Rounding using Random Walks - An Experimental Study

A thesis submitted in partial fulfillment of the requirements for the degree of

MASTER OF TECHNOLOGY

in

Computer Science & Engineering

by

Soumen Basu

2013MCS2584

Under the guidance of

Prof. Sandeep Sen

Department of Computer Science and Engineering,
Indian Institute of Technology Delhi.

June 2015

Certificate

This is to certify that the thesis titled **Rounding using Random Walks - An Experimental Study** being submitted by **SOUMEN BASU** (*Entry No.* 2013MCS2584) for the award of **Master of Technology** in **Computer Science & Engineering** is a record of bonafide work carried out by him under my guidance and supervision at the **Department of Computer Science & Engineering, IIT Delhi**. The work presented in this thesis has not been submitted elsewhere either in part or full, for the award of any other degree or diploma.

Dr. Sandeep Sen
Professor
Department of Computer Science and Engineering
Indian Institute of Technology, Delhi
New Delhi 110016

Abstract

We have carried out rigorous experimental analysis of iterative randomized rounding algorithms for the packing integer problem in this project. We have explored techniques based on multidimensional Brownian motion in $\mathbb{R}^n$. Let $\mathbf{x}'$ be a fractional feasible solution that maximizes a linear objective function with respect to the set of constraints $\mathbf{Ax} \leq \mathbf{1}$, $\mathbf{A} \in \{\mathbf{0},\mathbf{1}\}^{\mathbf{m}\times\mathbf{n}}$. The independent randomized rounding method proposed by Raghavan and Thompson [6] rounds each variable x_i to 1 with probability x'_i. This matches the expected value of the rounded objective function with the fractional optimum and no constraint is violated by more than $O(\frac{\log n}{\log\log n})$. Our research aims to find techniques that produce better bound than this. The experimental studies confirm that we can improve the error bound.

The first technique closely resembles the 'Edge-Walk' method proposed by Lovett and Meka [3]. We start from a fractional feasible solution, then do constrained multidimensional random walk that conforms to the constraints. Once the random walk hits a constraint A_i (or δ-close to it), it gets constrained within the hyper plane C_i that bounds A_i. The walk progresses along C_i till it hits another constraint A_j and then it is restricted within the hyperplane $C_i \cap C_j$. We proceed in this manner till the dimension becomes 0, i.e., the random walk is confined to a point. At this stage we relax the constraints by an amount Δ and repeat the procedure.

In the second technique we iteratively transform $\mathbf{x}'$ to $\mathbf{x}^*$ using random walk. This method sparsifies the constraint matrix and reduce it to a new matrix $\mathbf{A}^*$ where each constraint has no more than $\log n$ non-zero coefficients. At this point we exploit the reduced dependencies among the constraints by using the Moser-Tardos' constructive form of *Lovász Local Lemma*. For m constraints in n variables, with exactly k variables in each inequality, the constraints are satisfied within $O(\frac{\frac{\log(mk\log n)}{n}+\log\log(mn)}{\log\frac{\log(mk\log n)}{n}+\log\log(mn)})$ with high probability. For $\frac{\log(mk\log n)}{n} = o(\log n)$ this is better than the $O(\frac{\log n}{\log\log n})$ error produced by Raghavan and Thompson's method. In particular, for $m = O(n)$ and $k = polylog(n)$, this method incurs only $O(\frac{\log\log n}{\log\log\log n})$ error.

Acknowledgments

I would like to express my gratitude to my supervisor Prof. Sandeep Sen for his immense help and moral support to carry out this work. His invaluable guidance in the form of technical discussion and constructive criticism has helped me to proceed in the right direction.
I am ever grateful to my parents Mr. Jyotirmoy Basu and Mrs. Leena Basu and to my wife Arunima for their unconditional love and support. I would also like to thank my other family members and friends who were always there when I needed them the most.
Last but not the least, I bow to the omnipotent for giving me all the strength and courage to pursue my dream.

Soumen Basu

Contents

List of Figures

Chapter 1

Introduction

1.1 Problem Overview

We often come across combinatorial optimization problems that can be modeled as packing integer problems. These problems can be solved using problem specific techniques, but the methods are not generalized and they are not applicable to other problems. Thus we need some technique that can be applied grossly.
The *randomized rounding* technique is a well known generalized technique of solving the packing integer problem.

The packing integer program is described as following,

$$Max \sum_{i=1}^{n} c_i \cdot x_i$$

subject to,

$$\sum_{i \in S_j} x_i \leq 1,\ 1 \leq j \leq m,\ x_i \in \{0,1\} \forall x_i$$

Here w.l.o.g. we can say that $max_i\ c_i = 1$. The constraints are expressed as $A_j : V_j \cdot \mathbf{x} \leq 1$ where V_j is a 0-1 incidence vector corresponding to set $S_j \subset \{1, 2 \ldots n\}$.

For our analysis we have a total of m constraints and each constraint has no more than k non-zero coefficients. The constraint matrix is denoted as $\mathcal{A}_{\mathbf{k}}^{\mathbf{m \times n}} \mathbf{x^n} \leq \mathbf{1^m}$. Though the above formulation seems restrictive, we can extend our analysis to more generic problem parameters.

1.2 Similar Problems & Applications

Solving the packing integer problem has wide spread applications in computer science. We describe briefly some typical problems that map to the packing integer problem. One of such problems is the *Maximum Independent Set* problem. Our initial problem of interest was *Maximum Independent Set of Axis-Parallel Rectangles* (MISR) which eventually models into the packing integer problem.

1.2.1 Maximum Independent Set and MISR Problem

In graph theory, an independent set is a set of vertices in a graph such that no two of them are adjacent. That is, it is a set I of vertices such that for every two vertices in I, there is no edge connecting the two. The size of an independent set is the number of vertices it contains.
A maximum independent set is a largest independent set for a given graph G. The problem of finding such a set is called the maximum independent set problem.

MISR Problem

Maximum Independent Set of Axis Parallel Rectangles - MISR is the maximum cardinality subset of disjoint rectangles within the collection of axis parallel rectangles. Thus we need to find out the maximum set of rectangles such that no two of them intersect. For the 2-dimensional MISR problem, the input is a set of n rectangles in the 2- dimensional coordinate plane with each rectangle's sides parallel to one of the axis'. The task is to find the maximum cardinality set of non-overlapping rectangles.

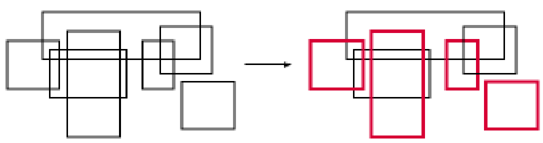


Figure 1.1: A pictorial view of the MISR problem.

This problem can be naturally generalized to n dimensions. Also, it is easy to see that MISR is a special case of the Maximal Independent Set problem with each rectangle representing a node and an edge between two nodes signifying that the corresponding rectangles overlap.

1.2.1.1 Modeling MISR into packing integer problem

An MISR problem can be modeled as an Integer Linear Programming. Given a set $S = S_1, S_2, \ldots, S_n$ of n axis parallel rectangles, for each rectangle S_i, we have a variable x_i, which can take value 1 if rectangle S_i has been included in the MIS, 0 otherwise. So, if k is the total number of maximal cliques(overlapping regions) formed and C_i denotes the set of rectangles in max-clique i, then we have the following optimization to solve -

$$Max \sum_{i=1}^{n} x_i$$

subject to,

$$\sum_{i \in C_j} x_i \leq 1,\ 1 \leq j \leq m,\ x_i \in \{0,1\} \forall x_i$$

1.2.2 Automated Label Placement

Automated label placement is an important problem in digital cartography and geographic information systems. This problem received considerable attention in recent years. The label-placement problem includes positioning labels for area, line and point features. A basic requirement in the label placement problem is that the labels are pairwise disjoint. Subject to this basic constraint, the most common optimization criteria are the number of features labeled and the size of the labels. Other variations include the choice of the shapes of the labels and the legal placements allowed for each point.

1.2.3 Network Resource Allocation

As network capabilities increase, their usage is also expanding. At the same time, the wide range of requirements calls for new mechanisms to control the allocation of network resources. However, while much attention has been devoted to resource reservation and allocation, the same does not apply to the timing of such requests. In particular, the prevailing assumption has been that requests are immediate, i.e., made at the same time the network resources are needed. This is a useful base model, but it ignores the possibility, present in many other resource allocation situations, that resources might be requested in advance of when they are needed. This can be a useful service, not only for applications, which can then be sure that the resources they need will be available, but also for the network, as it enables better planning and more flexible management of resources. Accordingly, advance reservation of network resources has been the subject of several recent studies. Now, Fundamental admission control in Networks with advance reservations is relevant in our context, because, these problems can essentially be modeled as 2- dimensional rectangle problem. we are presented with commodities (connection requests), each having a spatial dimension determined by its route in the specific topology, and a temporal dimension determined by its future duration. In this setup we can also associate a profit parameter, for which our corresponding rectangles have some weights associated to them.

1.3 Primitive Randomized Rounding

The LP is solved corresponding to the relaxation $0 \leq x_i \leq 1$ and the fractional optimum is denoted by OPT.
Let the optimum be achieved by the vector $[x'_1, x'_2 \dots x'_n]$ such that $\sum_i c_i \cdot x'_i = OPT$
Subsequently, each x'_i is rounded in the following manner,

$$\overline{x_i} = 1 \text{ with probability } x'_i,\ 0 \text{ otherwise.}$$

It can be easily shown that $\mathrm{E}[\sum_i \overline{x_i}] = OPT$ and $\forall j$, with high probability,

$$\sum_{i \in S_j} \overline{x_i} \leq \frac{\log n}{\log \log n}.$$

This *conventional(oblivious) randomized rounding* method was proposed by Raghavan and Thompson [6] in 1987.

Note: From this chapter onwards , we shall refer the Raghavan-Thompson rounding as Oblivious or, Primitive Rounding.

1.3.1 Observations

The original randomized rounding technique proposed by Raghavan and Thompson [6] was in context of the multicommodity flow problem. However the rounding bound $\frac{\log n}{\log \log n}$ was shown to be tight for congestion minimization later by Chuzhoy et al [2]. No rounding algorithm can simultaneously achieve error $e' = o(\frac{\log n}{\log \log n})$, and an objective function value of $\Omega(OPT)$ for all polynomial size input $(m \leq n^{O(1)})$.
In this context, we address the rounding problem for the *average* case of an input family that we will define more precisely in the next section 1.4. In order to decrease the error for each inequality after rounding we investigate alternate approaches based on constrained multidimensional random walk - more specifically methods based on Brownian motion. As opposed to the one-shot rounding, this rounding method is iterative and based on successive refinements of the LP solution.

1.4 Overview of our problem setup

The overview of our problem setup is discussed below. The description is due to Sen and Madan [9]. We would provide a more formal definition in chapter 4.2. Let us briefly describe the characteristics of the LP used for conducting the experiments. Let $\mathcal{A}_k^{m \times n}$ denote the family of $m \times n$ $0-1$ matrices where each row independently has 1's in k randomly chosen columns from $\{1, 2, \ldots n\}$ and 0 elsewhere. Let $\mathbf{A} \in \mathcal{A}_k^{m \times n}$, then for any point $\mathbf{x}' = (x'_1, x'_2, \ldots x'_n)$, $0 \leq x'_i \leq 1$ such that $\mathbf{A} \cdot \mathbf{x}' \leq 1^m$, $\mathbf{x}'$ can be rounded to $\hat{\mathbf{x}} \in \{0, 1\}^n$ such that the following holds with high probability,

$$\|A \cdot \hat{x}\|_\infty \leq O\left(\frac{\log(mk\log n/n) + \log\log m + \log\log n}{\log\log(mk\log n/n + \log m \log n)}\right)$$

and,

$$\sum_i \hat{x}_i \geq \Omega(n/k)$$

For $k = \log n$, $x'_i = 1/\log n$, $c_i = 1$, (unweighted case), and $m \leq n \cdot polylog(n)$, with high probability our algorithm outputs a vector $\hat{x}$ that contains $\Omega(n/\log n)$ 1's such that $\|A \cdot \hat{x}\|_\infty \leq O(\log\log n/\log\log\log n)$. This is a significant improvement over the $O(\frac{\log n}{\log\log n})$ bound of the oblivious rounding.

The result relates the rounding error to the average number of ones in a column, i.e., mk/n and characterizes tradeoffs between the parameters m, n, k. For example, for $k = \sqrt{n}$, we can bound the error to $O(\log\log n)$ for $m \leq \sqrt{n} polylog(n)$.

Just to sum uo the above discussion, we give the definition of our LP below,

$$Max \sum_i x_i$$

subject to,

$$\mathbf{A_k^{mxn}} \cdot \mathbf{x^n} \leq \mathbf{1^m}, x_i \in \{0, 1\} \forall x_i$$

m: No. of inequalities
n: No. of variables
And, $\sum_j A_{ij} = k \quad \forall A_i$

We conducted several experiments by varying the values of m, n and k. The number of variables, n was varied between 100 and 1000. The number of inequalities, m was kept between $O(n)$ and $O(n \log n)$ and the number of variables, k in each inequality was varied between $\log n$ to $\log^2 n$. Due to the limitation of scope we present results only for $n = 100$ and $m = 150, 700$ and $n = 1000$, $m = 1500, 9000$.

The problem is actually a multi-objective optimization, where we simultaneously want to minimize the *error* and maximize the objective function.

1.5 Prior related work

There exists a very extensive and rich literature on the use of randomized rounding for solving various generalizations of the edge-disjoint path problem starting with the seminal paper of Raghavan and Thompson [6]. Recently a number of techniques for rounding have emerged that are iterative in nature. The recent work on constructive discrepancy perhaps comes closest in terms of randomized iterative rounding. Bansal's [1] seminal paper on a constructive proof of Spencer's discrepancy theorem is based on rounding solutions of a semi-definite program that captures the discrepancy constraints. This method was further refined and simplified in the work of Lovett and Meka [3] who derived an alternate proof based on a very elegant analysis of a multidimensional random walk. The crux of the method called *partial coloring lemma* is a rounding strategy of an arbitrary $x \in [-1, +1]^n$ vector within the discrepancy polytope defined by the constraints starting with $x = 0^n$. It is done over logarithmic phases where in every phase approximately half the coordinates converge to $\{-1, +1\}$ (actually arbitrarily close to $\{-1, +1\}$) - this aspect is similar to Bansal [1]. Their method can be mapped to $\{0, 1\}$ rounding as well, as observed by Rothvoss [7].

Chapter 2

Iterative Rounding based on Random Walks

In this chapter we intend to familiarize the reader with the idea of random walk based rounding techniques. We begin with the description of primitive iterative rounding processes and then talk about their shortcomings. In the subsequent chapters we discuss the methodologies that overcome these shortcomings. The random walks that we refer to are Brownian motions in n dimensions where every step is a random variable chosen from a multidimensional Gaussian distribution. Each dimension is chosen independently from a standard normal distribution $\mathcal{N}(0,1)$. The random walk based rounding gradually converges to a solution in contrast to the classical one shot rounding. We briefly describe two basic types of random walk,

1. Independent random walk based rounding
2. Dependent (constrained) random walk based rounding

2.1 Independent Iterative rounding

2.1.1 Basic framework

The method propsed by Raghavan and Thompson uses one step rounding. Instead of using such one step method to round each x'_i to 0 or 1, we gradually add a small step $\pm\gamma$, $0 < \gamma < 1$ in an iterative manner to the variables. This γ is generated over a normal distribution. The sign of γ is also a random variable. We are effectively slowing down the oblivious rounding.
The parameters γ, δ are chosen to ensure that the walk stays within the feasible region. It suffices to have $\gamma \leq \frac{\delta}{\log n}$ from the pdf of the normal distribution

if we are executing $O(\frac{1}{\gamma^2})$ steps [3]. The value of δ will be determined according to the approximation factor and the error bound that we have in a specific application. In our analysis, the focus will be on variables absorbed at 0 that will be rounded down. So the loss in the objective value can be at most $n\delta$ (recall the maximum weight of the coefficients is 1). If we choose δ such that $n\delta$ is $o(OPT)$ it will suffice. For our framework, choosing $\delta \leq \frac{1}{polylogn}$ will work.
Let us give an outline of the independent random walk based rounding,

Algorithm 1: Iterative Rounding based on Independent Random walks

Data: $x'_i, 1 \leq i \leq n$ satisfying the LP

Result: $\overline{x}_i \in 0, 1$

Initialization: set all variables as *un-fixed* and set $X_0 = [x'_1, x'_2 \ldots x'_n]$;

while *not all variables are fixed* **do**

  Generate a random vector $R_i^n = U_i$ where U_i is a standard multidimensional Normal distribution.;

  $X_{i+1} = X_i + \gamma \cdot R_i^n$;

  if $0 \leq x_k \leq \delta$ OR $1 - \delta \leq x_k \leq 1$ **then**

    fix x_k;

  end

end

We will soon show that the probability distribution of independent rounding is actually the same as oblivious rounding method. Hence we can not use independent rounding as a standalone technique. In contrast to the independent walk, the 'dependent' walk constrains the multidimensional walk to the Nullspace of the constraint matrix. We shall show some experimental analysis regarding the independent rounding and consequently how the 'dependent' rounding benefits us. Then we will talk about the limitations of the dependent rounding and subsequently move on to the next chapter.

2.1.2 Limitations of independent walk based rounding

The simple random walk based algorithm outlined in the introduction doesn't take into account any of the constraints $V_j \cdot \mathbf{x} \leq \mathbf{1}$ and therefore likely to violate them depending on the number of steps T. However, the probability that an x'_i reaches 1 before it reaches 0 is equal to x'_i. This follows from a more general property of martingale known as Doob's Optional Stopping Theorem.

Theorem 2.1.1. *Let $(\Omega,\ \Sigma,\ P)$ be a probability space and $\{F_i\}$ be a filtration of Ω, and $X = \{X_i\}$ is a martingale w.r.t $\{F_i\}$. Let T be a stopping time such that $\forall\omega \in \Omega$, $\forall i\ \ |X_i(\omega)| < K$ for some positive integer K, and T is almost surely bounded. Then $\mathbb{E}[X_T] = \mathbb{E}[X_0]$.*

In the above application, the stopping times are $X_T = \{-b, a\}$ whichever is earlier. So $-b \cdot Pr[X = -b] + aPr[X = a] = 0 \cdot Pr[X = 0]$. Since $Pr[X = -b] + Pr[X = a] = 1$ from the stopping criteria, $Pr[X = a] = \frac{b}{a+b}$. In our context, we start from $X = x'_i$ and $b = \frac{x'_i}{\gamma}$ and $a = \frac{1-xi_i}{\gamma}$, so the probability that $\overline{x}_i = 1$ equals $\frac{x'_i/\gamma}{(1-x'_i+x'_i)/\gamma} = x'_i$. This shows that this basic approach is not very different from the oblivious randomized rounding and similar bounds can be proved. Also, if the initial value of a variable is x_i $0 \leq x_i \leq 1$, then probability of hitting 1 before 0 is x_i.

Suppose we start random walks for $\log n$ variables. Each variable was initialized to $\frac{1}{\log n}$. For independent rounding the probability that $\frac{\log n}{\log\log n}$ of these variables hit 1

$$= \left(\frac{1}{\log n}\right)^{\frac{\log n}{\log\log n}} \quad = \left(\frac{1}{2^{\log\log n}}\right)^{\frac{\log n}{\log\log n}} \quad = \Omega\left(\frac{1}{n}\right)$$

So, this calculation rules out the possibility that independent rounding will succeed with probability $1 - \frac{1}{n}$.

Convergence rate

We observed the rate at which the variables get converged to either 0 or 1. We define U_t as the number of non-convergent variables at step t. Following figure depicts one such result,

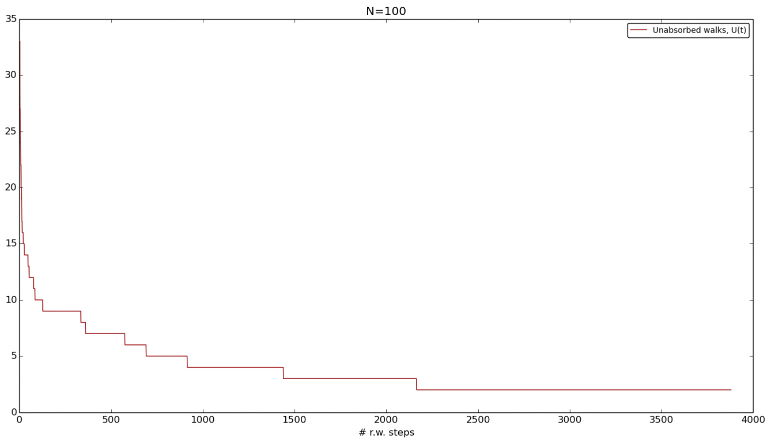


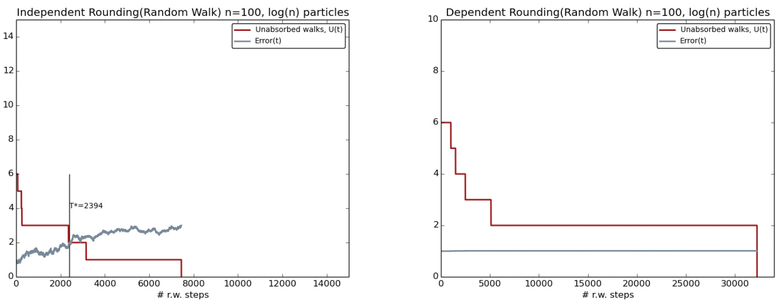
Figure 2.1: Convergence rate of variables

2.2 Dependent rounding

In this section we discuss the constrained Gaussian walks and the observations that we made in this context. We start our discussion by exploring the technique to solve an LP with a single constraint (we call this scenario as a single inequality, though it is a slight abuse of mathematical terms).

2.2.1 Solving a single inequality

For a single inequality, we do the 'dependent' walk by generating a multidimensional Gaussian on the orthonormal basis of the subspace of the inequality and then projecting back the steps in the original coordinate system. This will ensure that the inequality is never violated. We also show the comparison with independent walk 2.2.



(a) Rounding using independent gaussian walk

(b) Rounding using constrained gaussian walk

Figure 2.2: Solving a single inequality - using random walk based rounding

It is evident that in independent walk we can violate the constraints since the random walks can go outside the polytope constructed by the constraints. But that will never happen in 'dependent' random walk. It also turns out that independent walk converges much faster than the dependent walk. It is basically a trade off between time and quality of solution. The decision of choosing a method to solve the problem would be influenced by the specific demands of the situation in hand.

2.2.2 Solving an LP

The dependent walk does not violate the inequalities. So, we investigated this process further. In case of an LP, if we do not want to violate the constraints, we must confine our walk on the Nullspace of the constraint matrix.To solve a system of inequalities, we follow the following procedure,

1. Calculate the Nullspace $N(A)$ of the inequalities. Consider only alive variables
2. Generate a multidimensional Gaussian R^i constrained to $N(A)$.
3. $X_{i+1} = X_i + \gamma \cdot R^i$
4. If $0 \leq x_j \leq \delta$ OR $1 - \delta \leq x_j \leq 1$ *fix* x_j
5. Repeat till all x_i have converged.

2.2.2.1 Calculation of the constrained Gaussian

We calculate the Gaussian constrained to a subspace in the following manner. Suppose the constrained subspace is R^d with orthonormal basis $w = [w_1, w_2, \ldots]$ and V is the original axis of the variable walk. Now we can generate $\mathcal{N}(0, 1)$ along each of these orthonormal basis and project back the Gaussian in the original axis using the following formula,
$\mathcal{N}(0, \sum_i \alpha_i^2)$ where $\alpha_i =< w_i, V >$

2.2.3 Limitations of dependent walk based rounding

Although the dependent rounding seems very promising, it suffers from a major drawback. This method only works if the number of inequalities is smaller than the number of variables. Otherwise, $N(A) = \phi$. And we have no subspace to generate our walk. So, whenever $m > n$, this method fails. And even if initial value of n is greater than m, the nullspace dimension eventually becomes higher than the effective variable dimesion and the process eventually stalls.

So, it turned out even the dependent walk was unable to provide us a routine to solve the rounding problem. We started looking for refinements of these methods from this point and we would verse the reader with our findings in the following chapters.

Chapter 3

Edge Walk based Rounding

Using a method proposed by S. Lovett and R. Meka [3], which they call Edge-Walk, we came up with a randomized approximation algorithm. First we obtain a fractional feasible solution. Then we use a new approach named as Random Walk on Edges [3] for rounding.

3.1 Edge walk

The idea is to do a constrained walk on polytope $\mathcal{P}$, where $\mathcal{P}$ can be described as,
$\mathcal{P} = \{x \in R^n : |x_i| \leq 1, \forall i \in \{1, \ldots, n\}, | < x - x_0, v_j > | \leq 1 \ \forall j \in \{1, \ldots, m\}\}$.

The main essence of the idea is to discover the *null space* iteratively. And when the nullspace dimension becomes higher than the unabsorbed variable dimension, we expand the polytope by some factor Δ_j.

3.1.1 Differences with original edge walk

There are some points where our problem differs from the original edge walk proposed by Lovett and Meka.

- We do not start the walk from origin. In their proposed method, Lovett and Meka began the walk from 0^n. Our starting point is the fractional feasible solution of LP.
- It was fine for Lovett and Meka to come up with a $\sqrt{n}$ bound for discrepancy problem, whereas for our problem, we are looking forward to achieving better bounds in terms of *log* or *loglog* of the problem

size. So, our approach to the problem and analysis is a bit different from theirs.

- Our walk is in range $0, 1$ while theirs is $-1, 1$. But this is not much of a problem. Using simple transformations [7] we can reduce $0-1$ walk to $-1, 1$ walk and vice-versa.

3.2 Our Algorithm

Our algorithm conforms to the constraints while doing the random walk and will not violate them. Once the random walk hits a constraint (or δ-close to it), the random walk will be constrained within the hyperplane C_i bounding the constraint. The walk progresses along C_i till it hits another constraint C_j and then it is restricted within the hyperplane $C_i \cap C_j$, and so on till the dimension becomes 0, i.e., the walk is confined to a point. At this point, we will do a hypothetical scaling of the feasible polytope $\mathcal{P}$ to $\mathcal{P}^1$ by an appropriate amount $\Delta_1 > 0$. We refer to it as 'expansion' of the polytope. This will be achieved by relaxing the constraints $V_j \cdot \overline{x} \leq 1 + \Delta_1$. So, X_0 now becomes an interior point in $\mathcal{P}^1$, say v^1. We repeat the whole procedure till all the variables converge. We describe the modified approach below in the next algorithm. Let G_n denote an n-dimensional Gaussian random variable with mean 0 and variance 1, and let U_t denote the projection of G_n on the current (restricted) subspace in step t of random walk.

Algorithm 2: EdgeWalk based Rounding

Data: $x'_i \quad 1 \leq i \leq n$ satisfying the LP
Result: $\overline{x}_i \in 0, 1$
Initialization: $X_{0,0} = [x'_1, x'_2 \ldots x'_n]$, where $X_{i,j}$ denotes the position of the walk in the i^{th} iteration and jth phase ;
for *each phase* $j = 0, 1, 2 \ldots$ **do**
    Start independent random walk till any constraint C_k is 'nearly hit' (δ-close). Set $S_0 = C_k$;
    for *iteration* $i = 0, 1, 2, \ldots$ **do**
        Generate a random vector U_i within current subspace S_i. ;
        $X_{i+1,j} = X_{i,j} + \gamma \cdot U_i$;
        if $X_{i+1,j}$ *is* δ*-close to constraint* C^i_k **then**
            Modify subspace of the walk S_{i+1} restricted to $C^i_k \cap S_i$.;
        end
        if $x_k \leq \delta$ *OR* $x_k \geq 1 + \Delta_j - \delta$. **then**
            fix x_k ;
            If all variables are fixed then exit;
        end
        if X_{i+1} *is now a point (dimension 0)* **then**
            relax the constraints C by Δ_j;
            reset to the beginning of the phase $j + 1$;
        end
    end
end

3.3 Expansion of Polytope

In the process of edge-walk, it remains a major question that what should be the rate of expanding the polytope, i.e. what should be a good Δ_i at i-th iteration? In the initial stage of our work we tried with some adhoc values of Δ. But in the later stages of development, we conducted several experiments on this and tried to conceive an acceptable function for Δ_i.

Grossly, there would be three basic types of expansion function based on the rate of expansion.

1. Convex or increasing rate of polytope expansion
2. Linear or constant rate of polytope expansion
3. Concave or decreasing rate of polytope expansion

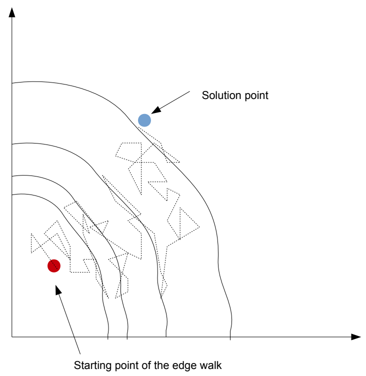
We show a brief comparative study of these three types of expansion functions and then it would be clear which function should be chosen and what trade-offs we need to make.

3.3.1 Convex polytope expansion

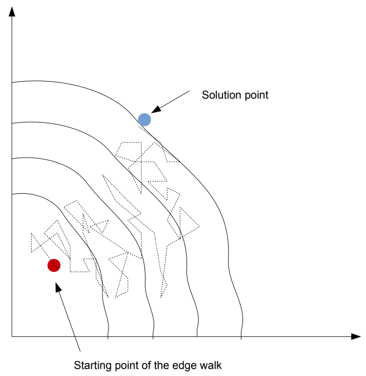
In this type of expansion the growth of the polytope is more in phase-p than it was in phase $p-1$. A pictorial depiction 3.1 is given below for reader's reference. These type of expansion seems very practical because it allows more and more space for the random walks in later phases, thus eventually making the convergence time faster. On the other hand there is a possibility to overshoot the error limit. There can be numerous such possible functions. We chose $\Delta_p = c \cdot O(p^\alpha)$, where p is the current phase of the Edge walk and $0 < c < 1$ is some constant. We did many experiments varying the value of $\alpha > 1$. Due to the limitation of scope we present results for $\alpha = 2$.

3.3.2 Linear polytope expansion

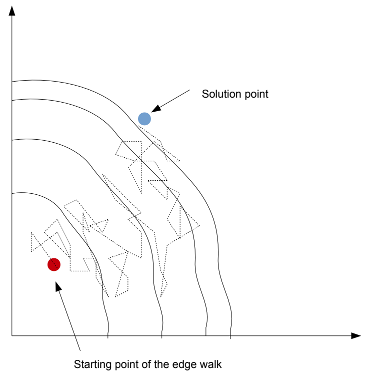
The linear expansion seems to be in the middle of two extremes. We let the polytope grow uniformly in this scheme. The uniform growth doesn't make the polytope go too beyond the error bound. In terms the random walk also doesn't speed up in later phases. Basically we use a growth function in linear order of phase. We used $\Delta_p = \Delta_{p-1} + c$ to grow the polytope linearly. We would present some results for all these growth functions soon.



(a) Convex expansion



(b) Linear expansion



(c) Concave expansion

Figure 3.1: An abstract pictorial depiction of different type of polytope expansions

3.3.3 Concave polytope expansion

This is the final variant of the polytope expansion function. In this type of function the rate of growth of polytope decreases in each phase. While this seem to produce a good error bound, the random walk gets terribly slow since the space to move becomes less and less in successive phases. Also there is another danger of not growing the polytope more than δ(the tolerance parameter) and γ(the step size). That is $|d\Delta| < \gamma, \delta$. This means that the random walk will stall and we need to restart with new control parameters. We experimented with two such functions $\Delta_p = c \cdot \log p$ and $\Delta_p = c \cdot p^{\beta}$, where $0 < c, \beta < 1$ and p is the current phase of random walk. Due to the limitation of scope, we present result for $\Delta_p = c \cdot p^{0.5}$.

3.3.4 Experimental Observation

We present the results in the following figures (3.2, 3.3). The first table shows results for worst case convergence time in terms of random walk steps and the second table shows frequency distribution of error for $n = 100,\ 1000$ variables. For both cases $m = O(n)$.

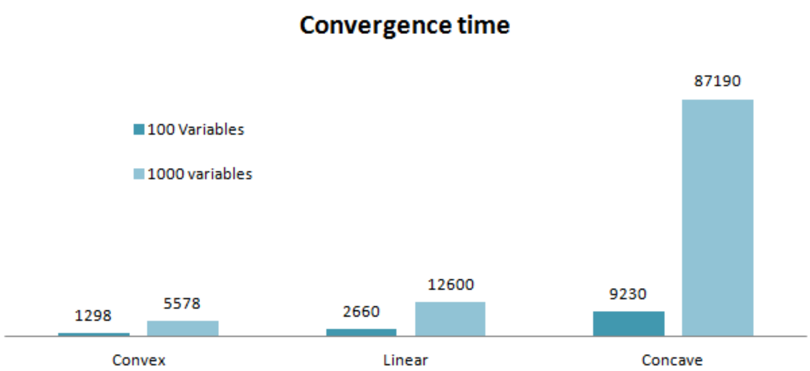
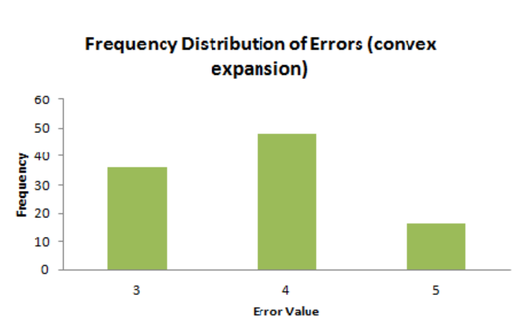
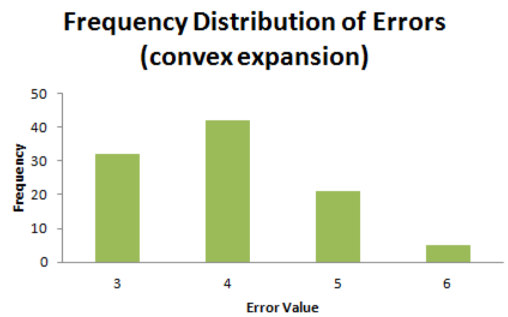


Figure 3.2: Convergence time for different type of polytope expansion

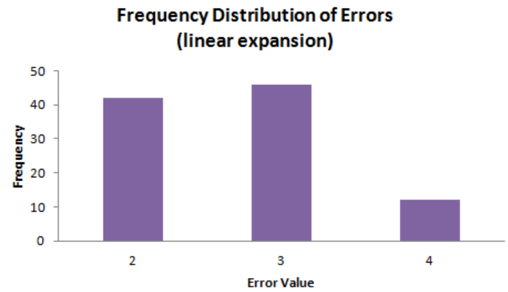
The obtained results supports our intuition. The convex expansion is the fastest but it results in highest amount of error. But in our context, the difference in error is negligible. Hence we adopted the convex expansion technique. The method takes 75 s, 4.76 m and 1.17 h for convex, linear and concave expansion for 1000 variables.



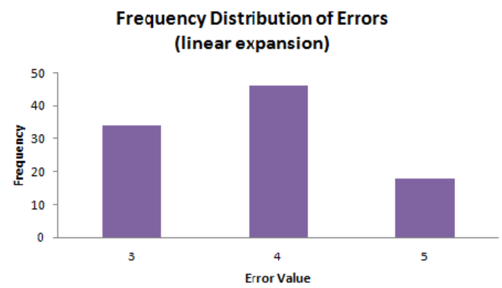
(a) Convex expansion 100 vars



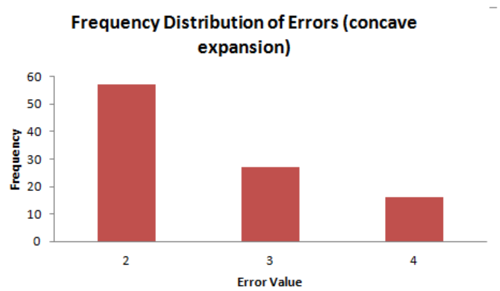
(b) Convex expansion 1000 vars



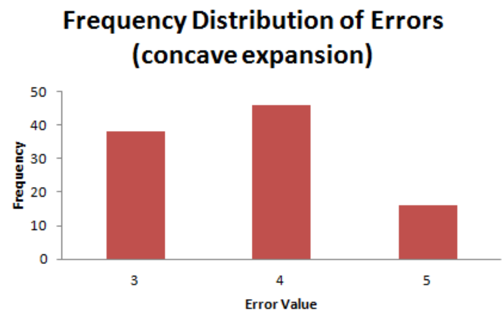
(c) Linear expansion 100 vars



(d) Linear expansion 1000 vars



(e) Concave expansion 100 vars



(f) Concave expansion 1000 vars

Figure 3.3: Frequency distribution of error for different types of polytope expansions

Chapter 4

Rounding using Lovász Local Lemma

4.1 Lovász Local Lemma

The Lovász Local Lemma by László Lovász and Paul Erdös [5] provides a strategy for proving the existence (or non-existence) of an object that satisfies some prescribed property. Generally, the argument involves selecting an object randomly from a specific set and demonstrating that it has the desired property with strictly positive probability. This in turn proves the existence of at least one such object.

Theorem 4.1.1 (Lovász Local Lemma)**.** *Let A_i, $1 \leq i \leq N$ be events such that $Pr[A_i] = p$ and each event is dependent on at most d other events. Then if $ep(d+1) < 1$, then*

$$Pr(\bar{A}_1 \cap \bar{A}_2 \ldots \bar{A}_N) > 0$$

Alternately, in a more general (asymmetric) case, where the dependencies are described by a graph $(\{1, 2 \ldots N\}, E)$ where an edge between i, j denotes dependency between A_i, A_j and y_i are real numbers such that

$Pr(A_i) \leq y_i \cdot \prod_{(i,j)\in E}(1 - y_j)$ then

$$Pr\left(\bigcap_{i=1}^{N} \bar{A}_i\right) \geq \prod_{j=1}^{N}(1 - y_j)$$

Moreover, such an event can be computed in randomized polynomial time using an algorithm of Moser and Tardos [4].

Remark The symmetric case follows from the general case by using $y_i = \frac{1}{d+1}$

4.2 Rounding using Brownian walk followed by LLL

In the first stage of Iterative Randomized Rounding, we run the Independent walk based rounding algorithm (Chap.2) till all constraints have less than $\log n$ non-zero variables. Note that each step of the Brownian motion takes O(mn) time, to inspect all the unfixed variables in each constraint.
Consider the matrix $A_{m\times n}$ has no more than ρ 1's in any column. Let OPT be the optimal fractional objective value for the weighted objective function $\sum_i c_i \cdot x_i$. Consider a fixed row r_i, that contains $\log n$ 1's and let $j_1(r_i), j_2(r_i)\ldots$ denote the columns that contain 1. We say that r_i is dependent on another row r_j if they share at least one column where the value is 1. So, the dependency of any row can be bounded by $\rho \log n$. If we use the oblivious randomized rounding to round the fractional solution $\mathbf{x}'$, the probability that the value of a row exceeds $t > 1$ is bounded by $\frac{1}{2^t}$ from Chernoff bounds. Let E_i denote the event that row i exceeds t when we use randomized rounding. We are interested to know the probability of the event which implies that all the inequalities are less than t. Such an event is given by,

$$\bigcap_{1\le i\le m} \bar{E}_i$$

This is an ideal situation to apply LLL.

In our context, $d = O(\rho \log n)$. If we choose t such that $e \cdot 2^{-t} \cdot (d+1) \le 1$ or equivalently $t = \log d$, then we can apply the previous theorem to obtain a rounding that satisfies an error bound of $O(\log \rho + \log\log n)$. For ρ bounded by $n \log^{O(1)} n$ this is $O(\log\log n)$ which is substantially better than the Raghavan-Thompson bound.

Sen [8] points out that the above argument is incomplete since we also want to guarantee a large value of the objective function. For example, setting all variables equal to zero gives a feasible solution but also returns an objective function value 0. For the sake of completeness, we provide the analysis presented by Sen [8]. We define an additional event A_{m+1} corresponding to the objective function value less than $(1-\alpha)\cdot OPT$ for some suitable $1 > \alpha > 0$. Since A_{m+1} is a function of all the variables, it has dependencies with all other $A_i \quad i = 1 \ldots N$, therefore we have to use the generalized version of LLL in this case. From Chernoff-Hoeffding bounds, $\Pr(A_{m+1}) \leq exp(-\epsilon^2 OPT/2)$. We define y_i for $i = 1, 2 \ldots m$ corresponding to the probability of exceeding $t = \log d / \log \log d$. By choosing $y_i = 1/(\alpha d) \quad i \leq m$ and $y_{m+1} = exp(-\epsilon^2 OPT/2)$, for some suitable scaling factor $\alpha \geq e$, we must satisfy the following inequalities

$$\Pr(A_i) \leq 1/(\alpha d)(1 - 1/(\alpha d))^d \cdot (1 - exp(-\epsilon^2 OPT/2)) \quad i = 1, 2 \ldots m$$
$$\Pr(A_{m+1}) \leq exp(-\epsilon^2 OPT/2)(1-1/(\alpha d))^m \leq exp(-\epsilon^2 OPT/2) \cdot exp(-m/(\alpha d))$$

The first condition is easily satisfied when $\Pr(A_i) = \frac{1}{\Omega(\alpha d)}$. To satisfy the second inequality, we can choose $\epsilon' > \epsilon$ so that $\Pr(A_{m+1}) \leq exp(\epsilon'^2 OPT/2)$ and assume that $OPT \geq \Omega(\frac{m}{\alpha d})$. This implies that $exp(-m/(\alpha d)) \geq exp(-\epsilon^2 OPT/2)$, so the condition is satisfied for LLL to be applicable.

Theorem 4.2.1 (Sen and Madan [9]). *Let $\mathcal{A}_k^{m \times n}$ denote the family of $m \times n$ matrices where each row (independently) has k ones in randomly chosen columns from $\{1, 2, \ldots n\}$ and 0 elsewhere*[1]*. Let $A \in \mathcal{A}_k^{m \times n}$, then for any point $x' = (x'_1, x'_2 \ldots x'_n)$, $0 \leq x'_i \leq 1$ such that $A \cdot x' \leq e^m$, where e^m is a vector of m 1's, $\bar{x'}$ can be rounded to $\hat{x} \in \{0, 1\}^n$ such that the following holds with probability $\geq 1 - \frac{1}{m}$*

$$||A\hat{x}||_\infty \leq O\left(\frac{\log(mk \log n/n) + \log\log m + \log\log n)}{\log\log(mk \log n/n + \log m \log n)}\right) \text{ and } \quad (4.1)$$
$$\sum_i \hat{x}_i \geq \Omega(n/k) \quad (4.2)$$

Let $\bar{c} = (c_1, c_2..c_n)$ such that $1 = \max_i\{c_i\} \geq \ c_i \geq \frac{1}{\log n}$. If x^* maximizes $< \bar{c}, \bar{x} >$ then it can be rounded to $\hat{x}$ satisfying (4.1) and

$$< \bar{c}, \hat{x} > \ \geq \frac{1}{C} < \bar{c}, x' > \text{ for some constant } C$$

Moreover such an $\hat{x}$ can be computed in randomized polynomial time.

[1] Alternately we can set every entry to be 1 with probability k/n independently

Remark For $k = \log n$, $x'_i = 1/\log n$, $c_i = 1$, (unweighted case), and $m \leq n \cdot polylog(n)$, with high probability our algorithm outputs a vector $\hat{x}$ that contains $\Omega(n/\log n)$ 1's such that $||A\hat{x}||_\infty \leq O(\log\log n/\log\log\log n)$. This is a significant improvement over the $O(\frac{\log n}{\log\log n})$ bound of the oblivious rounding.

4.2.1 LLL based iterative randomized rounding

We would provide a polynomial time algorithm based on the Moser-Tardos' constructive form of LLL [4]. We start by presenting the algorithm for randomized rounding based on LLL. Then we would show some experimental results.

Algorithm 3: Randomized Rounding based on LLL

Data: $x'_i \quad 1 \leq i \leq n$ satisfying the LP and t (error parameter)
Result: $\hat{x}_i \in \{0, 1\}$
Do oblivious rounding on all the variables having values x'_i to $\hat{x}_i$;
Compute the value of each constraint C_i as $< V_i, \hat{x} >$;
while *any inequality exceeds* t *OR objective value* $< OPT/2$ **do**
  1. Pick an arbitrary constraint C_j that exceed t and perform oblivious rounding on all the variables in V_j ;
  2. Update the value of the constraints whose variables have changed ;
end
Return rounded vector $\hat{\mathbf{x}}$

The error bound is obtained by setting $x_i = \frac{1}{\Omega(\alpha d)}$. The second condition in above discussion can be satisfied by ensuring $OPT \geq \frac{m}{(\alpha d)}$ by choosing an appropriately large α. Since $OPT \geq 1$ ($max_i\ c_i = 1$) we can bound $\alpha \leq m/d = m/(\rho \log n)$. The objective function still follows the martingale property since the variables starting the random walk at x'_i have probability x'_i of being absorbed at 1 which is identical to the oblivious rounding that we use in the Moser-Tardos algorithm. Although, the random walk is short-cut by a single step oblivious rounding, the distribution for absorption at 0/1 remains unchanged. The rest follow from LLL and the algorithm of

Moser-Tardos. The expected running time is $O(\frac{m^2}{n} polylog(n))$ in our case.

Note: A direct application of LLL (without running the Brownian motion) would increase the dependence to $O(\rho \cdot n)$ resulting in an error bound that is much larger.

We present the results in the next chapter in a compact and concise manner. [5.2, 5.3.1]

Chapter 5

Comparative study between randomized rounding techniques

5.1 Experimental set-up

We have already given the experimental setup in the first chapter 1.4. Nevertheless, we briefly describe the set up once again.

$$Max \sum_i x_i$$

subject to,

$$\mathbf{A_k^{mxn}} \cdot \mathbf{x^n} \leq \mathbf{1^m}, x_i \in \{0,1\} \forall x_i$$

m: No. of inequalities
n: No. of variables
And, $\sum_j A_{ij} = k \ \ \forall A_i$
$\delta \leq \frac{1}{polylogn}$: Tolerance value for absorption of variables
$\gamma \leq \frac{\delta}{\log n}$: The scaling factor for the Gaussian steps
$\Delta_p = c \cdot p^2$: Polytope expansion factor for Edge-walk, p is current phase.
t: Error parameter for *Lovasz Local Lemma (LLL)*

We conducted several experiments by varying the values of m, n and k. The number of variables, n was varied between 100 and 1000. The number of inequalities, m was kept between $O(n)$ and $O(n \log n)$ and the number of variables, k in each inequality was varied between $\log n$ to $\log^2 n$.

The problem is actually a multi-objective optimization, where we simultaneously want to minimize the *error* and maximize the objective function. To make a compact comparison, we can do either of the following -

1. Filter out those results for which objective function value $> c \cdot \Omega(n/k)$, where $0 < c < 1$ is a constant, and then compare the minimum *error* values. W.l.o.g. we chose $c = \frac{1}{2}$.

2. Filter out the results for which the $error \leq \beta$, where $\beta > 1,\ \beta \in \mathbb{Z}^+$, and then compare the maximum objective function values obtained.

Each experiment was repeated 100-times and the best result was chosen.

5.2 Comparing rounding methods in terms of error

In this section we show the comparison of different rounding methods in terms of the minimum error bound. Like we mentioned in the previous section, we filter out the results where the objective function $\geq \frac{1}{2}OPT$. The following two tables 5.1, 5.2 would depict the results.

	Oblivious Rounding	Edge walk based method	Rounding using LLL	Iterative rounding followed by LLL
$k = \log n$	3	2	2	2
$k = \log^{1.5} n$	3	2	2	2
$k = \log^2 n$	2	2	2	2

(a) $m = 150,\ n = 100$

	Oblivious Rounding	Edge walk based method	Rounding using LLL	Iterative rounding followed by LLL
$k = \log n$	3	2	2	2
$k = \log^{1.5} n$	3	2	2	2
$k = \log^2 n$	2	2	2	2

(b) $m = 700,\ n = 100$

Table 5.1: Error bounds for 100 variables

The error bounds for 100 variables look very similar for 150 and 700 inequalities. Due to the small problem size, the logarithmic bounds are almost indistinguishable. But all other methods are seemingly better than the Raghavan-Thompson rounding. So, we look into the error bounds for larger problem size, problems with 1000 variables. In table 5.2 we present the error bounds. The oblivious rounding is clearly producing the worst error bound.

The edge-walk and iterative rounding followed by LLL seems very much similar, although the latter seems to perform slightly better for large number of inequalities. The LLL based rounding as a stand alone method performs better than oblivious rounding, but it tends to suffer due to the increased dependencies among the inequalities when problem size grows.

It is worth noting that even though the dependencies among the inequalities increase with larger values of k, the error seems to decrease after a certain point. At first this looks strange, but a closer inspection reveals that even though the dependencies increase with larger value of k, the objective function $\Omega(n/k)$ decreases. This implies that the number of non-zero values in $\hat{\mathbf{x}}$ also decreases. As the number of non-zero variables per inequality is very small in the final rounded solution, the error is also small.

	Oblivious Rounding	Edge walk based method	Rounding using LLL	Iterative rounding followed by LLL
$k = \log n$	5	3	3	3
$k = \log^{1.5} n$	5	3	4	3
$k = \log^2 n$	4	3	3	3

(a) $m = 1500,\ n = 1000$

	Oblivious Rounding	Edge walk based method	Rounding using LLL	Iterative rounding followed by LLL
$k = \log n$	5	3	3	3
$k = \log^{1.5} n$	5	4	5	3
$k = \log^2 n$	4	3	3	3

(b) $m = 9000,\ n = 1000$

Table 5.2: Error bounds for 1000 variables

5.3 Comparison in terms of objective function

This is the second facet of comparison between the rounding techniques. Due to the limitation of scope, we would not be able to put the results extensively in this thesis. However, we shall show the distinctive results and try to draw a concluding remark based on them.

We filter out all the results where *error* is less than some $\beta,\ \beta > 1$. Then we look for the maximum value of the objective function. The results are shown in following tables in this section.

	Fractional feasible solution $\Omega(n/k)$	Oblivious Rounding	Edge walk based method	Rounding using LLL	Iterative rounding followed by LLL
$k = \log n$	15	14	14	12	12
$k = \log^{1.5} n$	6	7	6	3	5
$k = \log^2 n$	2	2	2	2	2

(a) $m = 150,\ n = 100$

	Fractional feasible solution $\Omega(n/k)$	Oblivious Rounding	Edge walk based method	Rounding using LLL	Iterative rounding followed by LLL
$k = \log n$	15	13	14	16	16
$k = \log^{1.5} n$	6	5	6	5	5
$k = \log^2 n$	2	2	2	3	3

(b) $m = 700,\ n = 100$

Table 5.3: Maximum Objective functions for 100 variables, $error \leq 3$

For problem size of 100 variables, the difference is negligible. The objective function values for larger k are indistinguishable and does not add much to our understanding.

Next we take a look on experiments with 1000 variables. The result is shown in table 5.4. Not all methods converged within error ≤ 4. The parenthesized values in the table indicate the value of error for such cases.

	Fractional feasible solution $\Omega(n/k)$	Oblivious Rounding	Edge walk based method	Rounding using LLL	Iterative rounding followed by LLL
$k = \log n$	100	88 (5)	84	67	67
$k = \log^{1.5} n$	32	28 (5)	26	25	31
$k = \log^2 n$	10	9	8	9	9

(a) $m = 1500,\ n = 1000$

	Fractional feasible solution $\Omega(n/k)$	Oblivious Rounding	Edge walk based method	Rounding using LLL	Iterative rounding followed by LLL
$k = \log n$	100	78 (5)	81	61	61
$k = \log^{1.5} n$	32	26 (5)	24	22 (5)	30
$k = \log^2 n$	10	8	7	7	9

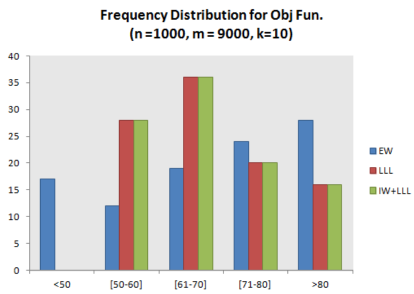
(b) $m = 9000,\ n = 1000$

Table 5.4: Maximum Objective functions for 1000 variables, $error \leq 4$

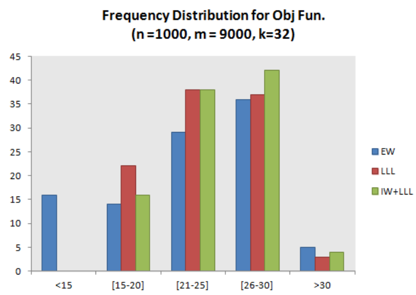
It is evident that the edge-walk based technique outperforms other methods when the number of non-zero variables per inequality is relatively small ($k = \log n$). We have already seen that in terms of error, the methods perform almost similarly. But edge-walk gives cut above performance $\log n$-sparse constraint matrix when we look at the objective function value. Although for larger values of k, all the methods seem to perform at a similar scale.

5.3.1 Frequency Distribution for Objective Functions

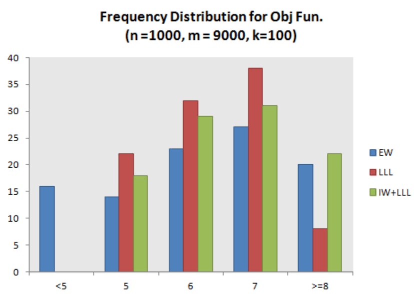
The following figure briefly captures the freuqency distribution of objective function for different type of methods. It gives a broader picture of how the objective functions are grouped for different rounding methods. The groups are very identical and actually show that the methods are not very different from each other, at least up to the problem size of 1000 variables. In this spirit, we now move on to the concluding chapter.



(a) $k = \log n$



(b) $k = \log^{1.5} n$



(c) $k = \log^2 n$

Figure 5.1: Frequency distribution of Objective Functions for $m = 1500, n = 1000$

Chapter 6

Conclusion

There are two main improved rounding techniques which seem to outperform Raghavan Thompson rounding. The algorithm described in Chapter 4 has been analyzed rigorously but for the algorithm in Chapter 3, we are yet to conclude the analysis. Nevertheless, it still provides a good alternative for rounding packing integer problem. The edge-walk seems to do better in case of $k = \log n$ non-zero variables per constraint. The objective function appears to be better than the LLL based method. But the edge-walk is the most expensive method among all since it calculates the orthonormal subspace in every step. The Gram-Schmidt orthogonalization process costs heavily. Following table shows example running time,

	Oblivious Rounding	Edge Walk based Method	LLL	Independent Walk + LLL
k = 10	30 ms	75.82 s	300 ms	4.75 s
k = 32	30 ms	137.2 s	1.17 s	14.42 s
k = 100	30 ms	187.45 s	5.76 s	32.38 s

Table 6.1: Running time for n = 1000, m = 1500

Why the Edge walk based method and LLL based method produce similar kind of error distribution is still an open question. The experiments provided us some useful insights and we hope to come up with rigorous analysis of the edge-walk algorithm in the future.

Bibliography

[1] Nikhil Bansal. Constructive algorithms for discrepancy minimization. *FOCS*, pages 3–10, 2010.

[2] Sanjeev Khanna Julia Chuzhoy, Venkatesan Guruswami and Kunal Talwar. Hardness of routing with congestion in directed graphs. *STOC*, pages 165–178, 2007.

[3] Shachar Lovett and Raghu Meka. Constructive discrepancy minimization by walking on the edges. *FOCS*, pages 61–67, 2012.

[4] Robin A Moser and Gábor Tardos. A constructive proof of the general lovász local lemma. *Journal of the ACM (JACM)*, 57(2):11, 2010.

[5] Erdös P. and Lovász L. Problems and results on 3-chromatic hypergraphs and some related questions. *In A. Hajnal, R. Rado and V.T. Sos, Eds. Infinite and Finite Sets*, pages 609–627, 1975.

[6] Prabhakar Raghavan and Clarke D. Thompson. Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica - Theory of Computing*, 7(4):365–374, 1987.

[7] Thomas Rothvoß. Approximating bin packing within o(log opt * log log opt) bins. *FOCS*, pages 20–29, 2013.

[8] Sandeep Sen. Improved randomized rounding using random walks. *manuscript*, 2014.

[9] Sandeep Sen and Dhiraj Madan. New techniques for randomized rounding. *manuscript*, 2015.