Rounding using Random Walks – An experimental study

Soumen Basu

Advisor: Prof. Sandeep Sen

June 17, 2015

Outline

Problem Overview

We often come across combinatorial optimization problems can be modelled using a weighted (non-negative) packing integer program

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

subject to,

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

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 $S_j \subset \{1, 2 \ldots n\}$.

Similar Problems

- Maximum Independent Set - MISR problem
- Automated Label placement
- Network resource allocation

Classical Approach

Packing ILP is **NP-Hard**

The LP is solved corresponding to the relaxation $0 \leq x_i \leq 1$ and the LP 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, we round each x_i' using the *conventional(oblivious) randomized rounding*[Raghavan, Thompson]

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

Classical Approach

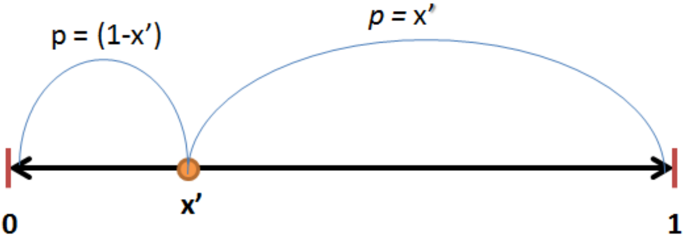


Figure: Pictorial view for Raghavan-Thompson rounding

It can easily be shown that, $E[\sum_i c_i \cdot \overline{x_i}] = OPT = \sum_i c_i \cdot x_i'$ and $\forall j$, with high probability, $\sum_{i \in S_j} \overline{x_i} \leq \frac{\log n}{\log \log n}$.

Need of new approach

- The $\mathcal{O}(\frac{\log n}{\log \log n})$ bound is tight in worst case [Chuzhoy et al.].
- The main objective is to obtain better error bound for integer linear packing problems for *average* problem size.

Need of new approach

- The $\mathcal{O}(\frac{\log n}{\log \log n})$ bound is tight in worst case [Chuzhoy et al.].
- The main objective is to obtain better error bound for integer linear packing problems for *average* problem size.
- $\mathbf{A} \in \mathcal{A}_k^{m \times n}$, family of $m \times n$ 0-1 matrices. Each row has 1's in k randomly chosen columns from $\{1, 2, \ldots n\}$ and 0 elsewhere.

Experimental Setup

$$Max \sum_i x_i$$

subject to,

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

No. of variables, $n = (100 \text{ to } 1000)$;

No. of inequalities $m = O(n)$ to $O(n \log n)$, $m = 150, 700$(for $n = 10^2$) and $m = 1500, 9000$(for $n = 10^3$);

$\sum_j A_{ij} = k = \log n, \log^{1.5} n, \log^2 n$

Iterative Rounding

Instead of using the one-shot conventional rounding, we use an iterative rounding approach based on random walk.

Input: $x'_i \quad 1 \leq i \leq n$ satisfying the LP
Output: $\overline{x_i} \in \{0, 1\}$
Initialize all variables as *un-fixed* and set $X_0 = [x'_1, x'_2 \ldots x'_n] = \bar{\mathbf{x}}$
Repeat

1. Generate a random vector $R^i = U_i$ where U_i is a standard multidimensional Normal distribution.
2. $X_{i+1} = X_i + \gamma \cdot R^i$
3. Fix a variable if it is less than δ or greater than $1 - \delta$
   If all variables are fixed then exit.

untill some stopping condition.
Set the *unfixed* variables to 1 and return the rounded vector X_{final}.

Calculation on Independent Rounding

Martingale Optional Stopping Theorem

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

Calculation on Independent Rounding

Martingale Optional Stopping Theorem

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

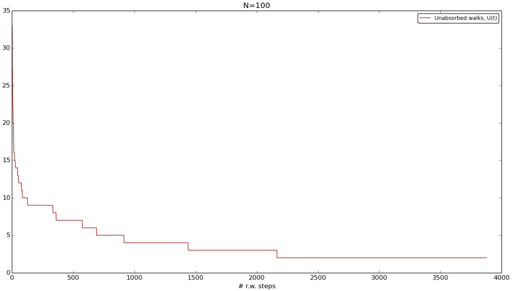
Suppose we start random walks for $\log n$ variables. Each variable was initialized to $\frac{1}{\log n}$. For independent rounding the probability that $\frac{\log n}{\log \log n}$ of these variables hit 1

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

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

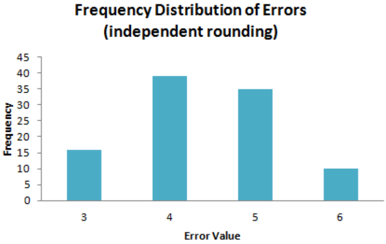
Observation on Convergence time

We define $U(t)$ by the no. of non-convergent(unabsorbed) variables at time t

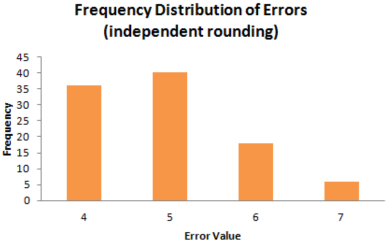


The walk converges in $O(1/\gamma^2)$ steps. $U(\frac{\log n}{\gamma^2}) = 0$.

Frequency distribution of Error



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

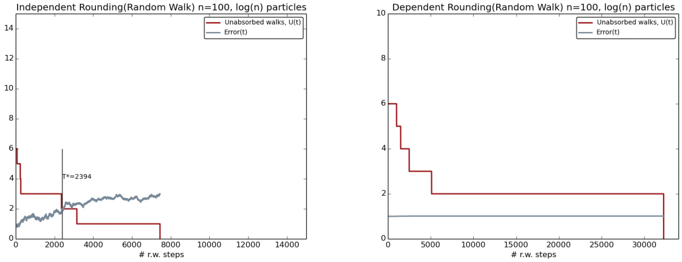


(b) $n = 1000, m = 1500 \;\; O(n)$

Figure: Frequency Distribution of Error

'Dependent' Gaussian walks for single inequality

'Dependent' Gaussian walks involves in generation of the multidimensional Gaussian on the orthonormal basis of the subspace of the inequality and then projecting back the steps in the original coordinate system. This ensures that the inequality is never violated.



(a) Rounding using independent gaussian walk

(b) Rounding using constrained gaussian walk

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

Solving an LP using this approach

For a system of inequalities, we Generate the Gaussian in the orthonormal basis of the Nullspace(A) and then project them on the original coordinates. But this method fails if $m > n$, since $N(A) = \{\phi\}$
But even if $m < n$, the nullspace dimension subsequently becomes higher than the unabsorbed variable dimension and this method fails.

Edge-walk Method

This approach was inspired by the idea of Discrepancy minimization using edge walks [Lovett and Meka].
The algorithm is largely regarded as edge-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.

Edge walk Algorithm

Input: $x'_i \quad 1 \leq i \leq n$ satisfying the LP
Initialize: $X_{0,0} = [x'_1, x'_2 \ldots x'_n]$, where $X_{i,j}$ denotes the position of the walk in the i^{th} iteration and jth phase
For each phase $j = 0, 1, 2, \ldots$

Start independent random walk till any constraint C_k is 'nearly hit' (δ-close). Set $S_0 = C_k$

For iteration $i = 0, 1, 2, \ldots$

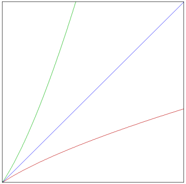
1. Generate a random vector U_i within current subspace S_i.
2. $X_{i+1,j} = X_{i,j} + \gamma \cdot U_i$
3. If $X_{i+1,j}$ is δ-close to constraint C_k^i, then the subspace of the walk $S_{i+1}$is restricted to $C_k^i \cap S_i$. Also, fix a variable if it is less than δ or greater than $1 + \Delta_j - \delta$. If all variables are fixed then exit.

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 $j + 1$.
until *not all variables are fixed.*

Choosing Δ_i - polytope expansion

When we add Δ_i to the right hand side of inequalities, the polytope effiectively expands. There are three basic ways to choose Δ_i

1. **Convex expansion:** $\Delta_p = c \cdot O(p^{\alpha})$ suffices. p is the current phase and $0 < c < 1$, $\alpha > 1$ are constants.



2. **Linear expansion:** $\Delta_p = \Delta_{p-1} + c$
3. **Concave expansion:** $\Delta_p = c \cdot p^{\beta}$, where $0 < c, \beta < 1$. Problem: $|d\Delta| < \gamma, \delta$.

Experimental Observations - Convergence time

The follwoing figure shows worst case convergence time in terms of random walk steps for $n = 100,\ 1000$ variables. For both cases $m = O(n)(1500)$.

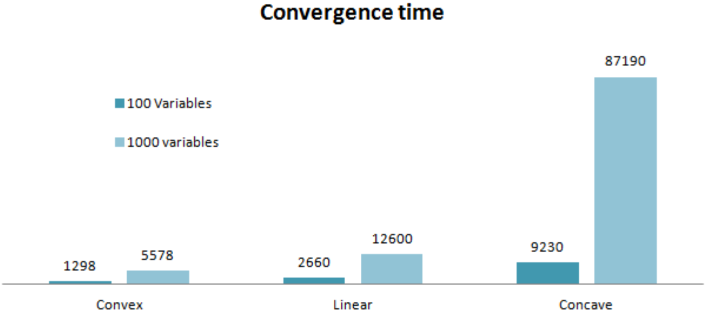
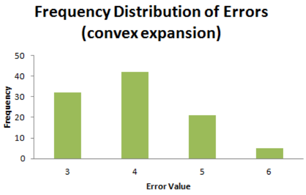
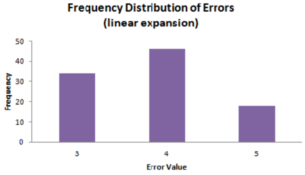


Figure: Convergence time for different type of polytope expansion

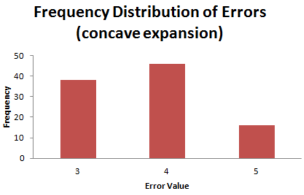
Frequency distribution of Error



(a) Convex expansion 1000 vars



(b) Linear expansion 1000 vars



(c) Concave expansion 1000 vars

Lovász Local Lemma

The Lovász Local Lemma provides a strategy for proving the existence (or non-existence) of an object that satisfies some prescribed property.

Theorem (Lovász Local Lemma)

Let A_i, $1 \leq i \leq N$ be events where the dependencies between the events 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

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

Applying LLL based rounding

- $A_{m\times n}$ has no more than ρ 1s in any column.
- The dependency of any row can be bounded by $\rho \log n$.
- $\Pr[\text{a row exceeds } t > 1] \leq \frac{1}{2^t}$ from Chernoff bounds.
- E_i: event that row i exceeds t.
- Define an additional event E_{m+1}: objective function value less than $(1-\alpha)\cdot OPT$ for $1 > \alpha > 0$.

$$\bigcap_{1\leq i\leq m+1} \bar{E}_i$$

LLL based Iterative Rounding

Input: $x'_i \quad 1 \leq i \leq n$ satisfying the LP and t (error parameter)
Output: $\hat{x}_i \in \{0, 1\}$
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* $< OPT/2$

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.

Return rounded vector $\hat{\mathbf{x}}$.

LLL - example

V_1: 1 0 1 1 0 0 0

V_2: 1 1 0 0 1 0 0

V_3: 0 0 1 1 0 1 0

LLL - example

V_1: 1 0 1 1 0 0 0

V_2: 1 1 0 0 1 0 0

V_3: 0 0 1 1 0 1 0

Suppose $X_0 = [1\ 0\ 0\ 1\ 1\ 0\ 0]$ $\qquad C = [2\ \ 2\ \ 1]$

LLL - example

V_1: 1 0 1 1 0 0 0

V_2: 1 1 0 0 1 0 0

V_3: 0 0 1 1 0 1 0

Suppose $X_0 = [1\ 0\ 0\ 1\ 1\ 0\ 0]$ $\quad C = [2\ \ 2\ \ 1]$

Lets say V_1 was picked. x_1, x_3, x_4 needs to be rounded again.

LLL - example

V_1: 1 0 1 1 0 0 0

V_2: 1 1 0 0 1 0 0

V_3: 0 0 1 1 0 1 0

Suppose $X_0 = [1\ 0\ 0\ 1\ 1\ 0\ 0]$ $\qquad C = [2\ \ 2\ \ 1]$

Lets say V_1 was picked. x_1, x_3, x_4 needs to be rounded again.

Now suppose $X_1 = [0\ 0\ 1\ 0\ 1\ 0\ 0]$ $\qquad C = [1\ \ 1\ \ 1]$ $\qquad$ stop

Comparison of all methods

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.

In terms of Error

	RT	EW	LLL	IW+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

Table: 100 vars, 150 inequalities

	RT	EW	LLL	IW+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

Table: 100 vars, 700 inequalities

In terms of Error

	RT	EW	LLL	IW+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

Table: 1000 vars, 1500 inequalities

	RT	EW	LLL	IW+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

Table: 1000 vars, 9000 inequalities

In terms of Objective Function

	$\Omega(n/k)$	RT	EW	LLL	IW+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

Table: 100 vars, 150 inequalities

	$\Omega(n/k)$	RT	EW	LLL	IW+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

Table: 100 vars, 700 inequalities

Error ≤ 3

In terms of Objective Function

	$\Omega(n/k)$	RT	EW	LLL	IW+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

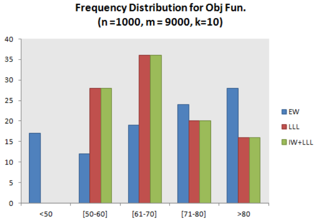
Table: 1000 vars, 1500 inequalities

	$\Omega(n/k)$	RT	EW	LLL	IW+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

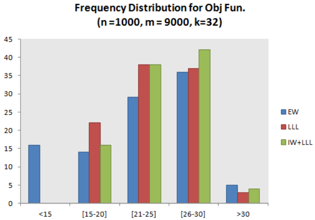
Table: 1000 vars, 9000 inequalities

Error ≤ 4

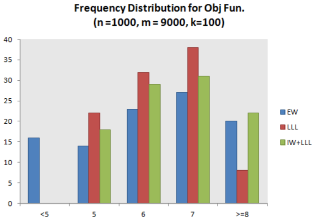
Frequency distribution of Objective Functions



(a) $k = \log n$



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



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

Actual Running Time

RT	EW	LLL	IW+LLL
30 ms	75 s	300 ms	4.75 s

Table: n = 1000, m = 1500, k= 10

RT	EW	LLL	IW+LLL
30 ms	137 s	1.17 s	14 s

Table: n = 1000, m = 1500, k= 32

RT	EW	LLL	IW+LLL
30 ms	187 s	5.76 s	32.38 s

Table: n = 1000, m = 1500, k= 100

EW takes 75 s, 4.76 m and 1.17 h for convex, linear and concave expansion.

Conclusion

There are two main improved rounding techniques which seem to outperform Raghavan Thompson rounding. 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. Why the two methods produce similar kind of error distribution is still an open question. We hope to come up with rigorous analysis of the edge-walk algorithm in the future.

References

1. Prabhakar Raghvan and Clark D. Thompson. Randomized rounding: a technique for provably good algorithms and algorithmic proofs.
2. Shachar Lovett, Raghu Meka; Constructive Discrepancy Minimization by walking on the edges.
3. Nikhil Bansal. Constructive Algorithms for Discrepancy Minimization.
4. Sanjeev Khanna Julia Chuzhoy, Venkatesan Guruswami and Kunal Talwar. Hardness of routing with congestion in directed graphs.
5. Robin A Moser and Gábor Tardos. A constructive proof of the general lovász local lemma.
6. Sandeep Sen. Improved randomized rounding using random walks.