events for each sample: $\{X_1 \leq (1-\beta)\mu\}$ and its complement $\{X_1 > (1-\beta)\mu\}$. (Note that an event is a set of outcomes.) Now let us impose a richer structure on the set of events. Let $k_\delta > 1$ and $\delta = (1-\beta)/k_\delta$. We now consider following partition of the outcome space into disjoint events:

$$\{X_1/\mu \in [0,\delta]\}, \{X_1/\mu \in (0,2\delta]\}, \ldots, \{X_1/\mu \in (0,(2k_\delta - 1)\delta]\}, \{X_1/\mu > (2k_\delta - 1)\delta\}.$$

We then need to compute the probabilities of each of the outcomes listed.

Lemma 2.15. *Under the assumptions of Proposition 2.13,*

$$\mathbb{P}\left(c^T X \in [\alpha\mu, \beta\mu]\right) = p_\alpha - p_\beta$$

where $0 \leq \alpha < \beta$ *and*

$$\begin{aligned} p_\theta &= \mathbb{P}\left(c^T X \geq \theta\mu\right) \\ &= \left[1 + \sum_{k=1}^{n-1} \frac{(\theta n)^k}{k!}\right] e^{-\theta n}. \end{aligned}$$

Proof. Compute p_θ by repeating the computation in the proof of Proposition 2.14 replacing $(1-\beta)$ with θ. ∎

Let the multinomial coefficient $\binom{m}{r_1, r_2, \ldots, r_k} := \frac{m!}{\prod_{j=1}^{k} r_j!}$ (where $\sum_{j=1}^{k} r_j = m$). Now, if we wish for the mean of q numbers to be less than 0, if the largest k of those numbers lie in the interval $[0, \beta]$, it is then sufficient that another k of those numbers be less than or equal to $-\beta$. Note also that the event that less than q samples have $X_1 < \mu$ is disjoint from the event that at least q are. The following result follows from the previous discussion.

Proposition 2.16. *Let* $k_\delta > 1$ *and* $\delta = (1-\beta)/k_\delta$. *For* $1 - k_\delta \leq j \leq k_\delta - 1$, *let* $p_j = \mathbb{P}\left(c^T X \in [(j + k_\delta - 1)\delta\mu, (j + k_\delta)\delta\mu]\right)$, *and* $q = \mathbb{P}\left(c^T X > (2k_\delta - 1)\delta\mu\right) = 1 - \sum_{j=1-k_\delta}^{k_\delta - 1} p_j$. *Under the assumptions of Proposition 2.13,*

$$\mathbb{P}\left(\mu^{[q]} \leq (1-\beta)\mu\right) \geq P_q + \sum_{S} M(m, r) \left(\prod_{j=1-k_\delta}^{k_\delta - 1} p_j^{r_j}\right) q^{m - \sum_{j=1-k_\delta}^{k_\delta - 1} r_j}$$

where P_q *is the right hand side term in the bound of proposition 2.14,*

$$S := \left\{ (r_{1-k_\delta}, \ldots, r_{k_\delta - 1}) : \sum_{j=1-k_\delta}^{k_\delta - 1} r_j \leq m, \sum_{j=1-k_\delta}^{0} r_j < q, \sum_{j=1-k_\delta}^{k_\delta - 1} j r_j \leq 0 \right\}$$

or a subset thereof, and

$$M(m,r) = \binom{m}{r_{1-k_\delta}, \ldots, r_{k_\delta-1}, m - \sum_{j=1-k_\delta}^{k_\delta-1} r_j}.$$

As discussed, the set S represents a subset of the event that $\mu^{[q]} \leq (1-\beta)\mu$. What is at issue now is finding sets S for which we have a strong bound, which is then a problem of enumeration. One could, in principle, extend the number of events further to include more intervals of length δ, but under an Erlang-n distribution, the probability of those events are small.

Chapter 3

Computational Methods

3.1 Components of an Algorithm: Improvements and Heuristics

Suppose one had a plain vanilla exponential annealing algorithm that performed hit-and-run (without transforming the unit sphere), reduced the temperature and repeated this until the temperature was sufficiently low.

In this section, we consider improvements and heuristics that can be made to the above. By rescaling the set to "round" it [7] at each step, one obtains Algorithm 1 from [7]. We consider, in addition, modifying the hit-and-run random walk by something we call "truncation."

Our practical objective is the construction of a robust algorithm that works efficiently for problems of respectably-large dimension. Practical interior point methods reduce the barrier parameter by a dimension-free factor; we would like to do the same for the temperature (as opposed to using a factor of $1 - 1/\sqrt{n}$).

In attempting to achieve this, we recognize that one need not converge to exponential distributions, but rather to distributions that converge in probability to the optimal solution set.

(Note: In all diagrams that follow, "Distance from Boundary (in $\mathbb{R}^n$)" refers to the Euclidean distance of the current candidate solution to the boundary of the set defined by the inequality constraints. This quantity is a lower bound on the distance to the boundary of the set in the smallest affine subspace containing the feasible region.)

(In addition, n refers to the dimension of the problem and n_{affine} refers to the dimension of the affine subspace described by the equality constraints.)

(Note also that for problems for which an initial feasible point was not supplied, relative optimality gaps are not shown in the phase where a feasible point is being sought.)

(In this chapter, we take a slightly different approach from that described in Section 1.2. In attempting to find a feasible point for to (P) of that section, we consider the following problem:

$$(P'') \quad \begin{array}{lrcl} \min\limits_{x,\theta} & \theta & & \\ s.t. & b_e - A_e x - \theta b_e & = & 0 \\ & b_c - A_c x + \theta(y - b_c) & \in & C \end{array}$$

for which $(x, \theta) = (0, 1)$ is strictly feasible for some given $y \in int\, C$. We seek a candidate solution with $\theta < 0$ and use convexity to obtain a strictly feasible point. This is only done for the computations described in this chapter.)

3.1.1 The Next Phase: Where to Begin

In Algorithm 1, the next hit-and-run random walk at a "lower temperature" is initiated from the last iterate of the previous hit-and-run random walk. The reason for this is that the last iterate is approximately distributed (in the sense of the Total Variation distance) with a density proportional to $e^{-c^T x/T_k}$, which is close, in the same sense, to a distribution with a density proportional to $e^{-c^T x/T_{k+1}}$. On the other hand, computationally, we start each random walk at a single point, and it is a necessary practicality to guard against the random walk having to begin at a point too close to the boundary of K as hit-and-run mixes slowly when started near the boundary.

This motivates two strategies: (i) using the mean of the m samples used to approximate the covariance matrix as the new starting point, and (ii) forcing the random walk to avoid the boundary by sampling only on part of the line segment intersecting K (which we will call "truncation"). Computing the sample mean $\bar{x}$ and using that as a starting point results in a point deeper in the interior of K as it is distributed more tightly around the true mean of the current exponential distribution. "Truncation" explicitly does not allow iterates in the random walk to approach the boundary too closely; this will be described in greater detail in Subsection 3.1.3.

Computational experiments suggest that using both strategies in tandem yields better performance in terms of the convergence of the relative optimality gap to 0. Figures 3-1 to 3-4 show results from solving the following problem:

$$(\text{Cone}_{n,h}) \quad \begin{array}{lll} \min\limits_{x} & -x_1 + \sum_{i=2}^{n} h/nx_i & \\ s.t. & x_1 & \geq \ 0 \\ & x_1 - hx_i & \leq \ h \ \ (i = 2, 3, \ldots, n) \\ & x_1 + hx_i & \leq \ h \ \ (i = 2, 3, \ldots, n) \end{array}$$

for which the optimal solution value is $-h$. Here we use $n = 20$, $h = 10$, and supply an initial strictly feasible solution $(h/n, 0, 0, \ldots, 0)$. Figures 3-5 to 3-8 show results for solving the (primal of the) linear program AFIRO (from the NETLIB suite). Cases where the mean is used will be labelled "Mean Used", and "Single Trajectory" otherwise. Likewise, the use of truncation will be indicated with "Truncation Used", and "No Truncation" otherwise. To compare the results fairly (since termination happens later under truncation), one needs to compare the relative optimality gap at a given iteration.

What these tests indicate are that the use of the sample mean and truncation provide marginal benefits in terms of convergence of the relative optimality gap to 0, but when used in tandem, a tremendous improvement is observed. For $\text{Cone}_{n,h}$, a single walk achieves a relative optimality gap of about 10^{-5} by iteration 40, but with use of both the mean and truncation at the same iteration number, one achieves a relative optimality gap of less than 10^{-8}. For AFIRO, a single walk achieves a relative optimality gap of about 5×10^{-7} by iteration 50, but with use of both the mean and truncation at the same iteration number, one achieves a relative optimality gap of less than 10^{-9}.

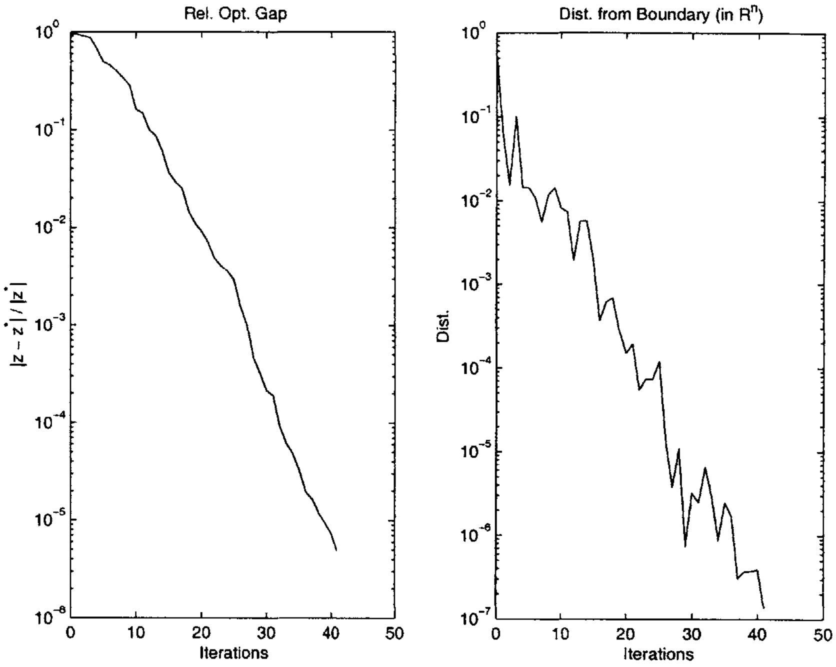

Figure 3-1: $\text{Cone}_{20,10}$ (Single Trajectory, No Truncation)

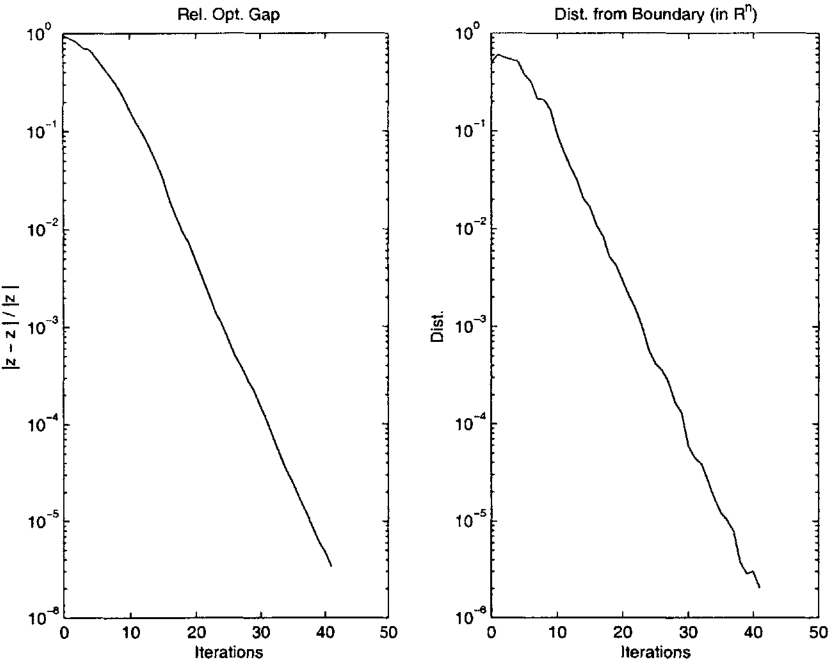

Figure 3-2: $\text{Cone}_{20,10}$ (Mean Used, No Truncation)

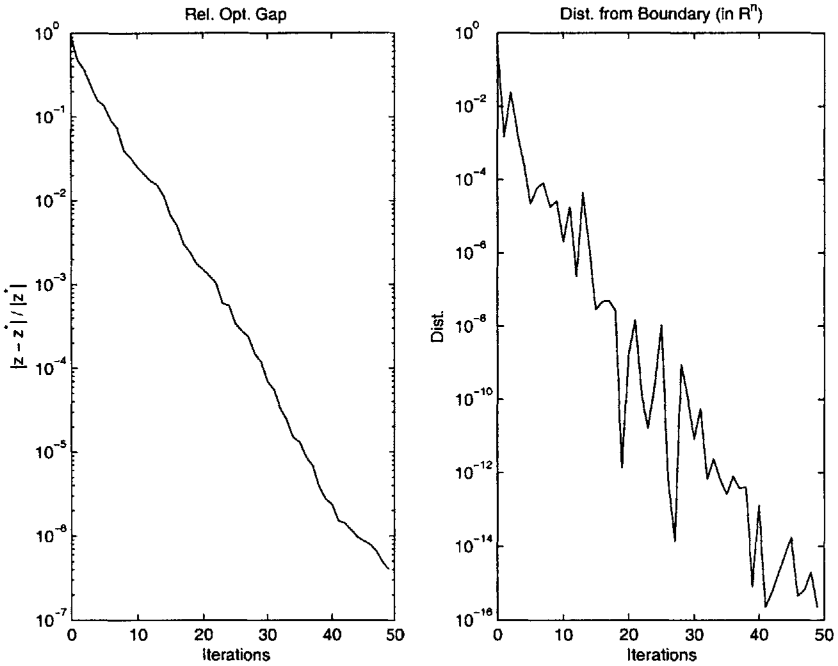

Figure 3-3: $\text{Cone}_{20,10}$ (Single Trajectory, Truncation Used)

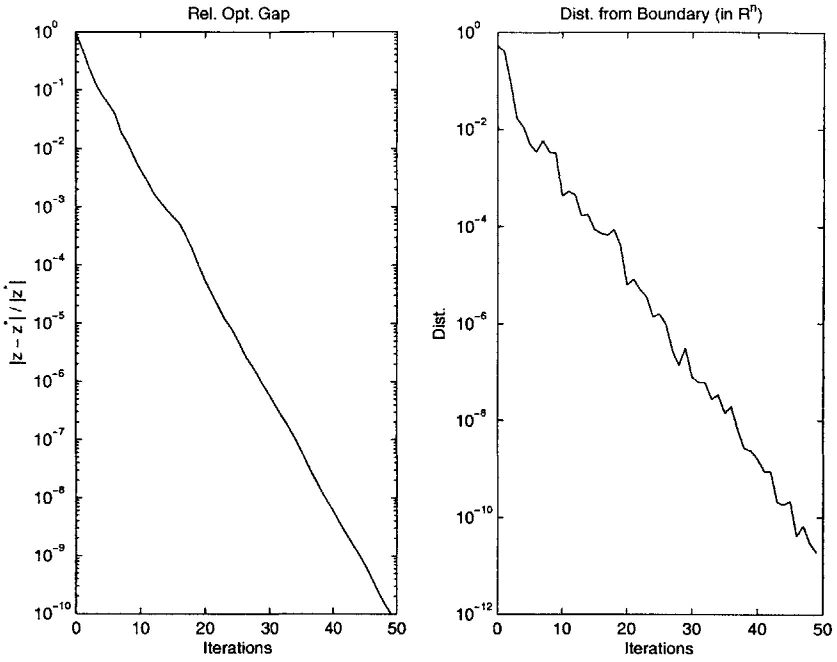

Figure 3-4: $\text{Cone}_{20,10}$ (Mean Used, Truncation Used)

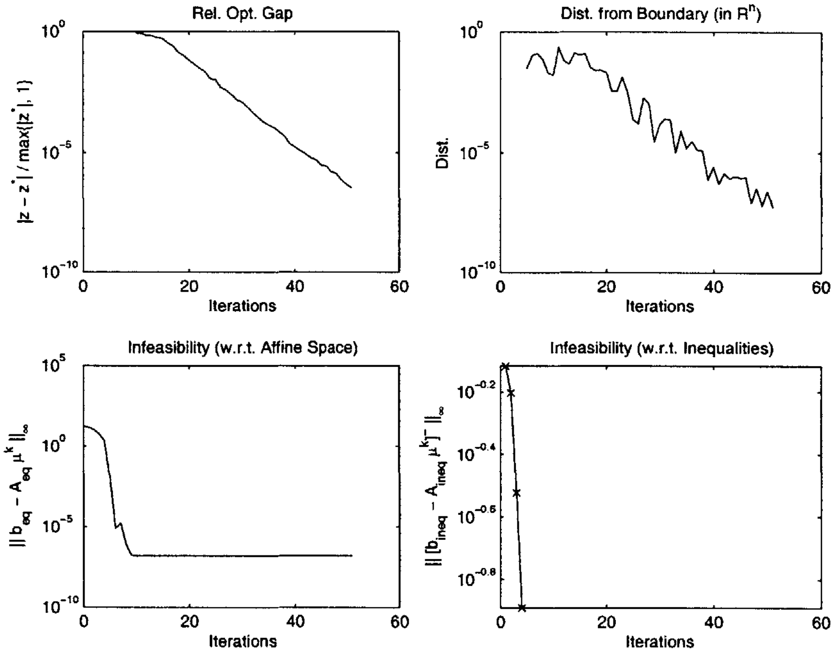

Figure 3-5: AFIRO (Single Trajectory, No Truncation)

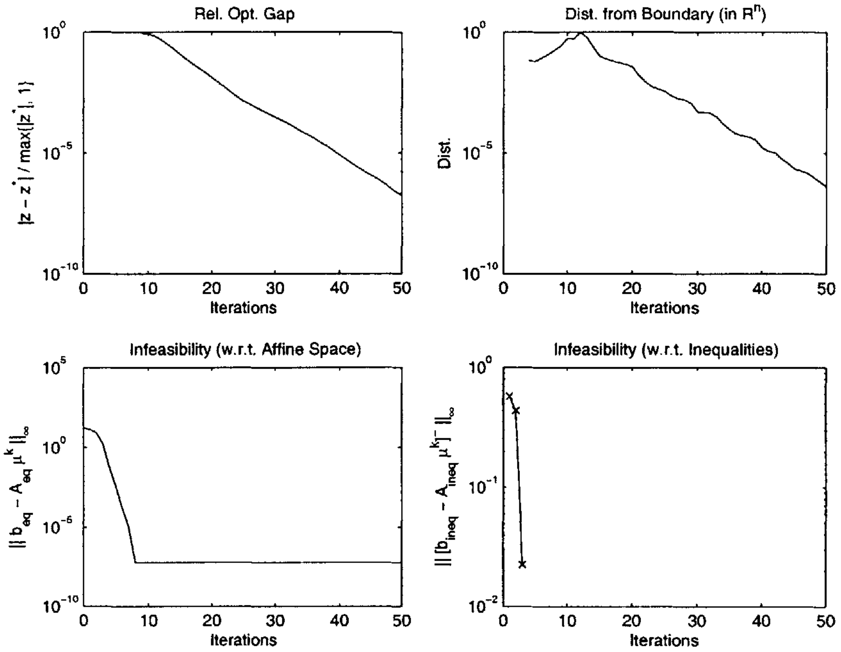

Figure 3-6: AFIRO (Mean Used, No Truncation)

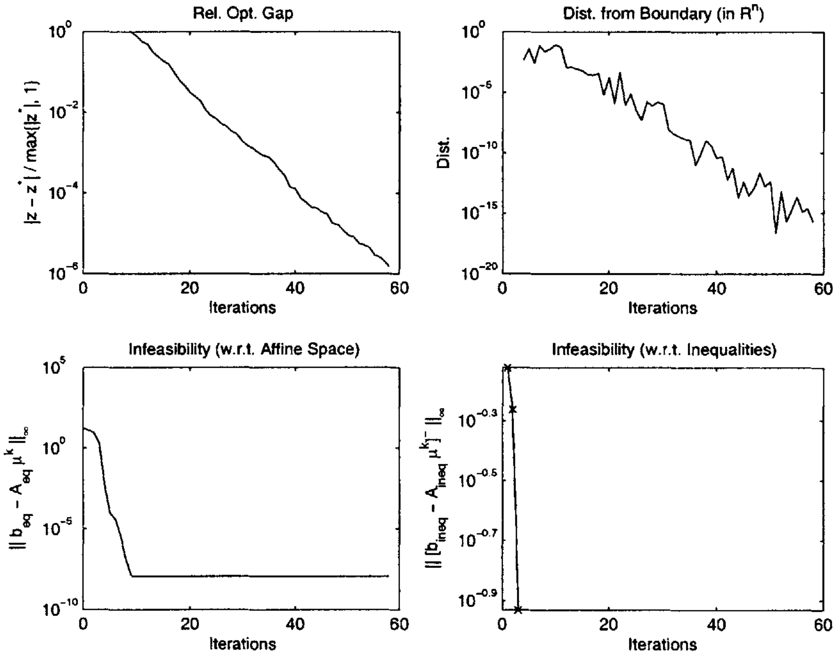

Figure 3-7: AFIRO (Single Trajectory, Truncation Used)

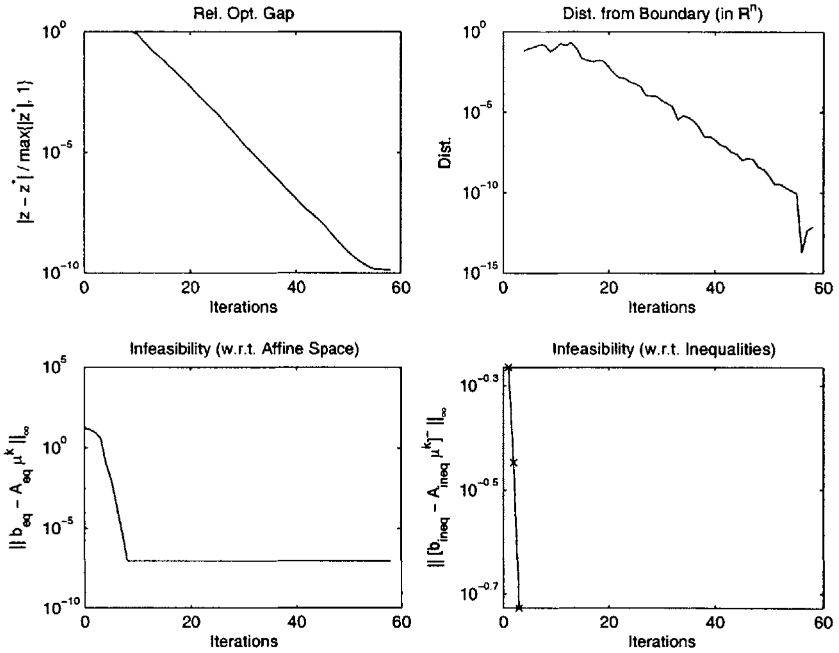

Figure 3-8: AFIRO (Mean Used, Truncation Used)

3.1.2 Rescaling towards Isotropicity

The first modification to the "plain vanilla" exponential annealing method just described is rescaling the feasible set. This is exactly what is done in Algorithm 1 described by [7]. The reasoning behind this is that convex bodies can be arbitrarily thin along certain directions relative to others. This may result in arbitrarily small steps in the hit-and-run random walk for all but an arbitrarily small set of directions (as a fraction of the volume of the unit sphere).

The covariance matrix over K is given by $\Sigma_K = \mathbb{E}_K\left[(X - \mathbb{E}_K[X])(X - \mathbb{E}_K[X])^T\right]$, and defines an ellipsoid $E := \left\{x : (x - \mathbb{E}_K[X])^T\Sigma^{-1}(x - \mathbb{E}_K[X]) \leq 1\right\}$. E can be said to locally approximate K in the sense that if K were such that the mean and its covariance were the origin and the identity respectively (this can be achieve by an appropriate affine transformation), E would be contained in the convex hull of K and $-K$. (Note that affine transformations preserve inclusion.) This is shown here:

Proposition 3.1. *Given a density f supported on a convex set K with mean 0 and covariance matrix I, for all $x \in E := \{x : x^Tx \leq 1\}$, x is contained in the convex hull of K and $-K$.*

Proof. Suppose for the sake of obtaining a contradiction that $v \in E$ and $v \notin conv\{K, -K\}$. We first note that v is strictly separated from the symmetric convex set $conv\{K, -K\}$ in the sense that there exists $u : \|u\| = 1$ such that $0 < \alpha_v = v^Tu$ and for all $x \in conv\{K, -K\}$, $-(\alpha_v - \epsilon) \leq x^Tu \leq \alpha_v - \epsilon$ (for $\epsilon > 0$ small enough). Note that $1 \geq v^Tv \geq \alpha_v^2$, and that $\alpha_v u$ is also separated from $conv\{K, -K\}$ in the same way.

Consider the marginal density $\tilde{f}$ where $\tilde{f}(\alpha) := \int_{\{x \in K : u^Tx = \alpha\}} f(x)\, dx$. Now,

$$\int_{-(\alpha_v - \epsilon)}^{\alpha_v - \epsilon} \alpha^2 \tilde{f}(x)\, d\alpha < \alpha_v^2$$

and

$$\begin{aligned}\int_{-(\alpha_v-\epsilon)}^{\alpha_v-\epsilon} \alpha^2 \tilde{f}(x)\, d\alpha &= \int_K (u^T x)^2 f(x)\, dx \\ &= u^T \left(\int_K x x^T f(x)\, dx \right) u \\ &= u^T u \\ &= 1.\end{aligned}$$

This gives

$$1 < \alpha_v^2 \leq v^T v \leq 1$$

which gives the desired contradiction and completes the proof. ■

(At this point, it is unknown to the author whether the stronger assumption of logconcavity of the density function would yield the inclusion $E \subseteq K$.)

(Note: to compute the sample covariance matrix given m samples $X_1, X_2, \ldots, X_m$ with mean μ, the sample covariance matrix is given by $\Sigma_{\text{sample}} = \frac{1}{m}\sum_{j=1}^{m} (X_j - \mu)(X_j - \mu)^T$.)

Proceeding on the original line of inquiry, we have the factorization $\Sigma = VV^T$, which ensures that for all unit vectors d, $Vd + \mathbb{E}_K[X]$ lies on the surface of E. This also means that the set $K' := \{x : Vx \in K\}$ has the identity matrix as its covariance as shown below:

$$\begin{aligned}\Sigma_{K'} &= \mathbb{E}_{K'}\left[(X - \mathbb{E}_{K'}[X])(X - \mathbb{E}_{K'}[X])^T\right] \\ &= \mathbb{E}_K\left[(V^{-1}X - \mathbb{E}_K\left[V^{-1}X\right])(V^{-1}X - \mathbb{E}_K\left[V^{-1}X\right])^T\right] \\ &= V^{-1}\mathbb{E}_K\left[(X - \mathbb{E}_K[X])(X - \mathbb{E}_K[X])^T\right] V^{-T} \\ &= I.\end{aligned}$$

Rescaling allows more appropriate selection of directions for the hit-and-run random walk and makes K "look" like a set with the identity as its covariance. This is illustrated in Figure 3-9.

Without loss of generality, we may assume that our problem has no equality

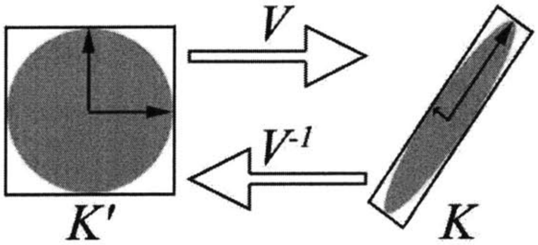

Figure 3-9: Rescaling Illustrated

constraints, since we can find a feasible point and a basis for the affine space which the feasible set lies in and modify the conic constraints accordingly. Now, if d is a random vector on the unit sphere chosen to define the line in the hit-and-run random walk, the search direction under linear transformation becomes Vd (assume V non-singular), and the exponential distribution on the line (the restriction of the distribution on K to that line) is defined by the parameter $c^T Vd$ (suppose d is chosen such that this is positive).

Now, $K := \{x : b - Ax \in C\}$ and $K' := \{x : Vx \in K\} = \{x : b - AVx \in C\}$. So given $x_0 \in K$, one seeks the maximum and minimum values of α such that $b - AV(V^{-1}x) - AV(\alpha d) \in C$. From this point of view, we seem to be working within K' which has the identity as its covariance, using the cost vector $V^T c$.

3.1.3 Truncation

Interior Point Methods (IPMs) for convex optimization are iterative methods that work on the basis of maintaining strict feasibility of iterates (staying away from the boundary of the feasible set). Each point in the feasible set has an associated inner product (and induced local norm) and it can be shown that a ball of radius 1 in the local norm is always contained in the feasible set, giving a means of ensuring the strict feasibility of iterates.

Maintaining the strict feasibility of iterates in the hit-and-run random walk is implicit, since the probability of any hit-and-run iterate lying on the boundary is 0. On the other hand, as sample points are being evaluated (by performing a random walk), one usually observes "stalling" in the sense of very little further movement after an intermediate point (during the random walk) gets close to a boundary with a normal making a positive inner product with the cost vector (see Figure 3-10).

This is intuitively true in the sense that near a boundary (which is locally a hyperplane), the volume of the set of descent directions on the unit sphere on which "reasonable progress" can be made can be small compared to the volume of (half) the unit sphere itself. The fraction in question depends on the inner product $c^T a$ where c is the cost vector and a is the normal of the boundary pointing into the set (at the point on the boundary closest to the current iterate). Supposing that $\|c\| = \|a\| = 1$, the closer $c^T a$ is to 1, the smaller this fraction can be. This is illustrated in Figure 3-10.

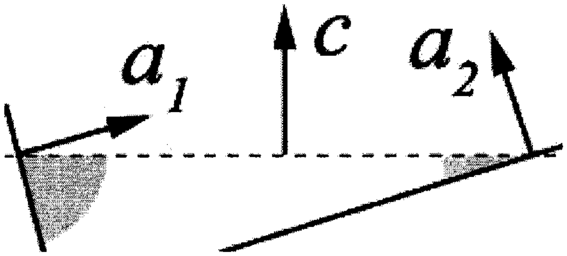

Figure 3-10: The Effect of the Boundary

What results from starting near a boundary is reduced accuracy at a given phase and hence an increased number of phases. (Recall that a phase, as described in Algorithm 1, involves a change in the current exponential distribution to one with a lower temperature parameter.) This fall in accuracy at a given phase is due to a low probability of hit-and-run iterates leaving such subsets of K containing parts of the boundary resulting a slower progress. To alleviate this problem, we employ a heuristic known as "truncation" in order to avoid subsets of K close to the boundary.

The hit-and-run random walk with an underlying exponential distribution is simple to perform due to the memoryless property of the exponential distribution, allowing easy sampling on the restriction of the distribution to any line l intersecting K. It is just as easy to sample on a subset of $l \cap K$ omitting a segment of the line with the lowest cost with a total measure (or probability) of p for some $p \in [0, 1)$. This enables the walk to avoid the boundary.

Sampling on an interval or a halfline with a density proportional to $e^{-c^T(x+\alpha d)/T}$ is in effect equivalent to sampling with a density proportional to $e^{-\lambda\alpha}$ on $\alpha \in [\alpha_{min}, \alpha_{max})$, where $\lambda = c^T d/T$ (α_{max} may be ∞, but α_{min} must be finite by assumption). If, instead, we draw samples from the subset of $[\alpha_{min}, \alpha_{max})$, $[\alpha_p, \alpha_{max})$ with probability mass $1-p$, iterates are forced to avoid the boundary to the extent that the temperature is high. Note that, given p, as $T \downarrow 0$, $\alpha_p \downarrow 0$. While the reduction in temperature is similar in spirit to the effect of changing the barrier parameter in IPMs, truncation is analogous to step-size rules.

To give an example of truncation, suppose $c^T d/T = 1$ and $[\alpha_{min}, \alpha_{max}) = [-1, \infty)$. Let $p = 0.5$. We then have $\alpha_p = -1 + \log(2)$ and we sample on $[\alpha_p, \infty)$ with the density $f(x) = \frac{1}{1-p} e^{-1-x}$. This is illustrated in Figure 3-11.

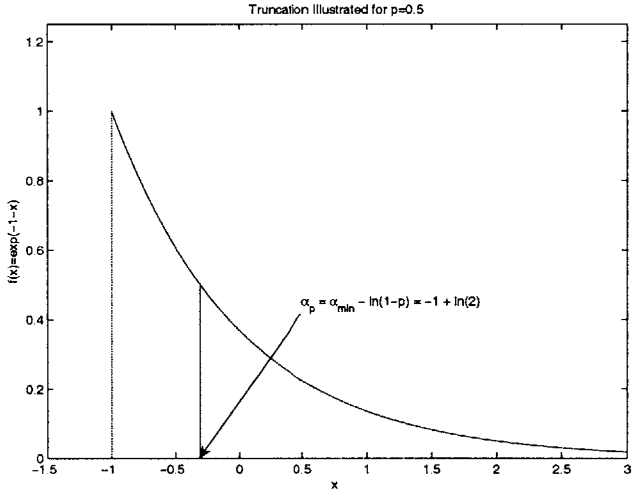

Figure 3-11: Truncation Illustrated

The chief question one might ask is what distribution do the iterates converge to. It is unclear. However, looking at the new Markov chain generated, one notes that there is a "layer" of transient states on the boundary. Also, as the temperature goes to 0, this "layer" thins and vanishes.

In informal tests on the cone $\{x : \max_{k=2,3,\ldots,n} 10|x_k| \leq x_1\}$ with an objective vector $(1, 0, \ldots)$, it was found that "truncation" resulted in improved performance in iteration count. In this problem, the number of phases is the number required to get within 1e-6 of the optimal objective value of 0. Figures 3-12 to 3-15 illustrate this.

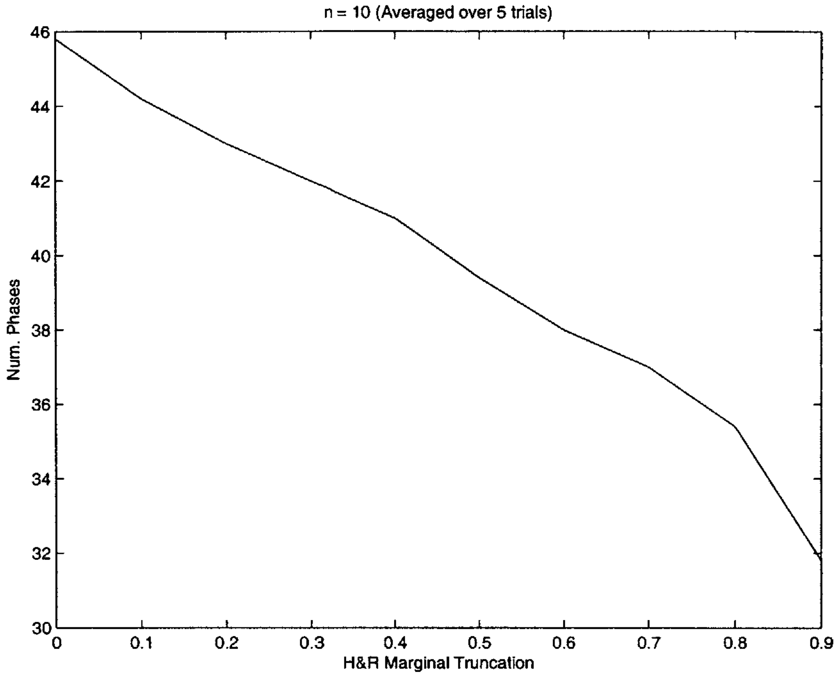

Figure 3-12: Phase Count on a Cone versus p ($n = 10$)

The results suggest that it is good to keep the iterates of the random walk away from the boundary so subsequent iterates are able to make greater progress, cutting down on steps that make little progress (reducing "wasted" computation).

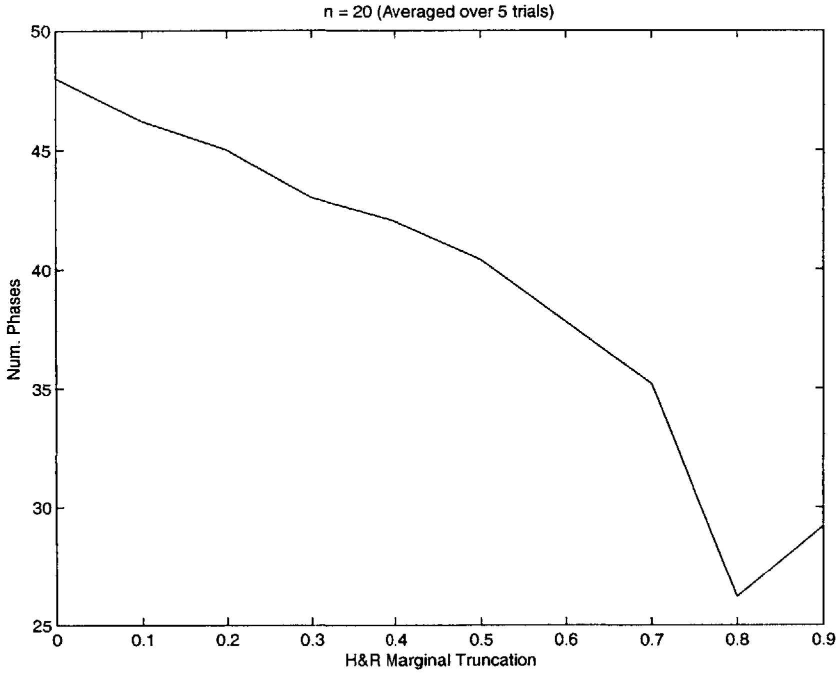

Figure 3-13: Phase Count on a Cone versus p ($n = 20$)

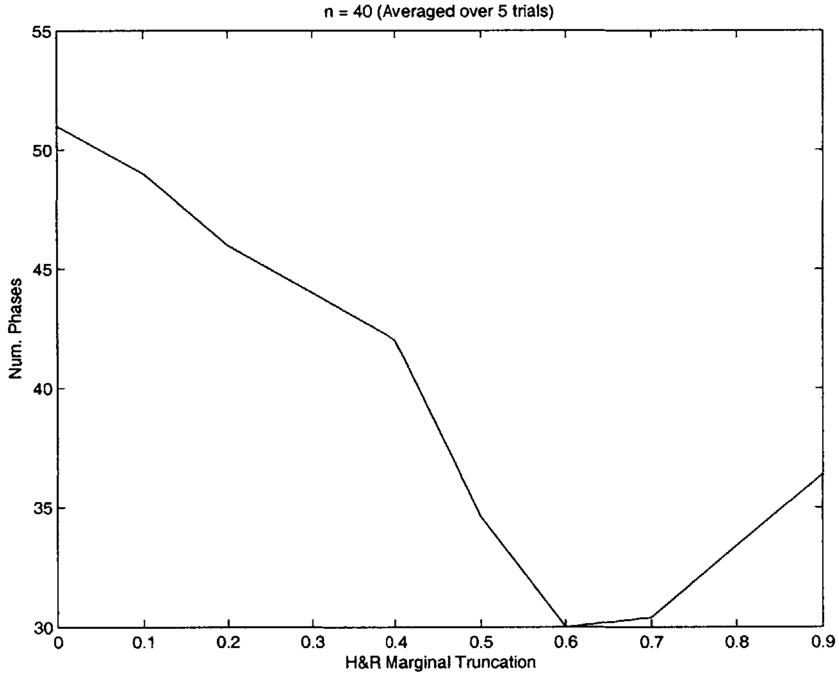

Figure 3-14: Phase Count on a Cone versus p ($n = 40$)

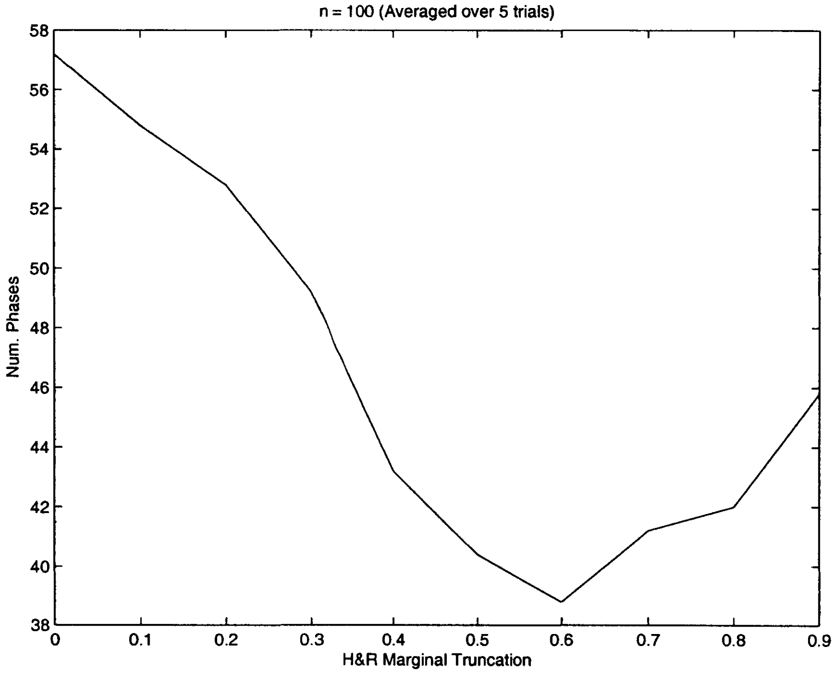

Figure 3-15: Phase Count on a Cone versus p ($n = 100$)

Schedules for p

To gain a sense of this, suppose we have a density of $\frac{1}{T}e^{-x/T}$ on $[0,\infty)$. It is straightforward to derive that the interval $\left[0, T\log\frac{1}{1-p}\right]$ has probability mass p for $p \in [0,1)$, and the restricted distribution on the interval $\left[T\log\frac{1}{1-p}, \infty\right)$ has a mean of $T\left(1+\log\frac{1}{1-p}\right)$ which is an increase in the mean by a multiplicative factor of $1+\log\frac{1}{1-p}$. As previously mentioned, as $T \downarrow 0$, the mean goes to zero as well.

Now it stands to reason that at higher values of T, we might wish to use a lower value of p since iterates are naturally kept from the boundary during the sampling process and only use a higher value when T gets low.

Suppose we fix $\bar{p} \in [0,1]$ and have a non-increasing sequence $\{T_k\}_{k=1}$ with $T_1 = 1$. Consider a corresponding sequence of "truncation levels" $p_k = \min\{1 - T_k, \bar{p}\}$. Then the mean goes to 0 as $T_k \downarrow 0$ (even if $p = 1$, which gives a mean of $T_k(1 - \log T_k)$). In our tests, $T_k = 0.7^{k-1}$.

We now present tests of some schedules on problems from the NETLIB suite. All the following illustrations are single runs, but are representative (since averages would be rather meaningless). Note that the runs are terminated once a relative accuracy of 1e-6 is achieved.

We first present a schedule with $\bar{p} = 0$ which implies no truncation is used (this is denoted "No Truncation"). We then present a schedule with $\bar{p} = 1$ (denote this Schedule 1). We then present a schedule with $\bar{p} = 0.7$ (denote this Schedule 2).

The results for "No Truncation" are shown in Figures 3-16 to 3-18. The results for "Schedule 1" are shown in Figures 3-19 to 3-21. The results for "Schedule 2" are shown in Figures 3-22 to 3-24.

To tabulate the results:

	Iteration Count		
Schedule	AFIRO	SHARE2B	STOCFOR1
No Truncation ($\bar{p} = 0$)	88	94	93
Schedule 1 ($\bar{p} = 1$)	36	54	69
Schedule 2 ($\bar{p} = 0.7$)	31	45	58

Table 3.1: Number of Phases Required with Various Schedules

Considering the results where no truncation is used, we find that truncation aids in solving the feasibility problem. Also, without truncation, it appears that finding a point that satisfies the equality constraints takes more phases than finding one that satisfies the inequality constraints.

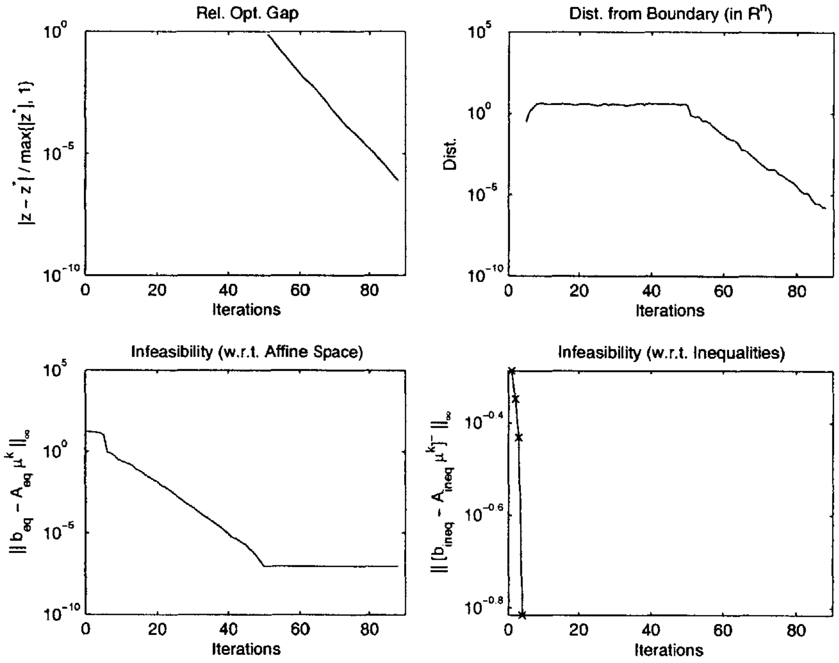

Figure 3-16: No Truncation: NETLIB Problem AFIRO ($n = 32$, $n_{\text{affine}} = 24$)

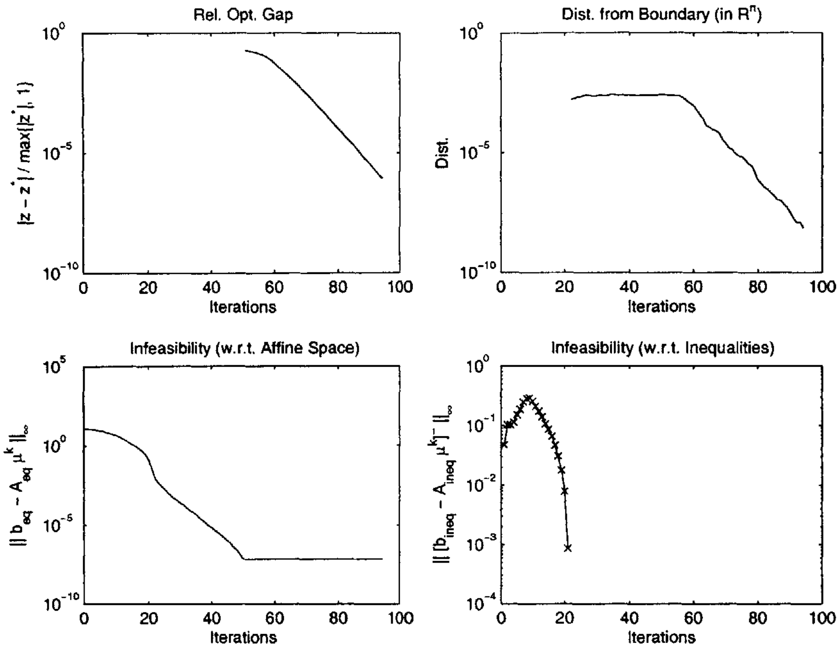

Figure 3-17: No Truncation: NETLIB Problem SHARE2B ($n = 79$, $n_{\text{affine}} = 66$)

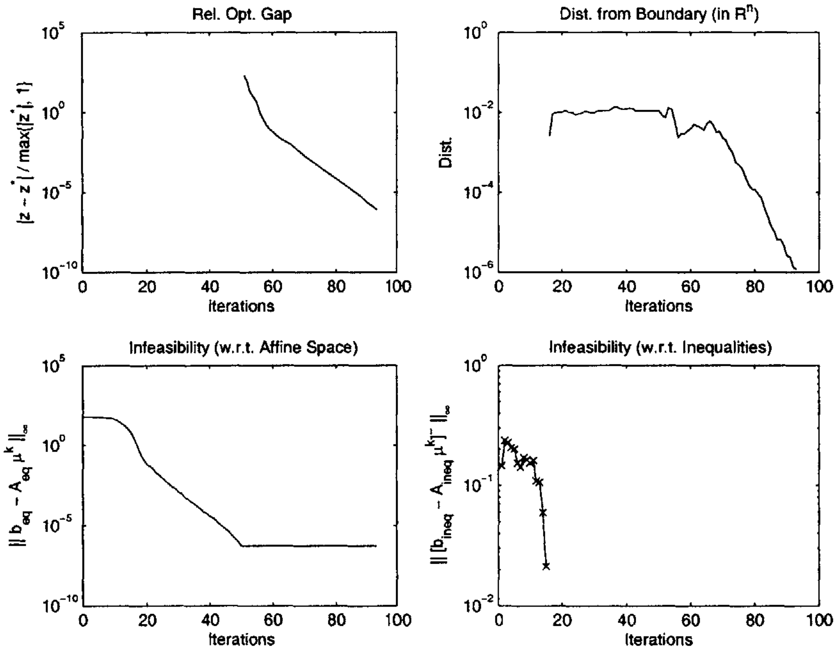

Figure 3-18: No Truncation: NETLIB Problem STOCFOR1 ($n = 111$, $n_{\text{affine}} = 48$)

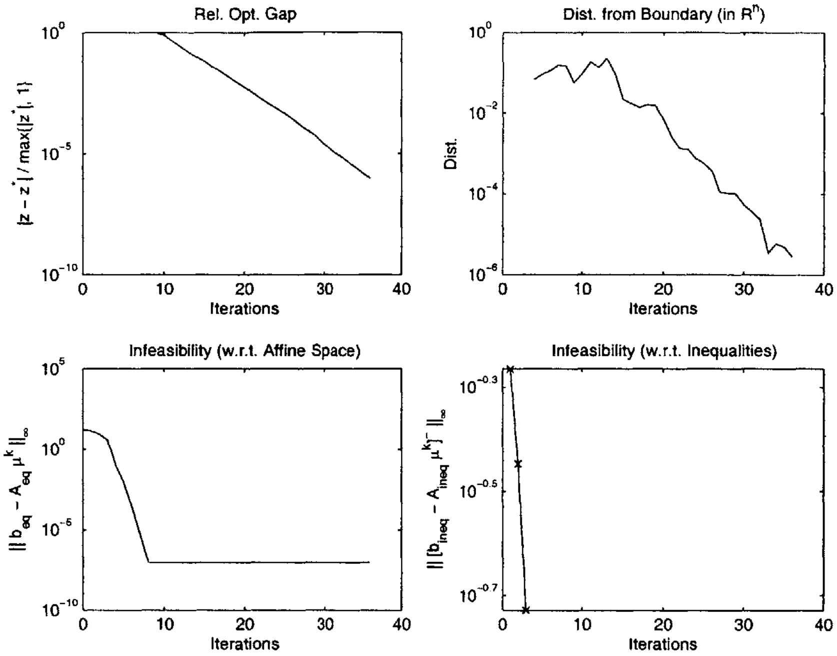

Figure 3-19: Schedule 1: NETLIB Problem AFIRO ($n = 32$, $n_{\text{affine}} = 24$)

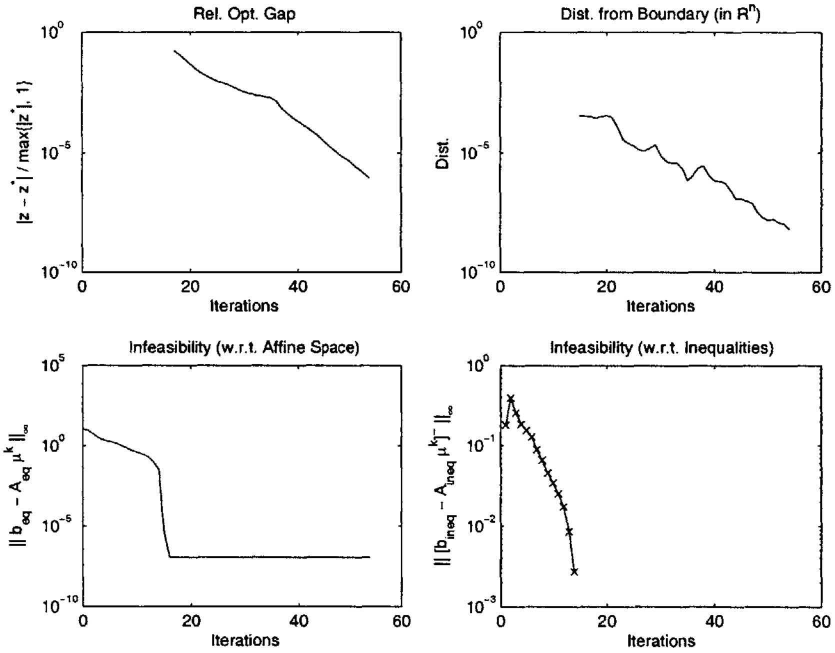

Figure 3-20: Schedule 1: NETLIB Problem SHARE2B ($n = 79$, $n_{\text{affine}} = 66$)

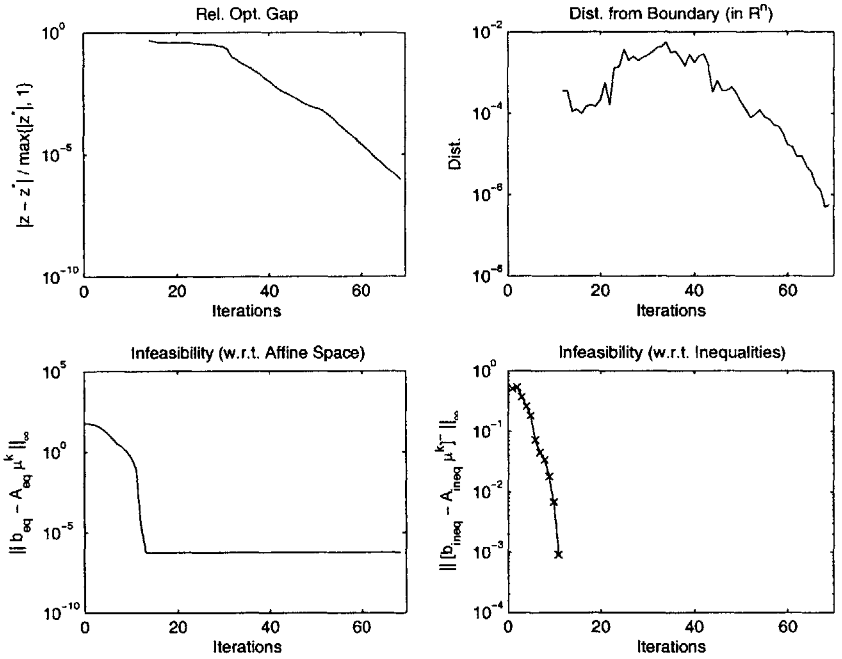

Figure 3-21: Schedule 1: NETLIB Problem STOCFOR1 ($n = 111$, $n_{\text{affine}} = 48$)

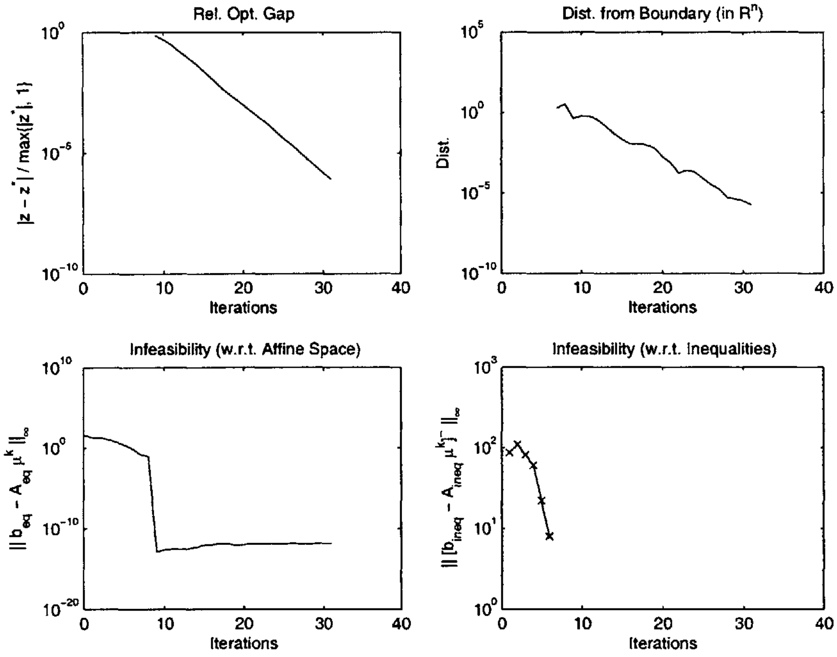

Figure 3-22: Schedule 2: NETLIB Problem AFIRO ($n = 32$, $n_{\text{affine}} = 24$)

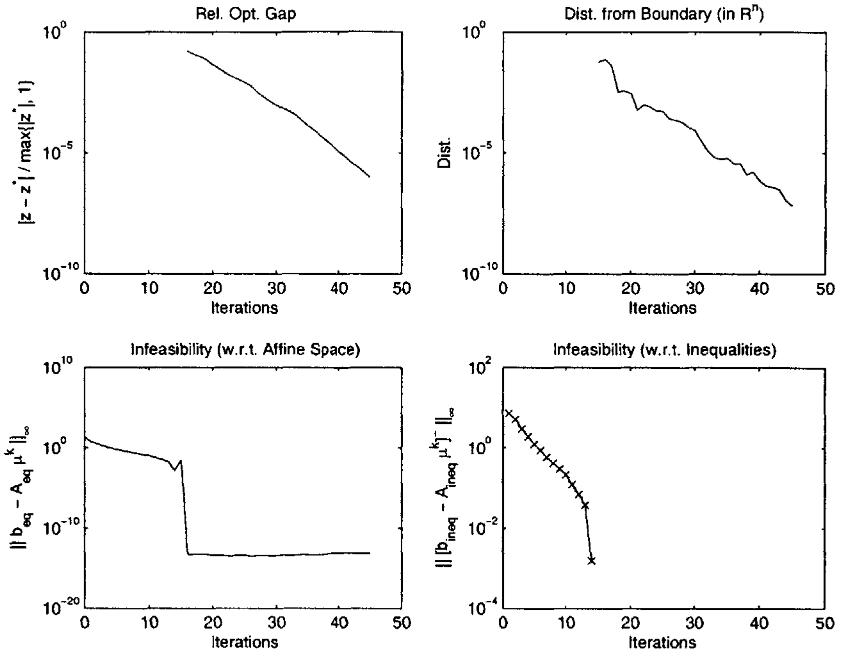

Figure 3-23: Schedule 2: NETLIB Problem SHARE2B ($n = 79$, $n_{\text{affine}} = 66$)

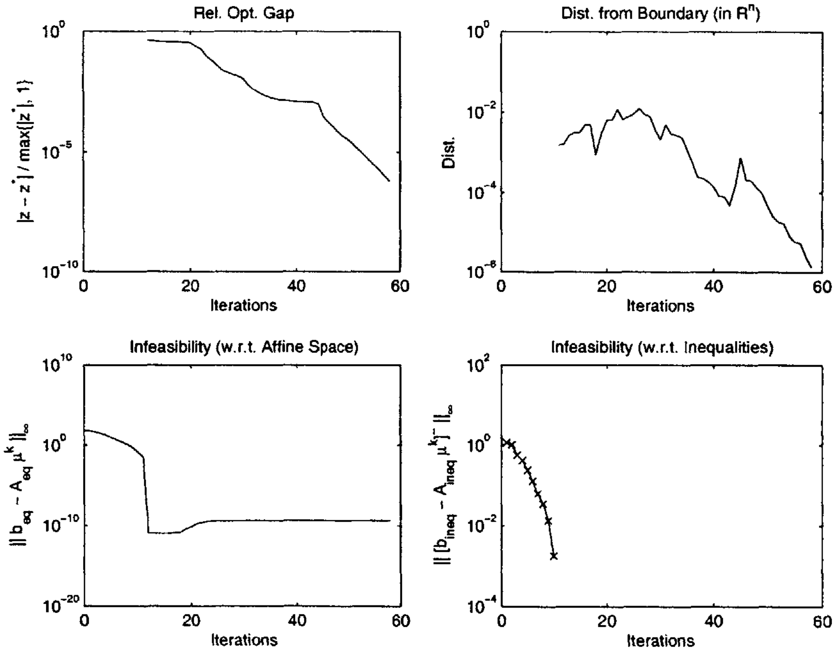

Figure 3-24: Schedule 2: NETLIB Problem STOCFOR1 ($n = 111$, $n_{\text{affine}} = 48$)

3.2 An Algorithm

3.2.1 Exponential Annealing for Conic Optimization

Here we present an algorithm using parts of what was previously described in the chapter. We name the algorithm “Exponential Annealing” not to expropriate the work of Kalai and Vempala, but for the reason that it sounds more appropriate.

```
Input  : $n \in \mathbb{N}$ (dimension)
       : $\mathcal{O}_K : \mathbb{R}^n \to \{0,1\}$ (membership oracle for convex set $K$)
       : $c \in \mathbb{R}^n$ (direction of minimization, $\|c\| = 1$)
       : $\rho \in (0,1)$ (factor by which to reduce temperature)
       : $X_{init} \in K$ (starting point)
       : $k \in \mathbb{N}$ (number of steps per walk)
       : $N \in \mathbb{N}$ (number of samples for rounding)
       : $\epsilon > 0$ (desired accuracy)
       : $\mathcal{P} : \mathbb{R}_+ \to [0,1)$ (truncation schedule)
Output: $X_{out}$ (candidate optimal solution)

$\mu_0 \leftarrow X_{init}$;
$V_0 \leftarrow I$;
$i \leftarrow 0$;
repeat
    $i \leftarrow i + 1$;
    $T_i \leftarrow \rho^{i-1}$ ;      /* reduce temperature independent of dimension */
    Sample:
    for $j \leftarrow 1$ to $N$ do
        $X^i_j \leftarrow$ hit-and-run($e^{-c^T x/T_i}, \mathcal{O}_K, V_{i-1}, \mu_{i-1}, k, \mathcal{P}(T_i)$)
    end
    $\mu_i \leftarrow (1/N)\sum_{j=1}^N X^i_j$
    $V_i \leftarrow (1/N)\sum_{j=1}^N (X^i_j - \mu_i)(X^i_j - \mu_i)^T$
until $nT_i[1 - \log(1 - \mathcal{P}(T_i))] \le \epsilon$ ;
$X_{out} \leftarrow \mu_i$;
Return $X_{out}$;
```

Algorithm 2: Exponential Annealing

In this algorithm, the `hit-and-run` procedure differs from that in Algorithm 1 in that it accepts an additional parameter which informs the procedure what truncation parameter to use. Naturally, we take advantage of the description of the feasible set to perform hit-and-run more efficiently.

Note also that Algorithm 2 solves only problem (P) of Section 1.2, but not its dual. The solution of the dual problem provides valuable information about the problem itself. In particular, the dual solution describes the impact that perturbing each of the constraints has on the optimal solution value. Hence, the lack of the dual solution in the output of this algorithm may be viewed as an inadequacy.

In Chapter 4, we use $\rho = 0.7$, $\mathcal{P}(T) := \min\{1-T, 0.7\}$, $N = \lceil \max\{3/2\ n, n^{1.25}, 1/2\ n^{1.5}\} \rceil$ and $k = 6n$. (Admittedly, there has been little analysis in the choice of N.)

What is immediately apparent is that Algorithm 2 suffers from a similar computational bottleneck as IPMs. The factoring of the covariance matrix to rescale the feasible set has a similar computational complexity, $O(n^3)$, as solving for the Newton direction, and the covariance matrices are invariably dense.

While it is unnecessary to "redilate" the feasible region at every phase, any discussion of complexity issues must also consider the cones in question and the complexity of computing the interval of allowed step lengths (as described in Section 1.3). This can be expensive, especially in the case of the semidefinite cone (this is clear from the discussion in Section 1.3).

3.2.2 On the Operation Count for Semidefinite Optimization

Much interest in the study of randomized methods arises from the rapid growth in the computational cost of computing the Newton step at each iteration of IPMs. As a rough guide, consider applying a simple "short-step barrier method" to the following semidefinite optimization problem:

$$\text{(SDP)} \qquad \begin{array}{ll} \min\limits_{y} & b^T y \\ s.t. & B - \sum_{i=1}^{n} A_i y_i = S \\ & S \in C \end{array}$$

where B, S, A_i $(i = 1, 2, \ldots, n)$ are $d \times d$ symmetric matrices and C is the semidefinite cone of $d \times d$ symmetric matrices (hence of dimension $m = d(d+1)/2$).

In this elementary analysis, assume that all given matrices are dense. We note that sparsity affects IPMs and exponential annealing in different ways (but we have not studied this). Also, practical IPMs are primal-dual algorithms, and we take the perspective that this analysis presents the cost of a dual step of a dual-only IPM.

The assembly of the Newton system has an operation count with leading term $2n^2d^3$, and its solution has an operation count with leading term $\frac{1}{3}n^3$. (See A.2 for details.) In practice, practical IPMs tend to require "between 25 to 40 iterations", we estimate that practical IPMs have a required operation count of $O(n^2d^3)$ to solve problem (SDP).

We now provide a simple analysis of the required operation count of Algorithm 2 for solving problem (SDP). We note first that the required number of phases is at most $\frac{\log n+\log\frac{1}{\epsilon}+\log D}{\log\frac{1}{\rho}} = O(\log n)$ where D is an upper bound for $1-\log(1-\mathcal{P}(T_i))$.

Suppose C_α is the cost of finding upper and lower limits for α such that for an interior point x, a direction d and a closed convex cone C, $x+\alpha d \in C$. Suppose further that C_{rand} is the cost of sampling on the unit sphere or some linear transformation thereof. Performing `hit-and-run` at each phase costs $Nk(C_\alpha+C_{\text{rand}})$. One may check that $C_{\text{rand}} = O(n^2)$ where the quadratic factor results from the linear transformation of the unit sphere.

Computation of the sample mean and covariance matrix as well as the factorization of the latter has an operation count with leading term $\frac{1}{3}n^3$.

Now, for determining upper and lower limits for step lengths in the semidefinite cone, one can check that $C_\alpha = O(n^3)$ (see Section 1.3).

If $N = O(n)$ and $k = O(n)$, then the operation count requirement of Algorithm 2 would be $O(C_\alpha n^2 \log n)$, giving an operation count requirement of $O(n^5 \log n)$ for Algorithm 2. This is comparable with that of a practical IPM.

On the other hand, when determining upper and lower limits for step lengths using the method of A.1 of Appendix A (which is iterative) and being conservative with respect to minimum step lengths (supposing an ascent direction), one may find that one requires $O(n^2)$ operations in practice instead, giving an operation count requirement of $O(n^4 \log n)$ (bear in mind that one has no guarantee of this).

Chapter 4

Computational Results and Concluding Remarks

4.1 Correlation of Required Phases to Measures of Optimization Problems

In this section, we consider measures of the ease of solving linear optimization problems and semidefinite optimization problems by exponential annealing. To this end, one requires a measure of performance and one of the inherent difficulty of a problem. As a measure of performance, we use the number of phases required to achieve a given relative optimality gap (`1e-6` for problems from the NETLIB suite, and `1e-5` for problems from the SDPLIB suite). As a measure of inherent difficulty, we make use of primal/dual condition measures as first presented in [15] (for linear and semidefinite optimization problems) and certain geometry measures developed in [4] (for semidefinite optimization problems). These measures are tabulated for linear optimization problems from the NETLIB suite in [13], and for semidefinite optimization problems from the SDPLIB suite in [4].

4.1.1 Condition Measures

The notion of (primal) distance to infeasibility was first presented in [15] for the purpose of developing perturbation theory for linear optimization problems. Consider the following problem:

$$
\begin{array}{lll}
 & \min\limits_{x} & c^T x \\
(\mathrm{CP}_d) & s.t. & Ax - b \in C_Y \\
 & & x \in C_X,
\end{array}
$$

where $d := (A, b, c)$ is the data of the problem ($A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m, c \in \mathbb{R}^n$; $C_X \subseteq \mathbb{R}^n$ and $C_Y \subseteq \mathbb{R}^m$ are closed convex cones).

Given norms $\|\cdot\|^Y$ on $\mathbb{R}^m$ and $\|\cdot\|^X$ on $\mathbb{R}^n$, using the usual operator norm $\|\cdot\|_{X,Y}$ for operators mapping elements of X to Y, and denoting $\|\cdot\|_*$ to be the dual norm associated with $\|\cdot\|$, we define the norm of a data instance d to be $\|d\| := \max\{\|A\|_{X,Y}, \|b\|^Y, \|c\|_*^X\}$.

Now, we define the feasible region of (CP_d) to be

$$X_d := \{x \in \mathbb{R}^n : Ax - b \in C_Y, x \in C_X\}.$$

Define the distance to primal infeasibility as

$$\rho_P(d) := \inf\{\|\Delta d\| : X_{d+\Delta d} = \emptyset\},$$

which is the norm of the smallest perturbation (in the same norm) that would render the primal problem infeasible.

Accordingly, the conic dual of (CP_d) is:

$$
\begin{array}{lll}
 & \min\limits_{y} & b^T y \\
(\mathrm{CD}_d) & s.t. & c - A^T y \in C_X^* \\
 & & y \in C_Y^*,
\end{array}
$$

where C^* denotes the dual cone to the cone C, and define the distance to dual infeasibility $\rho_D(d)$ in the same way as for the distance to primal infeasibility.

We define the primal condition number as $C_P(d) := \|d\|/\rho_P(d)$ and define the dual condition number $C_D(d) := \|d\|/\rho_D(d)$. The Renegar condition measure $C(d) := \max\{C_P(d), C_D(d)\}$ is then a scale-invariant reciprocal of the smallest perturbation of the data d that would render the problem either primal or dual infeasible. A problem is called ill-posed if $C(d) = \infty$.

In the event that we are not given $\|d\|$ exactly, we use the geometric mean of the tabulated upper and lower bounds (in [13] and [4]). For details of the respective norms used in the data, the reader should consult [13] and [4].

4.1.2 Geometry Measures

Aside from the distance to primal/dual infeasibility, [4] introduces a separate set of geometry-based measures. We are interested in particular in the measure g_d. Note that SDPLIB suite problems are in dual form:

$$\begin{array}{lll} & \min\limits_{y} & b^T y \\ (\text{DSDP}) & s.t. & B - \sum_{i=1}^n A_i y_i = S \\ & & S \in C \end{array}$$

where B, S, A_i $(i = 1, 2, \ldots, n)$ are $d \times d$ symmetric matrices and C is the semidefinite cone of $d \times d$ symmetric matrices (hence of dimension $m = d(d+1)/2$).

The dual geometry measure g_d for the problem (DSDP) is computed by solving the following problem:

$$\begin{array}{lll} & g_d := \min\limits_{y,S} & \max\left\{\|S\|, \frac{\|S\|}{\text{dist}(S,\partial C)}, \frac{1}{\text{dist}(S,\partial C)}\right\} \\ (P_{g_d}) & s.t. & B - \sum_{i=1}^n A_i y_i = S \\ & & S \in C \end{array}$$

For details on the choice of norm, which is crucial to obtain a tractable formulation of (P_{g_d}), the reader may consult [4].

Note that g_d is smaller to the extent that the feasible region of (DSDP) contains a point (positive semidefinite matrix) S whose norm is not too large and whose distance from the boundary of the semidefinite cone is not too small.

[4] also introduces another geometry measure D_d^ϵ for $\epsilon > 0$, which measures the largest norm among the set of ϵ-optimal solutions. Unfortunately, many of our test problems have $D_d^\epsilon = \infty$ (implied by the fact that $C_P(d) = \infty$ for those problems). In particular, all the `hinf` problems from the SDPLIB suite, except `hinf2` and `hinf9`, have $D_d^\epsilon = \infty$. In [4], analogs for the primal problem of g_d and D_d^ϵ, g_p and D_p^ϵ, are presented as well.

In addition, [4] develops an aggregate geometry measure g^m which is the geometric mean of g_d, D_d^ϵ, g_p and D_p^ϵ.

4.1.3 Correlation for Linear Optimization Problems

The results for some NETLIB suite problems are presented in Figures B-1 to B-5 and summarized in Table B.1.

In Figure 4-1, we look for a relationship between the required number of iterations to reduce the relative optimality gap to `1e-6` and the dimension of the problem n (and that of the relative interior n_{affine}; points corresponding to the same problem are connected by dots). The results suggest that there is a potential positive correlation between n_{affine} and the required number of iterations.

In Figures 4-2 and 4-3, we look for a relationship between the required number of iterations to reduce the relative optimality gap to `1e-6` and the Renegar condition measure $C(d)$. The results suggest that there is a possible positive correlation between $C(d)$ and the required number of iterations.

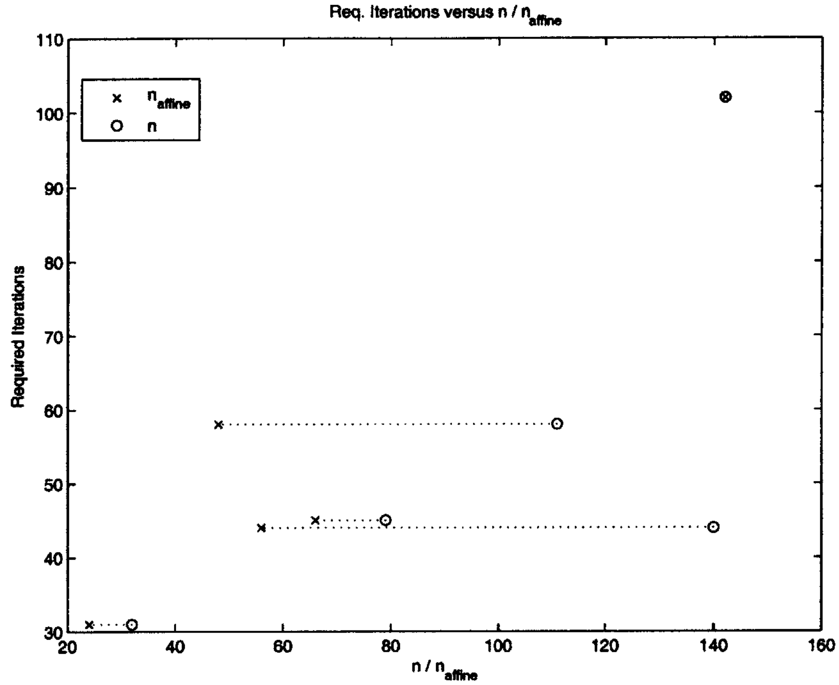

Figure 4-1: Required Iterations versus Dimension for 5 NETLIB Problems

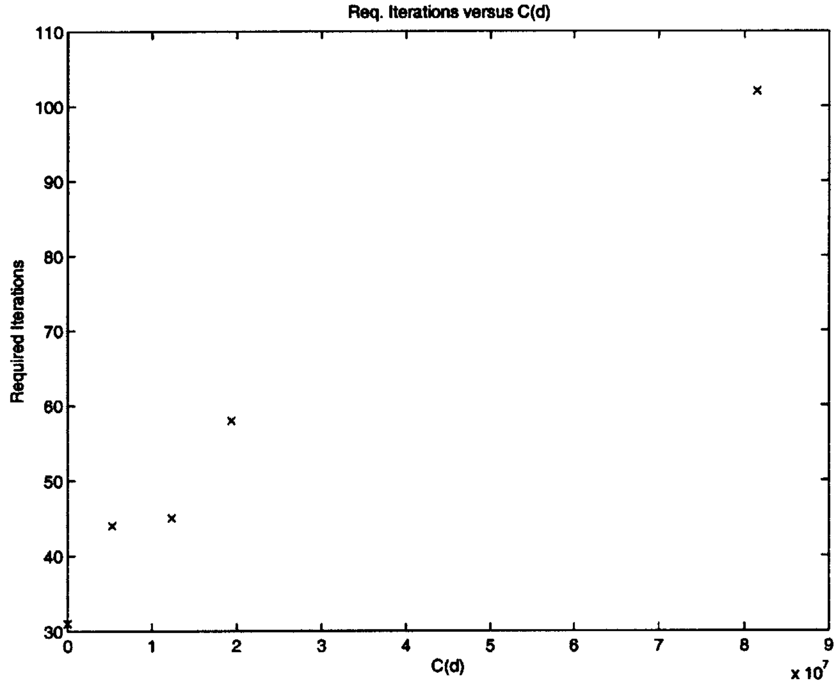

Figure 4-2: Required Iterations versus $C(d)$ for 5 NETLIB Problems

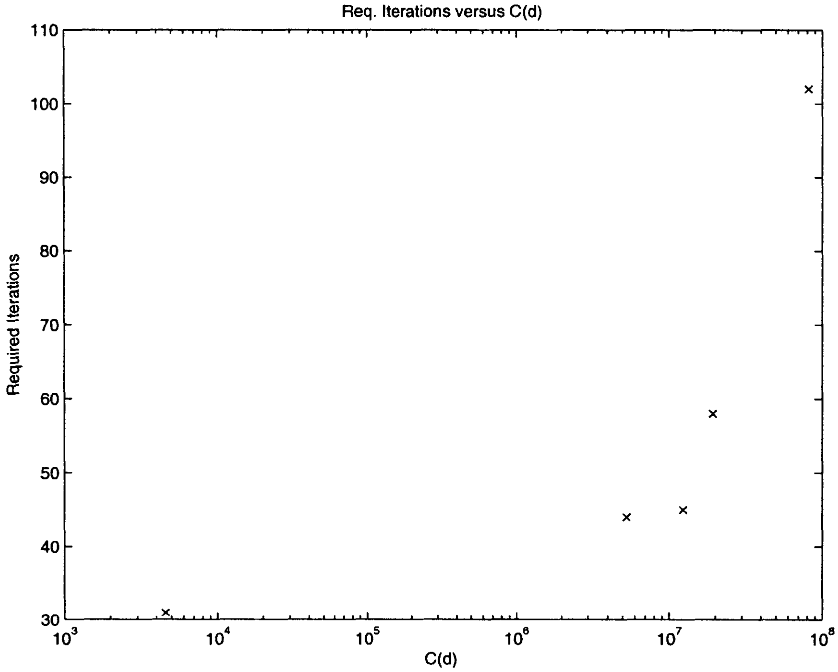

Figure 4-3: Required Iterations versus $C(d)$ for 5 NETLIB Problems

4.1.4 Correlation for Semidefinite Optimization Problems

The results for some SDPLIB suite problems are presented in Figures B-6 to B-16 and summarized in Table B.2.

In Figure 4-4, we look for a relationship between the required number of iterations to reduce the relative optimality gap to 1e−5 and the dimension of the problem n. The results suggest that there is not a relationship between n and the required number of iterations.

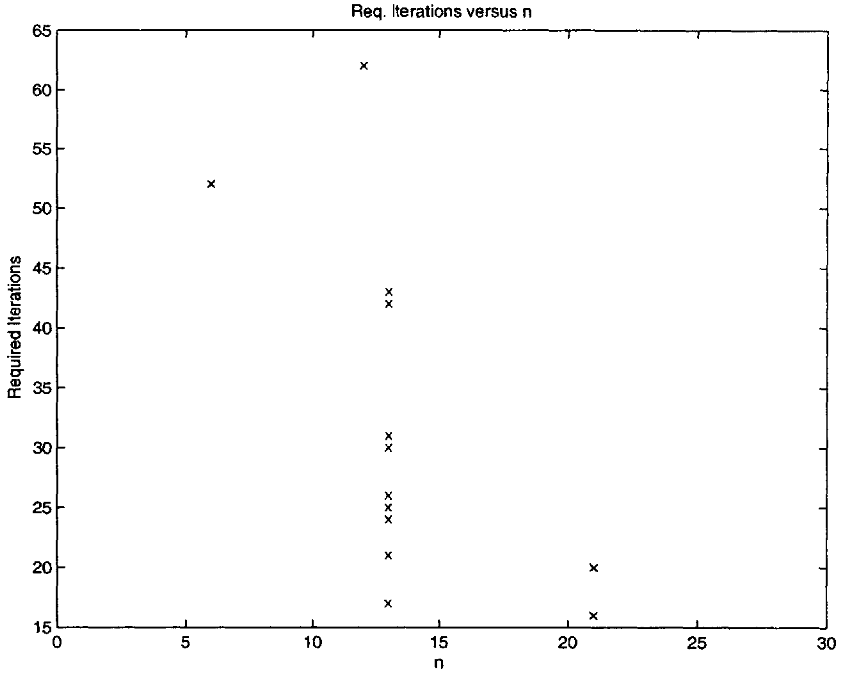

Figure 4-4: Required Iterations versus Dimension for 13 SDPLIB Problems

In Figure 4-5, we look for a relationship between the required number of iterations to reduce the relative optimality gap to 1e−5 and the order d of the semidefinite cone. The results suggest that there is not a relationship between d and the required number of iterations.

It is unfortunate that the results suggest neither the dimension of the problem n nor the order of the matrices in the semidefinite cone d play an important role in determining the required number of iterations to achieve a given relative optimality

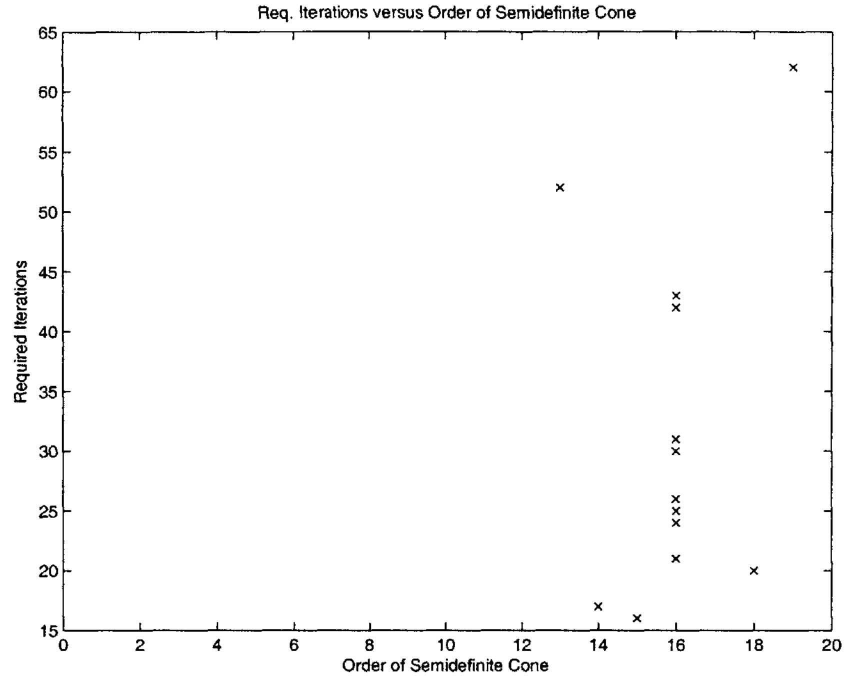

Figure 4-5: Required Iterations versus Order of Matrices in the Semidefinite Cone for 13 SDPLIB Problems

gap. This implies that the bound in Lemma 2.9 may not be very useful as a stopping criteria. However, the computational results may result from not making sufficient steps in the hit-and-run random walk so that the distribution of points did not to converge sufficiently to an exponential distribution.

In fact, obtaining a means of choosing a number of hit-and-run steps for convergence under reasonable conditions warrants further study (the requirement in Theorem 2.8 is far too stringent).

There are five semidefinite optimization problems from the SDPLIB suite among the problems tested for which $C(d)$ and g^m are finite. For those, the required iteration count versus $C(d)$ is graphed in Figures 4-6 and 4-7; the required iteration count versus g^m is graphed in Figures 4-8 and 4-9. Unfortunately, these plots do not indicate anything of note.

Noting that for the 13 SDPLIB suite problems that were tested, $C_D(d)$ was finite, in Figures 4-10 and 4-11, we look for a relationship between the required number

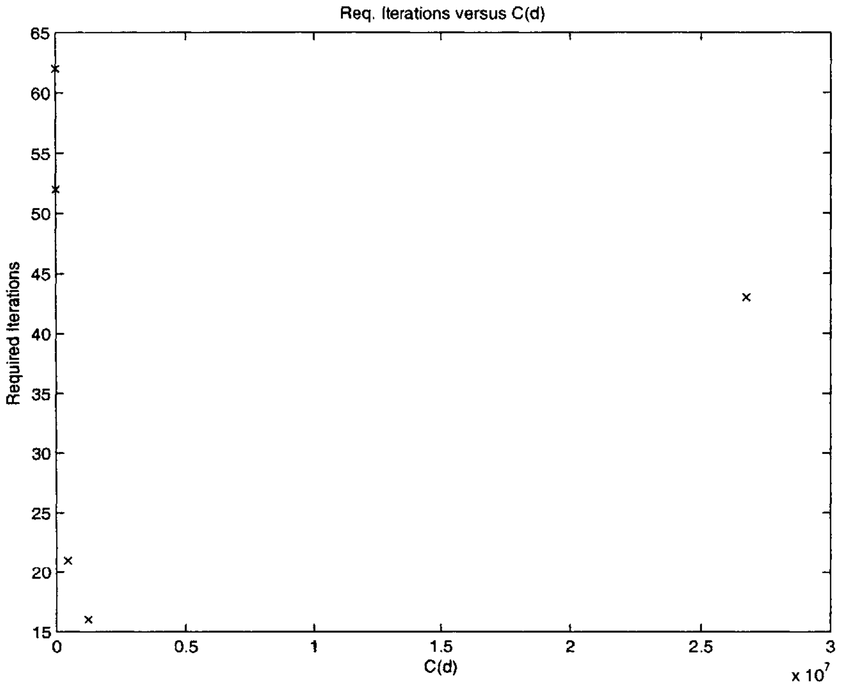

Figure 4-6: Required Iterations versus $C(d)$ for 5 SDPLIB Problems

of iterations to reduce the relative optimality gap to 1e-5 and the dual condition number $C_D(d)$. Unfortunately, no relationship between the two is indicated.

In Figures 4-12 and 4-13, we look for a relationship between the required number of iterations to reduce the relative optimality gap to 1e-5 and the geometry measure g_d. There appears to be a reasonable positive relationship between the required number of iterations and $\log g_d$. (Note once more that there are insufficient data points to make a definitive statement, and results here are only indicative.)

The computational results suggest that for semidefinite optimization problems, the underlying geometry may play a very important role in determining the number of iterations to solve a problem to some relative optimality tolerance. Unfortunately, this quantity cannot be known *a priori* and is not useful for obtaining a good termination criterion.

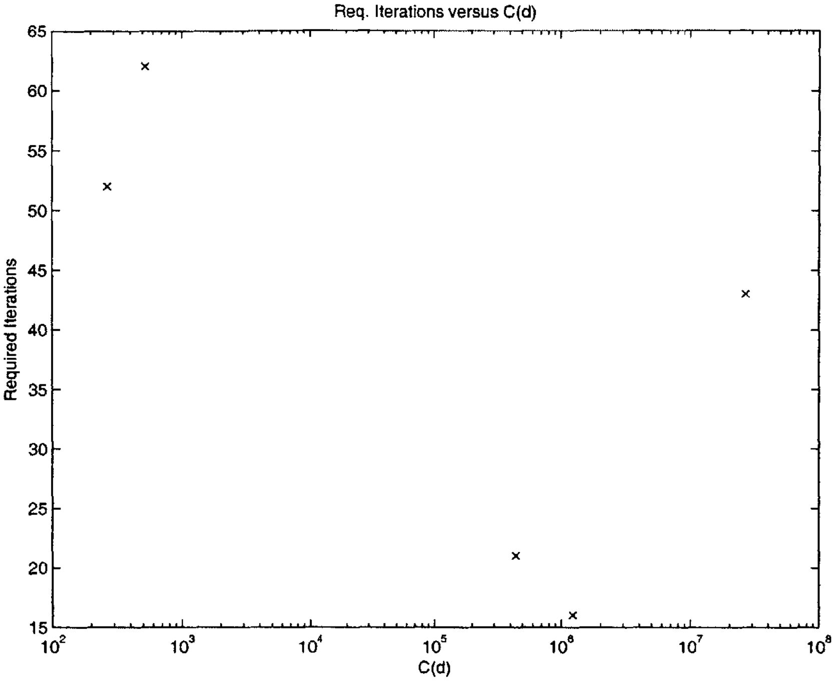

Figure 4-7: Required Iterations versus $C(d)$ for 5 SDPLIB Problems

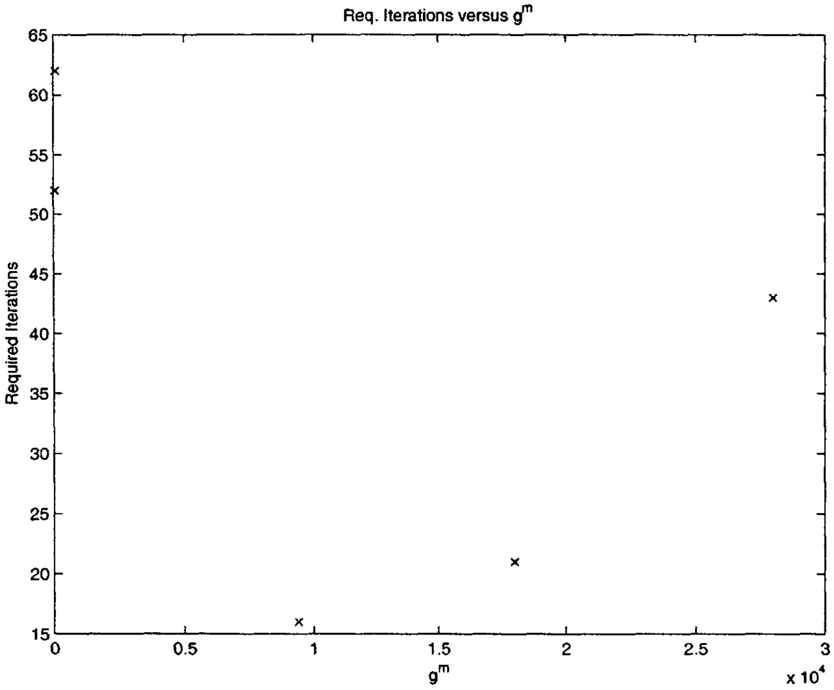

Figure 4-8: Required Iterations versus g^m for 5 SDPLIB Problems

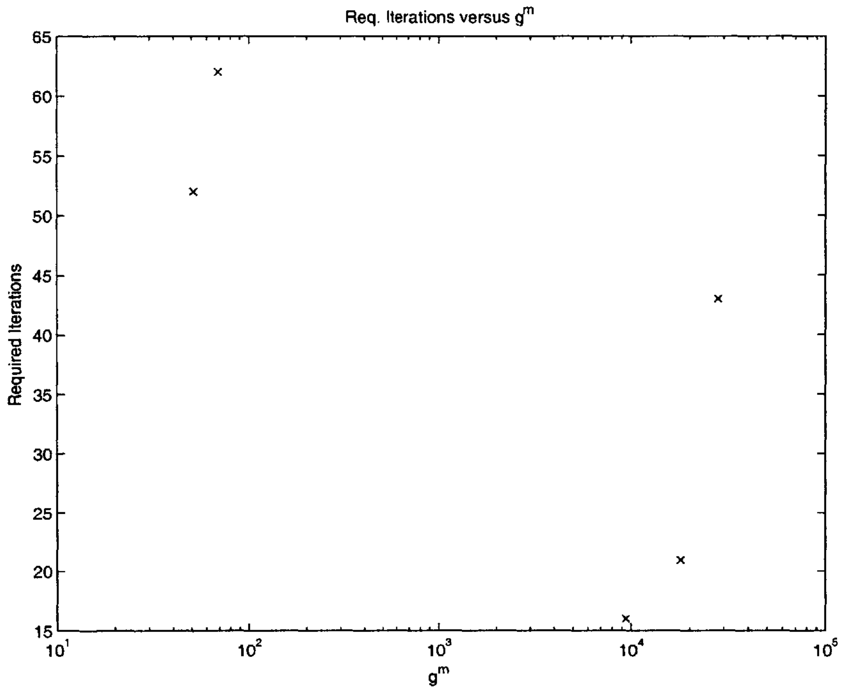

Figure 4-9: Required Iterations versus g^m for 5 SDPLIB Problems

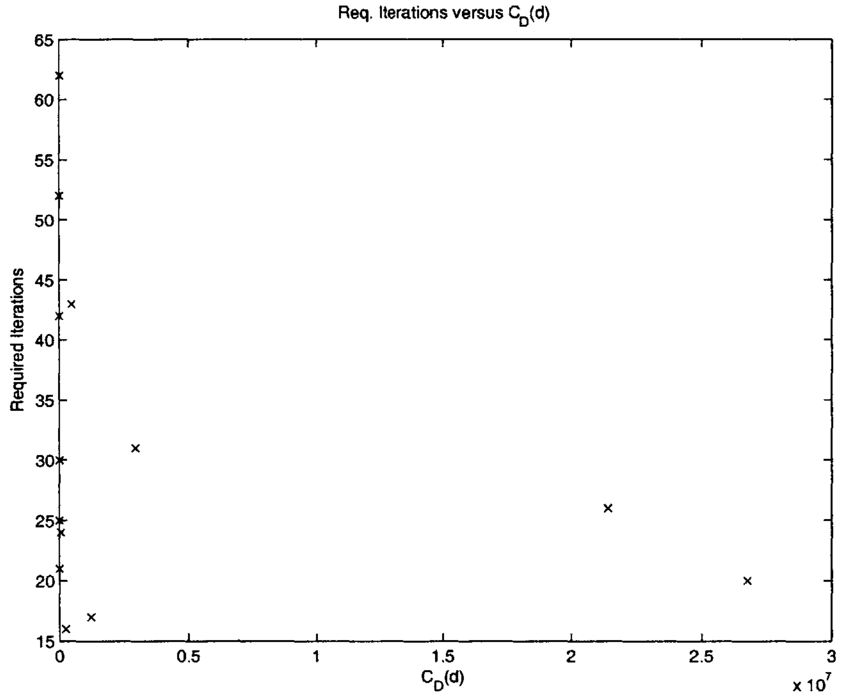

Figure 4-10: Required Iterations versus $C_D(d)$ for 13 SDPLIB Problems

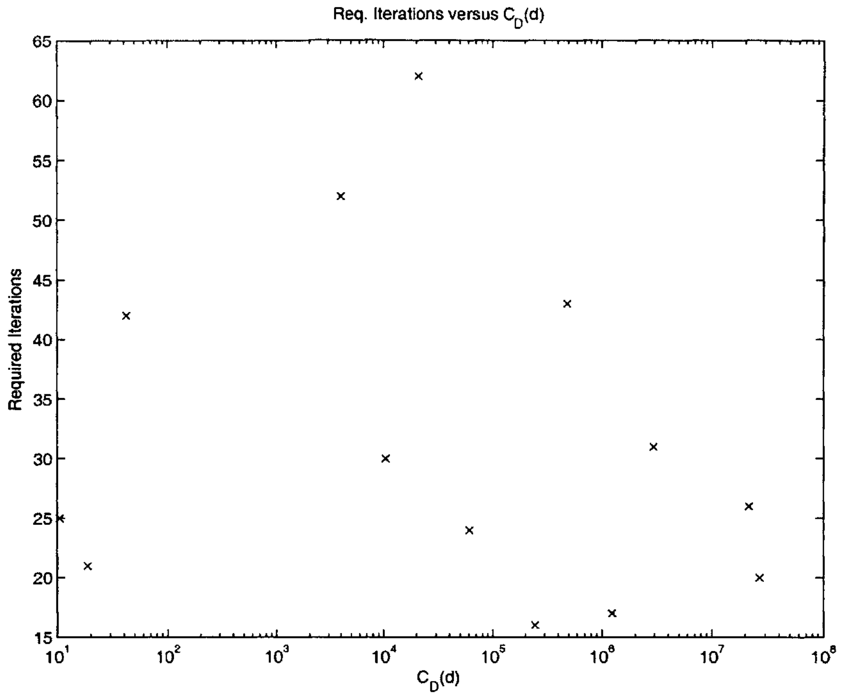

Figure 4-11: Required Iterations versus $C_D(d)$ for 13 SDPLIB Problems

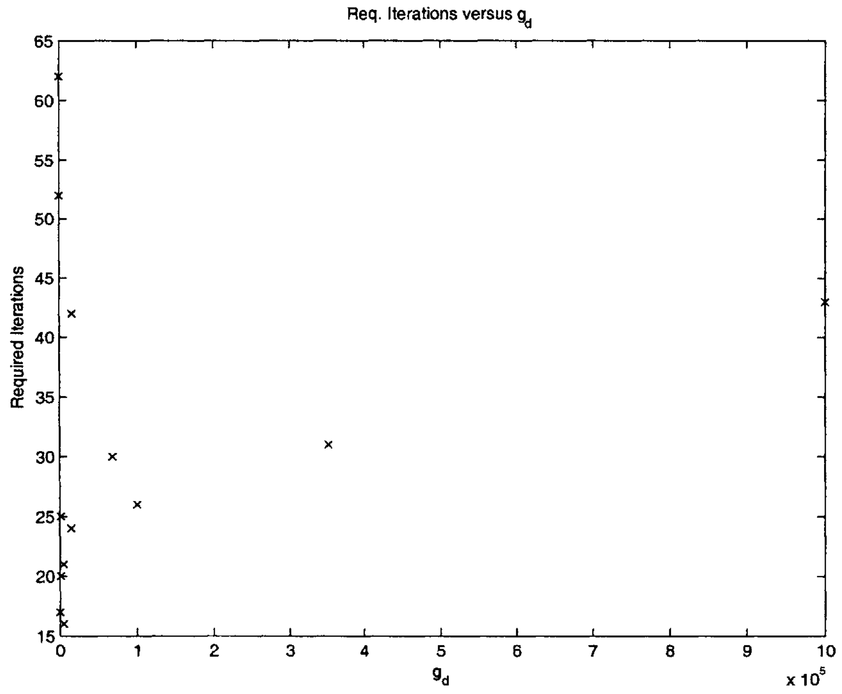

Figure 4-12: Required Iterations versus g_d for 13 SDPLIB Problems

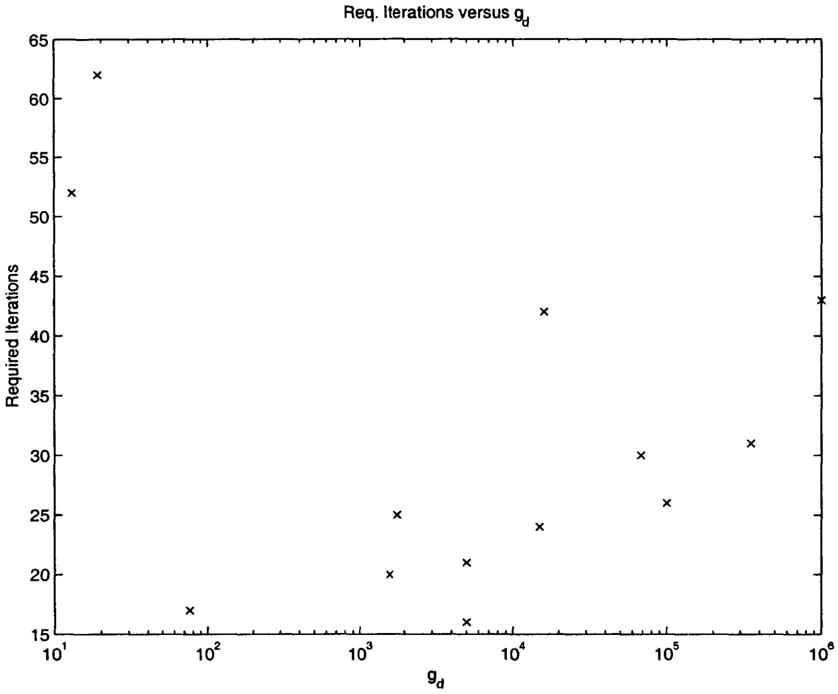

Figure 4-13: Required Iterations versus g_d for 13 SDPLIB Problems

4.2 Possible Extensions

4.2.1 Stopping Rules

One might consider stopping the random walk when sufficient progress has been made. In order to quantify "sufficient progress", consider the following lemma by Lovász and Vempala in [10]:

Lemma 4.1 (Lovász and Vempala [10]). *Let $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ be an isotropic logconcave density function, and let z be a point where it assumes its maximum. Then*

$$\|z\| \leq n + 1,$$

and this bound is tight.

By translating and scaling, we obtain the following Corollary:

Corollary 4.2. *Let $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ be a logconcave density function with mean μ and covariance matrix Σ, and let z be a point where it assumes its maximum. Then*

$$\sqrt{(z-\mu)^T \Sigma^{-1} (z-\mu)} \leq n + 1,$$

and this bound is tight.

Since the exponential distribution is logconcave, this lemma and its corollary give a useful bound on the distance from the mean to the optimal solution set. For details on logconcave functions, the reader is referred to [10].

This implies that starting from the mean of an exponential distribution transformed to isotropicity, the optimal solution set is at most a distance of $n + 1$ away (after scaling the Euclidean unit sphere).

Indexing the steps of the random walk beginning at $x^{(0)}$ with r, consider the following stopping rule: stop when $(V^T c)^T (x^{(0)} - x^{(r)}) \geq \eta(n+1)\|V^T c\|$ (sufficient decrease), or when r reaches N_s where N_s is the maximum number of steps in the random walk. For this, one chooses $\eta \in (0, 1)$. This would serve to reduce the number of steps taken, and hence the computational work to be performed.

When tested, however, it was found that in later phases, the random walk usually concluded with $r = N_s$. One way to explain the problem is that the bound of Corollary 4.2 is unduly strict for our problem. Consider an isotropic exponential distribution on a cone. One can verify (using the result of Lemma 2.10) that the distance of the mean to the vertex of the cone is $\sqrt{n}$. This implies that one might wish to consider instead $(V^Tc)^T(x^{(0)} - x^{(r)}) \geq \eta\sqrt{n}\|V^Tc\|$.

A natural concern with using such stopping rules would be whether their use would invalidate the bound of Lemma 2.9 since under the stopping rule the iterates would in general not be close to exponentially distributed on the feasible region.

This extension may be an interesting area for further investigation.

4.2.2 A Primal-Dual Reformulation for a Guarantee of ϵ-Optimality

In order to obtain a guarantee for ϵ-optimality, one might want to consider the following instead. For the original problem (from Section 1.2):

$$\begin{array}{cclcl} & \min\limits_{x} & c^Tx & & \\ (P) & s.t. & b^{(e)} - A^{(e)}x & = & 0 \\ & & b^{(c)} - A^{(c)}x & \in & C, \end{array}$$

where $b^{(e)} \in \mathbb{R}^{m_e}$, $A^{(e)} \in \mathbb{R}^{m_e \times n}$, $b^{(c)} \in \mathbb{R}^{m_c}$, $A^{(c)} \in \mathbb{R}^{m_c \times n}$. The dual of (P) is

$$\begin{array}{cclcl} & \max\limits_{y,z} & b^{(e)T}y + b^{(c)T}z & & \\ (D) & s.t. & c - A^{(e)T}y - A^{(c)T}z & = & 0 \\ & & z & \in & C^* \end{array}$$

where C^* is the dual cone of C. Now, taking advantage of weak duality, we consider the following primal-dual formulation:

$$
(PD) \qquad \begin{array}{rlcl}
\min\limits_{x,y,z} & c^T x - b^{(e)T} y - b^{(c)T} z & & \\
s.t. & b^{(e)} - A^{(e)} x & = & 0 \\
 & c - A^{(e)T} y - A^{(c)T} z & = & 0 \\
 & b^{(c)} - A^{(c)} x & \in & C \\
 & z & \in & C^*
\end{array}
$$

which can be seen to be of the same form as (P). Now, the dimension of the problem has been increased, and to begin the problem we have to find primal-dual feasible points as opposed to just primal feasible points. But, if the objective function value is less than ϵ, one has a guarantee that the current primal-dual pair x and (y, z) are ϵ-optimal.

One may be concerned with the increase in the number of unknowns in the problem from n to $n + m_e + p$ where p is the dimension of the smallest Euclidean space where one may embed the dual cone of C. In the case of the semidefinite cone, if p is $O(n^2)$, then redilating the space would require $O(n^6)$ operations.

Another aspect that one may be concerned about would be the fact that while some primal problems have strictly feasible solutions, their dual problems may not. This necessitates pre-processing of the problem such that both primal and dual problems have strictly feasible solutions. Examples of this can be found in the NETLIB and SDPLIB suites, and are characterized by either $C_P(d) = 0$ or $C_D(d) = 0$ (these are tabulated in [13] and [4]). This may be overcome by the use of the homogeneous self-dual model which has strictly feasible solutions.

4.2.3 Adaptive Cooling

In the process of developing exponential annealing, it was found that the conditions for achieving certain probabilistic guarantees were too stringent to be practical (see for instance, Theorem 2.8). In seeking a practical algorithm, we have ignored much theory, especially in reducing the temperature parameter by a constant factor at each phase. This proposal seeks to modify the method to reduce the number of theoretical loopholes.

One may consider instead a scheme of adaptive temperature reduction. Consider first the feasibility problem and then the primal-dual reformulation of Section 4.2.2.

When attempting to find a feasible point for problem (P) from Section 1.2, reproduced here:

$$(P) \quad \begin{array}{rrcl} \min\limits_{x} & c^T x & & \\ s.t. & b^{(e)} - A^{(e)}x & = & 0 \\ & b^{(c)} - A^{(c)}x & \in & C, \end{array}$$

we attempt to find a feasible (interior) point for (P') from Section 1.2, with $\theta < 0$, reproduced here:

$$(P') \quad \begin{array}{rlcl} \min\limits_{x,\delta,\theta} & \theta & & \\ s.t. & \delta b^{(e)} - A^{(e)}x - \theta b^{(e)} & = & 0 \\ & \delta b^{(c)} - A^{(c)}x + \theta(y - b^{(c)}) & \in & C \\ & \delta & \geq & 0 \\ & \|(x, \delta)\|_2 & \leq & 2. \end{array}$$

where $y \in int\, C$.

This problem and the primal-dual reformulation have in common that 0 is an upper bound for the optimal solution (assuming that the primal is feasible). This piece of *a priori* information turns out to be very useful.

Suppose, at the end of some phase, given that one has a candidate solution for either problem with value v, one may use Lemma 2.9 to choose the next value for the temperature parameter such that the expected value of the next candidate solution is at most γv where $\gamma \in (0, 1)$ (note that 0 is an upper bound on the optimal solution value).

We then simply perform sufficient hit-and-run steps until the "current iterate" has a solution value of at most $\gamma(1 + \beta)v$ (where $\beta > 0$). Note that Theorem 2.8 gives an upper bound on the number of steps we have to take in order for the distribution of the current iterate in any hit-and-run random walk to approach our desired distribution. This is more useful in a practical setting as we are not using Theorem 2.8 to set the required number of hit-and-run steps.

In doing this, we will no longer have any theoretical loopholes in our algorithm.

4.3 Conclusions

Exponential annealing looks promising from a complexity perspective for semidefinite optimization (see Subsection 3.2.2). Unfortunately computational results for semidefinite optimization suggest that the result of Lemma 2.9 may not be useful as a termination criteria.

On the other hand, the proposed use of the primal-dual reformulation of the primal problem in tandem with adaptive cooling addresses the problem of obtaining a termination criterion and ensures that the algorithm no longer falls afoul of theory. It remains to study if this extension can be competitive, now competing as a primal-dual method.

While exponential annealing as presented in Algorithm 2 is not yet competitive with IPMs and appears to have a problem with regard to performance guarantees, it may be that the primal-dual reformulation and an adaptive cooling scheme would make exponential annealing a practical and competitive algorithm.

Appendix A

Addendum for Semidefinite Optimization

A.1 Maximum Step Lengths in Semidefinite Optimization

Consider the problem of finding the maximum step length such that $X + \alpha D \in C$ where X is a positive definite matrix, D is a symmetric matrix of the same dimension, and C is the semidefinite cone. This is equivalent to finding the maximum value of α such that $I + \alpha R^{-T} D R^{-1}$ remains positive semidefinite, where $X = R^T R$. (Let $A = R^{-T} D R^{-1}$.)

It may not be necessary to compute extremely accurate values for the upper and lower bounds for which $I + \alpha A \succeq 0$. In [18], Toh describes how the Lanczos iteration (which is how the Hessenberg form is computed) can be used to find approximations to the extremal eigenvalues of A, and gives computationally useful *a posteriori* error bounds for the computed eigenvalues.

Given an initial vector q_1, the Lanczos iteration generates an orthonormal sequence of vectors $\{q_j\}$, a sequence of orthonormal matrices $\{Q_j\}$ and a sequence of symmetric tridiagonal matrices $\{T_j\}$ (where $Q_j = [q_1 \; q_2 \; \dots \; q_j]$) which satisfy $AQ_j = Q_j T_j + t_{j+1,j} q_{j+1} e_j^T$ and $Q_j^T A Q_j = T_j$. Given T_j and Q_j, q_{j+1} and $t_{j+1,j}$ may be easily computed.

In this description, we assume that we do not "get stuck" in an invariant subspace, giving $t_{j+1,j} = 0$, but this can be fixed by choosing q_{j+1} orthogonal to the previous vectors, then setting $t_{j+1,j} = 0$, $t_{j+1,j+1} = q_{j+1}^T A q_{j+1}$. One "gets stuck" when q_1 is chosen such that the subspace spanned by $\{q_1, Aq_1, A^2q_1, \ldots, A^{n-1}q_1\}$ has dimension less than n. (Fortunately, if q_1 is chosen randomly, this occurs with probability 0.) We have also neglected to describe other numerical issues such as the possible need for re-orthogonalization, for which the reader may refer to [14].

In any event, the extremal eigenvalues of T_j are good estimates for the extremal eigenvalues of A. It is well known that accurately computing the eigenvalues of a symmetric tridiagonal matrix can be done in essentially $O(n)$ steps with powerful iterative methods. Suppose the largest and second largest eigenvalues of T_j are $\tilde{\lambda}_1$ and $\tilde{\lambda}_2$ respectively (we traditionally label eigenvalues in descending order), with corresponding eigenvectors $\tilde{y}_1$ and $\tilde{y}_2$. Let $\tilde{u}_i = Q_j\tilde{y}_i$ and $r_i = A\tilde{u}_i - \tilde{\lambda}_i\tilde{u}_i$.

Toh proves the following upper bound for the maximum eigenvalue of A. (One may apply the same to $-A$ to obtain a lower bound for the minimum eigenvalue of A.)

Theorem A.1 (Toh [18]). *Suppose the eigenvalues closest to $\lambda_1(A)$ and $\lambda_2(A)$ are $\tilde{\lambda}_1$ and $\tilde{\lambda}_2$, then*

$$\lambda_1(A) \leq \tilde{\lambda}_1 + \delta$$

where

$$\delta = \begin{cases} \|r_1\|_2 & \text{if} \lambda_1 \leq \tilde{\lambda}_2 + \|r_2\|_2 \\ \min\left\{\|r_1\|_2, \frac{\|r_1\|_2^2}{\tilde{\lambda}_1 - \tilde{\lambda}_2 - \|r_2\|_2}\right\} & \text{otherwise} \end{cases}$$

The condition given may sound strange as there is no way to check it computationally, but the reason for its presence is that it attends to the matter of "getting stuck" in an invariant subspace, which implies the possibility that eigenvectors corresponding to the largest and second largest eigenvalues may lie in another invariant subspace of A. Again, it is fortunate that for a random choice of q_1, this occurs with probability 0.

A.2 On the Computational Cost of a Single IPM Newton Step for Semidefinite Optimization

Consider the simple semidefinite optimization problem:

$$\text{(SDP)} \quad \begin{array}{ll} \min\limits_{y} & b^T y \\ s.t. & B - \sum_{i=1}^n A_i y_i = S \\ & S \in C \end{array}$$

where B, S, A_i $(i = 1, 2, \dots, n)$ are $d \times d$ symmetric matrices and C is the semidefinite cone of $d \times d$ symmetric matrices (hence of dimension $m = d(d+1)/2$). In this elementary analysis, assume that all given matrices are dense.

A simple IPM would begin with a feasible interior point and given a sequence of problems indexed by increasing values of the parameter ν, a Newton step would be taken for each problem and used as the starting point for the next Newton step of the next problem (this is the elementary "short-step" method presented in [16] to introduce the "barrier method"),

$$\text{(SDPB}_\nu\text{)} \quad \begin{array}{ll} \min\limits_{y} & f(y) := \nu b^T y - \ln\det(S) \\ s.t. & B - \sum_{i=1}^n A_i y_i = S \\ & S \in C. \end{array}$$

Now, given y such that the corresponding matrix S is positive definite with Cholesky factorization $R^T R$, the quadratic approximation to f at y can be shown to be given by (see for instance [16] or [3]):

$$\tilde{f}(y+h) := f(y) + \sum_{i=1}^{m} \left(\nu b_i + \langle A_i, S^{-1}\rangle\right) h_i + \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \langle A_i, S^{-1} A_j S^{-1}\rangle h_i h_j.$$

Before the assembly of the gradient vector and Hessian matrix, one performs a Cholesky factorization of S requiring $O(d^3)$ operations with leading term $\frac{1}{3}d^3$. (For future reference, a triangular back solve requires $O(d^2)$ operations with leading term

d^2, and the dimension of the Newton system is n.)

For the assembly of the gradient vector, one requires $O(nd^3)$ operations with leading term $2nd^3$ (one computes $\left\langle R^{-T}A_iR^{-1}, I\right\rangle$ using $2n$ back solves instead of $\langle A_i, S^{-1}\rangle$ by explicitly forming S^{-1}, then computes the trace product).

For the assembly of the Hessian matrix, one requires $O(n^2d^3)$ operations with leading term $2n^2d^3$ (one computes $\langle A_i, S^{-1}A_jS^{-1}\rangle$ using $4d$ back solves and a trace product, taking advantage of the symmetry of the Hessian). Hence, the assembly of the Newton system has an operation count with leading term $2n^2d^3$. [Alternatively, one may form $S^{-1}A_j$ for $j = 1, 2, \ldots, n$, and perform $d(d+1)/2$ trace products for an operation count with leading (undominated) terms $2nd^3 + \frac{1}{2}n^2d^2$. Note that this is done at the cost of storing n $d \times d$ matrices.]

The operation count of solving the Newton system by a Cholesky factorization and back solve is $O(n^3)$ with leading term $\frac{1}{3}n^3$.

This gives a guide to the computational cost of taking a Newton step in an interior point method for semidefinite optimization without the benefit of sparsity.

Appendix B

Computational Results for NETLIB and SDPLIB Test Problems

In this section, n refers to the dimension of the problem and where mentioned, n_{affine} refers to the dimension of the affine subspace described by the equality constraints.

All problems are solved using Algorithm 2 with parameters outlined in Subsection 3.2.1.

(Note also that for problems for which an initial feasible point was not supplied, relative optimality gaps are not shown in the phase where a feasible point is being sought.)

B.1 Linear Optimization Problems from the NETLIB Suite

In all diagrams that follow, "Distance from Boundary (in $\mathbb{R}^n$)" refers to the Euclidean distance of the current candidate solution to the boundary of the set defined by the inequality constraints. This quantity is a lower bound on the distance to the boundary of the set in the smallest affine subspace containing the feasible region.

Problem	Req. Iter. (Rel Tol: 1e-6)	n	$C_P(d)$	$C_D(d)$	$C(d)$
AFIRO	31	32	4565	1814	4565
ISRAEL	102	142	8.147e+007	1.331e+007	8.147e+007
SCAGR7	44	140	5.307e+006	2.449e+006	5.307e+006
SHARE2B	45	79	1.233e+007	7.48e+004	1.233e+007
STOCFOR1	58	111	1.939e+007	2.139e+005	1.939e+007

Table B.1: Computational Results for 5 NETLIB Suite Problems

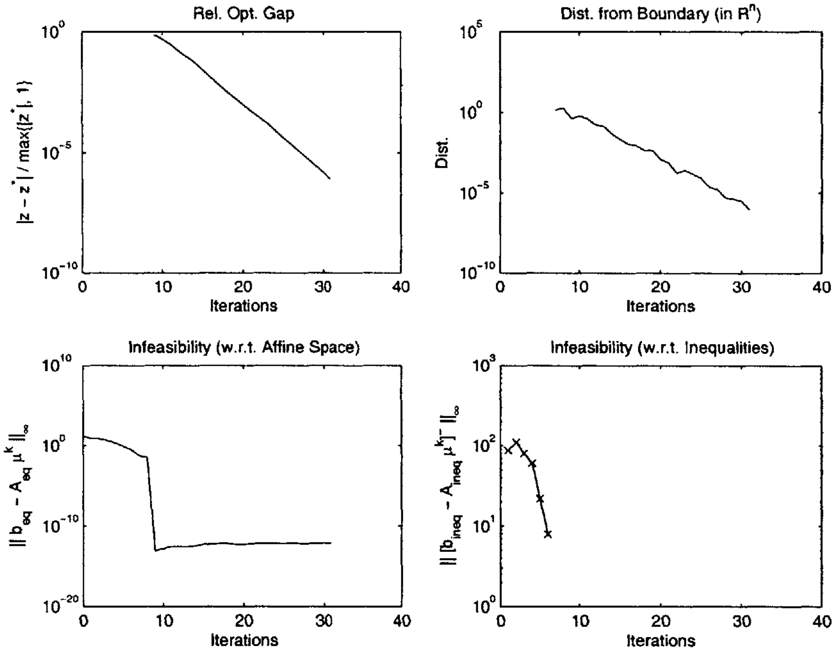

Figure B-1: NETLIB Problem AFIRO ($n = 32$, $n_{\text{affine}} = 24$)

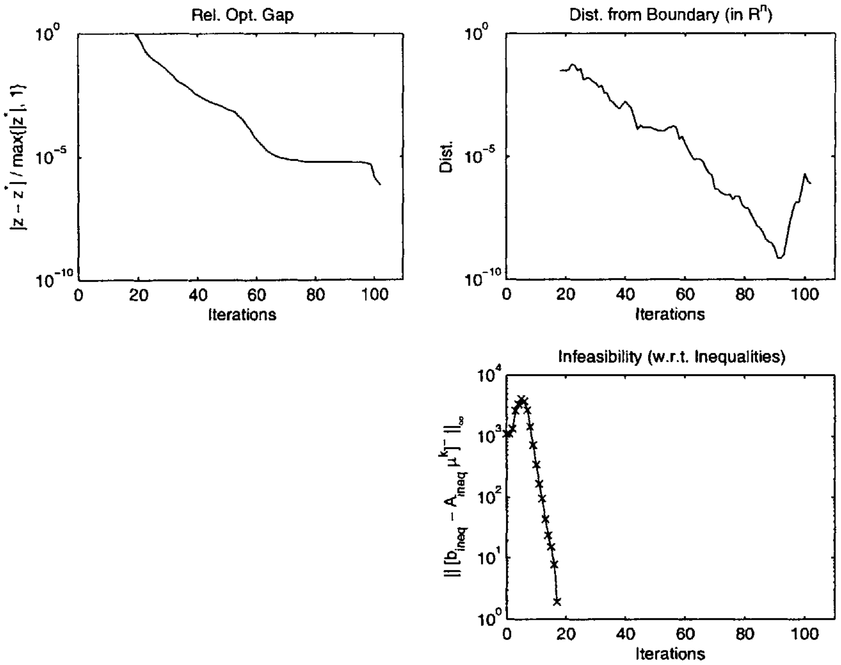

Figure B-2: NETLIB Problem ISRAEL (n = 142, n_{affine} = 142)

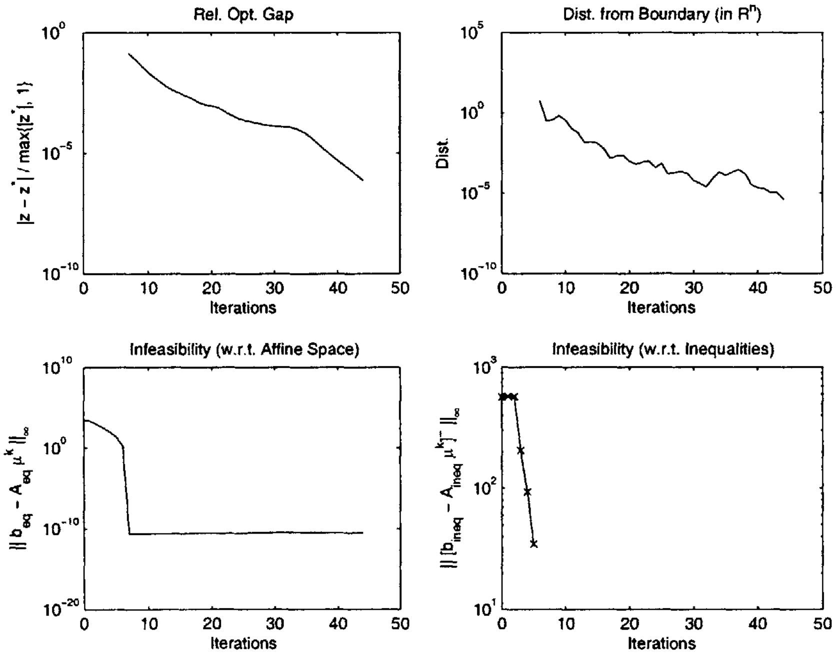

Figure B-3: NETLIB Problem SCAGR7 (n = 140, n_{affine} = 56)

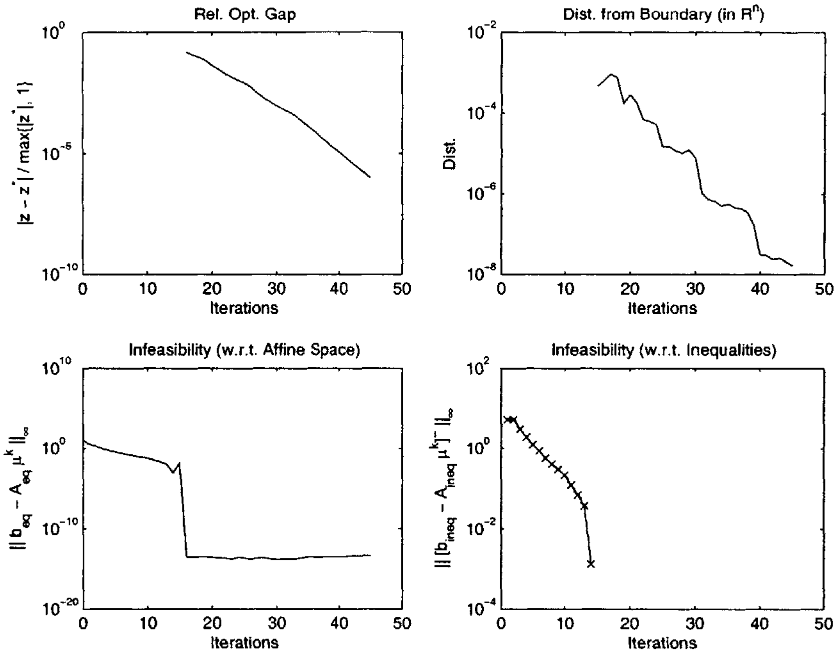

Figure B-4: NETLIB Problem SHARE2B ($n = 79$, $n_{\text{affine}} = 66$)

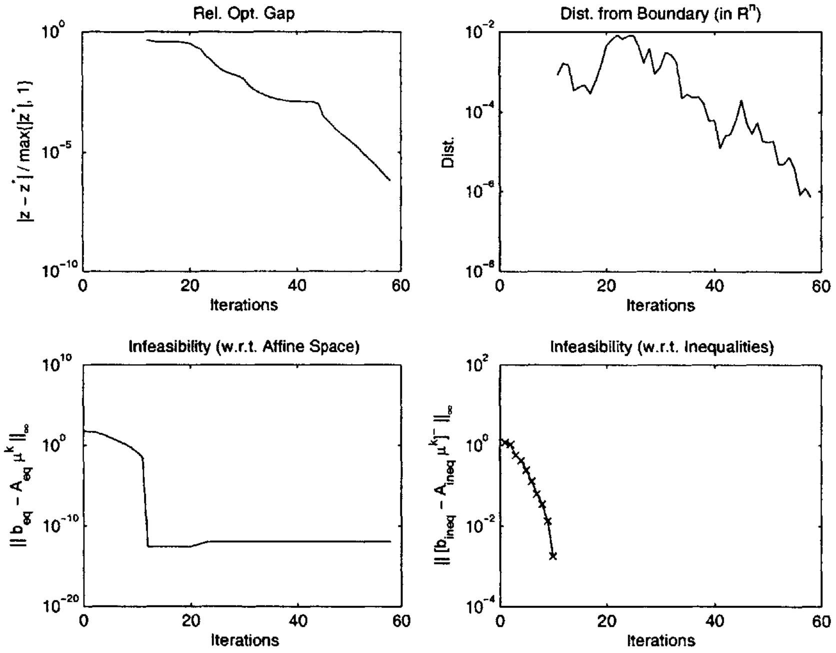

Figure B-5: NETLIB Problem STOCFOR1 ($n = 111$, $n_{\text{affine}} = 48$)

B.2 Semidefinite Optimization Problems from the SDPLIB Suite

In all diagrams that follow, "Distance from Boundary" refers to the minimum eigenvalue of the matrix $B - \sum_{i=1}^{n} y_i A_i$ (which is constrained to be positive semidefinite) given a current candidate solution y, where B and the A_i's are symmetric matrices defined by the problem data. Also, d gives the order of the matrices in the semidefinite cone ($d \times d$ symmetric matrices).

Note that the "optimal solution values" for each problem, z^*, were obtained from the SDPLIB documentation [2] and other sources [6] (the lowest available value was used). These "optimal solution values" were sometimes found to be only loose upper bounds.

The graphs for `hinf9` and `hinf10` are not shown because the method terminated with candidate solution values less than the lowest upper bound on the optimal solution value found. For `hinf9`, the method terminates with candidate solution value `237.262294312903`, while the best upper bound was `237.39`. For `hinf10`, the method terminates with candidate solution value `108.848461210598`, while the best upper bound was `108.86`. (For these two problems, the upper bound from [6] was used, though the article reports that the bounds it supplies for these two problems are likely to be loose.)

	Req. Iter.						
Problem	(Rel Tol: 1e-5)	m	s	$C_D(d)$	$C(d)$	g_d	g^m
control1	17	21	15	1.233e+006	1.233e+006	5000	9400
hinf1	42	13	14	42.15	∞	76	∞
hinf2	52	13	16	4025	4.427e+005	5000	1.8e+004
hinf3	24	13	16	6.078e+004	∞	1.5e+004	∞
hinf4	30	13	16	1.039e+004	∞	1800	∞
hinf5	31	13	16	2.939e+006	∞	1e+005	∞
hinf6	43	13	16	4.861e+005	∞	6.8e+004	∞
hinf7	26	13	16	2.142e+007	∞	3.5e+005	∞
hinf8	16	13	16	2.465e+005	∞	1.6e+004	∞
hinf9	20	13	16	2.675e+007	2.675e+007	1e+006	2.8e+004
hinf10	62	21	18	2.087e+004	∞	1600	∞
truss1	25	6	13	10.5	266.5	13	51
truss4	21	12	19	18.8	522.1	19	69

Table B.2: Computational Results for 13 SDPLIB Suite Problems

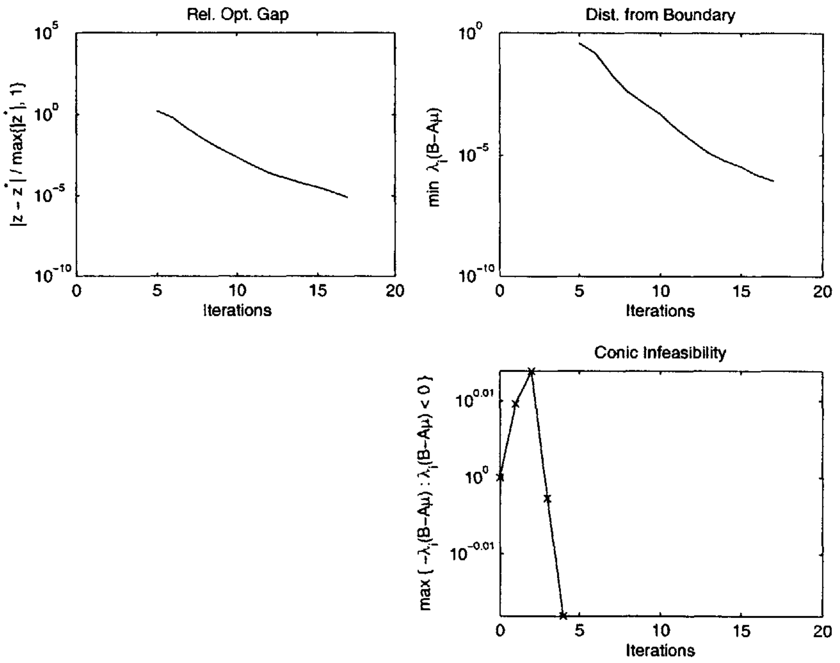

Figure B-6: SDPLIB Problem control1 ($n = 21$, $s = 15$)

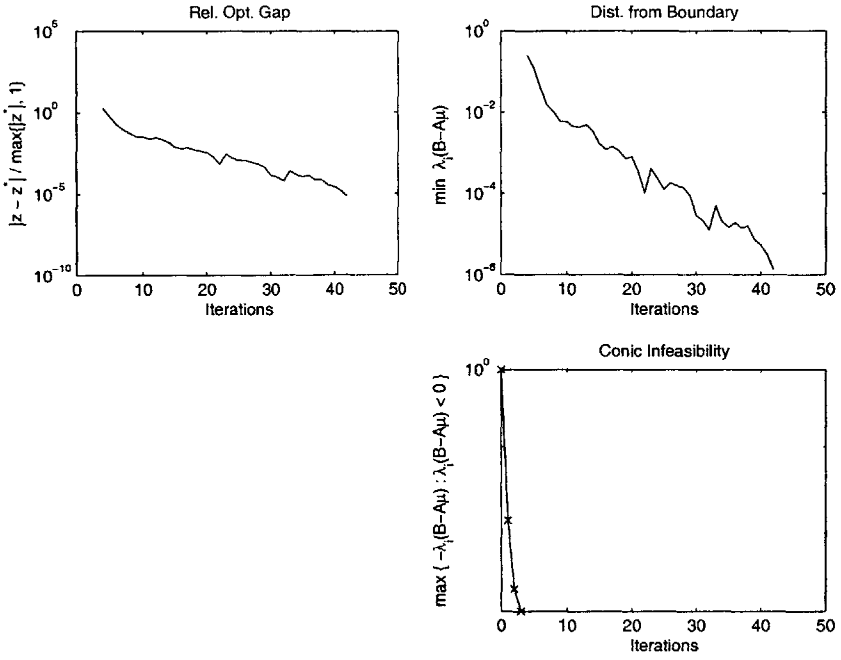

Figure B-7: SDPLIB Problem hinf1 ($n = 13$, $s = 14$)

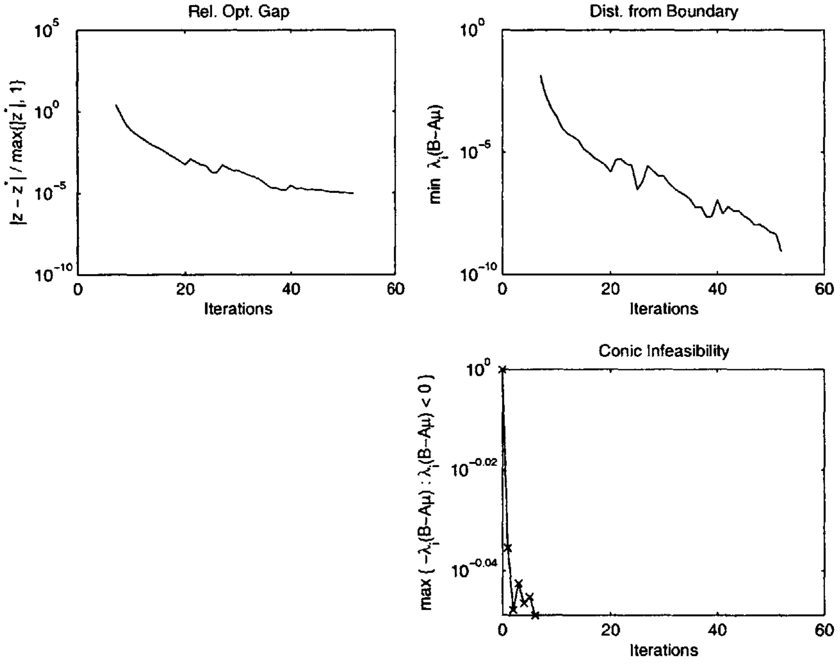

Figure B-8: SDPLIB Problem hinf2 ($n = 13$, $s = 16$)

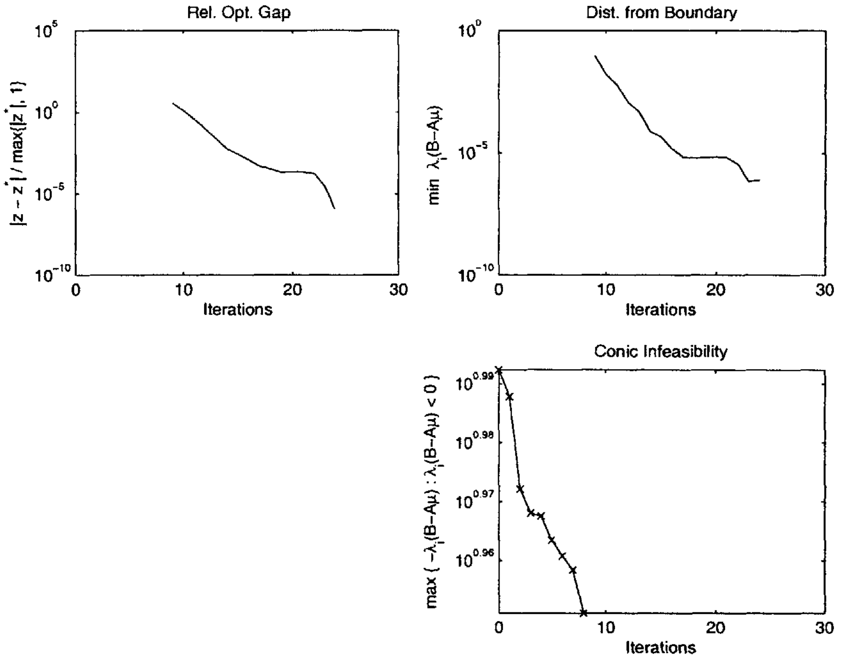

Figure B-9: SDPLIB Problem hinf3 ($n = 13$, $s = 16$)

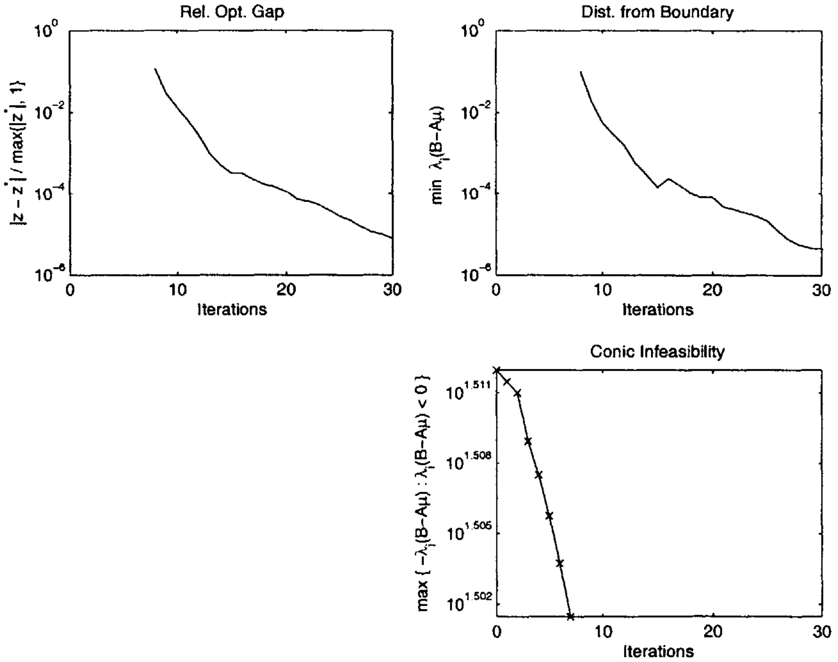

Figure B-10: SDPLIB Problem hinf4 ($n = 13$, $s = 16$)

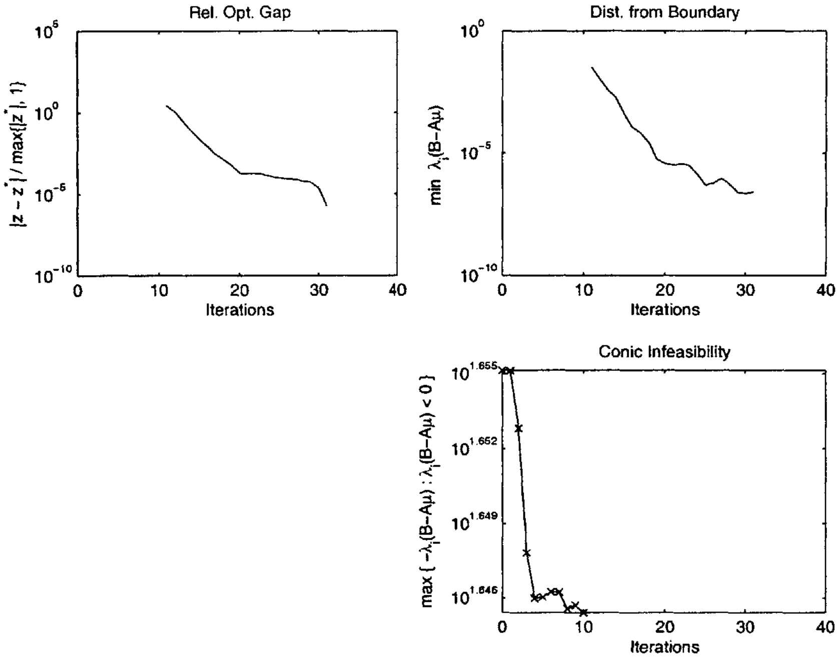

Figure B-11: SDPLIB Problem hinf5 ($n = 13$, $s = 16$)

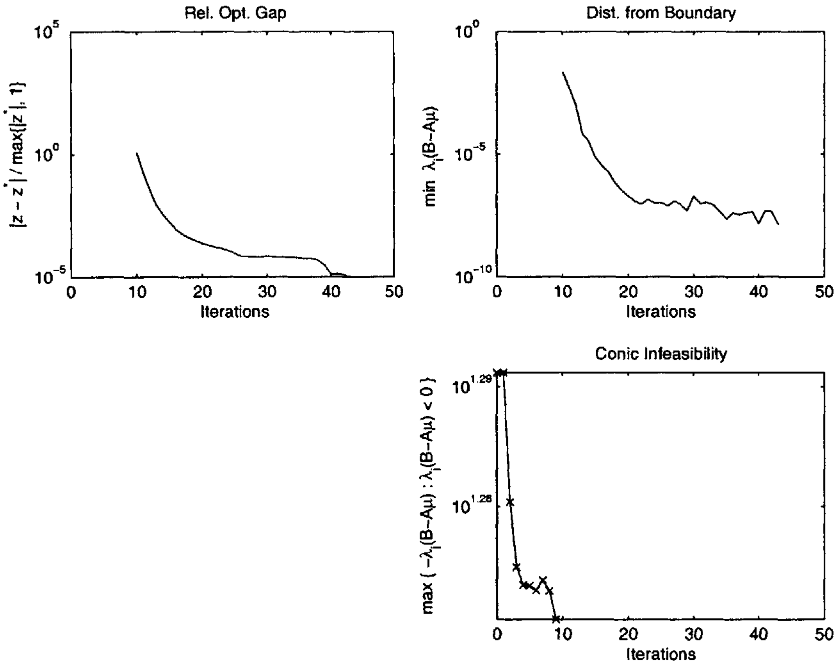

Figure B-12: SDPLIB Problem hinf6 ($n = 13$, $s = 16$)

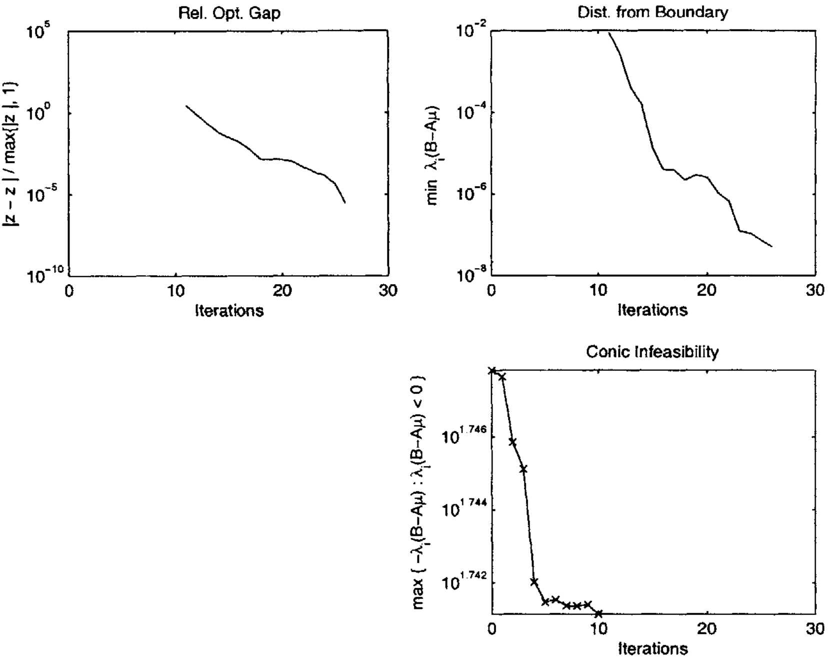

Figure B-13: SDPLIB Problem hinf7 ($n = 13$, $s = 16$)

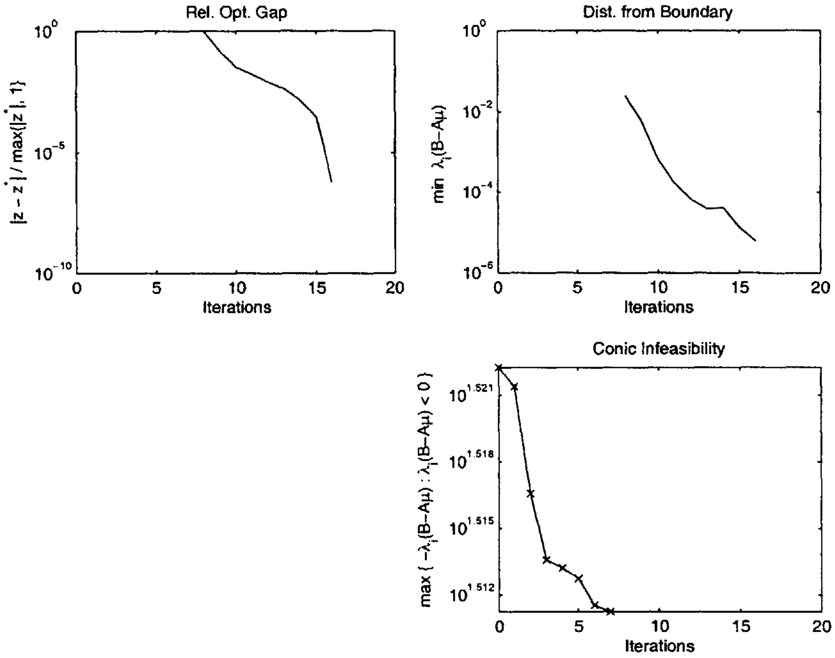

Figure B-14: SDPLIB Problem hinf8 ($n = 13$, $s = 16$)

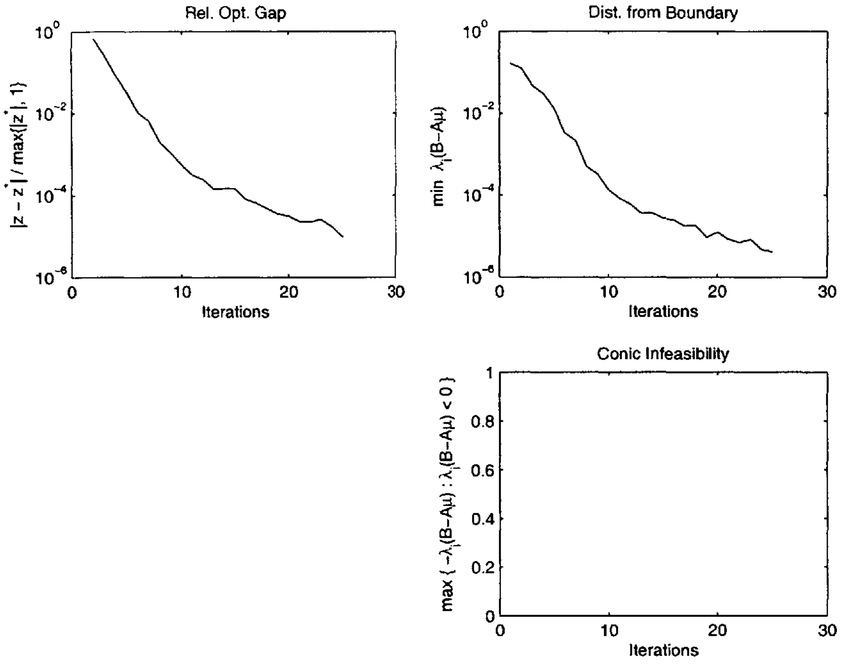

Figure B-15: SDPLIB Problem truss1 ($n = 6$, $s = 13$)

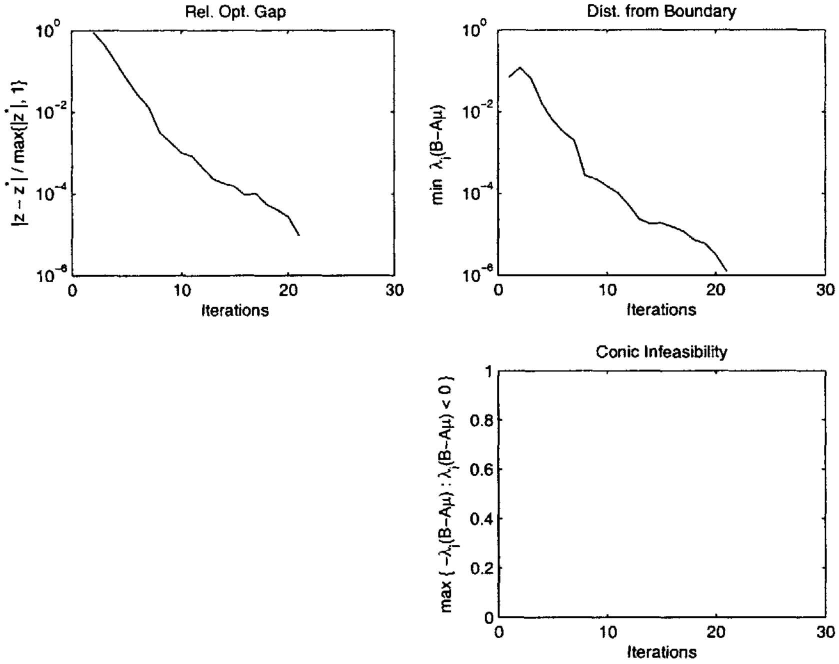

Figure B-16: SDPLIB Problem truss4 ($n = 12$, $s = 19$)

Bibliography

[1] Dimitris Bertsimas and Santosh Vempala. Solving convex programs by random walks. *J. ACM*, 51(4):540–556, 2004.

[2] Brian Borchers. SDPLIB 1.2 : A library of semidefinite programming test problems. *Optimization Methods and Software*, 11(1):683–690, 1999.

[3] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization.* Cambridge University Press, March 2004.

[4] Robert M. Freund, Fernando Ordóñez, and Kim-Chuan Toh. Behavioral measures and their correlation with ipm iteration counts on semi-definite programming problems. *Math. Program.*, 109(2):445–475, 2007.

[5] Geoffrey R. Grimmett and David R. Stirzaker. *Probability and Random Processes.* Oxford University Press, August 2001.

[6] Christian Jansson. Vsdp: Verified semidefinite programming. Technical report, Hamburg University of Technology, 2006.

[7] Adam Tauman Kalai and Santosh Vempala. Simulated Annealing for Convex Optimization. *Mathematics of Operations Research*, 31(2):253–266, 2006.

[8] László Lovász and Santosh Vempala. Simulated annealing in convex bodies and an $O^*(n^4)$ volume algorithm. In *FOCS '03: Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, page 650, Washington, DC, USA, 2003. IEEE Computer Society.

[9] László Lovász and Santosh Vempala. Hit-and-run from a corner. *SIAM J. Comput.*, 35(4):985–1005, 2006.

[10] László Lovász and Santosh Vempala. The geometry of logconcave functions and sampling algorithms. *Random Struct. Algorithms*, 30(3):307–358, 2007.

[11] Carl D. Meyer. *Matrix analysis and applied linear algebra.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.

[12] Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis.* Cambridge University Press, New York, NY, USA, 2005.

[13] Fernando Ordónez and Robert M. Freund. Computational experience and the explanatory value of condition measures for linear optimization. *SIAM J. on Optimization*, 14(2):307–333, 2003.

[14] Beresford N. Parlett. *The symmetric eigenvalue problem.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1998.

[15] James Renegar. Some perturbation theory for linear programming. *Math. Program.*, 65(1):73–91, 1994.

[16] James Renegar. *A mathematical view of interior-point methods in convex optimization.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2001.

[17] Elias M. Stein and Rami Shakarchi. *Real Analysis: Measure Theory, Integration, and Hilbert Spaces.* Princeton University Press, Princeton, NJ, USA, 2005.

[18] Kim-Chuan Toh. A note on the calculation of step-lengths in interior-point methods for semidefinite programming. *Computational Optimization and Applications*, 21:301–310, 2002.