Refinements of Randomized Rounding

Nikhil Bansal
(IBM $\rightarrow$ TU Eindhoven)

Linear/Convex Programming

Heart of combinatorial optimization

Discrete Problem (E.g. Travelling Salesman, …)

Solve continuous relaxation $x_i \in \cancel{\{0,1\}}\ [0,1]$

(LP: min cx s.t. Ax <= b)

"Round" fractional solution

Relax and Round

By far the most powerful and ubiquitous technique.

[Raghavendra 08 and others]: For several problems LPs/SDPs best among any polynomial time algorithm (assuming P $\neq$ NP)

Our focus on Rounding:

Polyhedral Combinatorics
Geometry
Probability
…

Two basic approaches

Careful +/- Rounding

Transform LP solution to integral solution by careful updates (guided by problem structure)

60 – 80s Exact Algorithms (Work of Edmonds)

Combinatorial optimization.

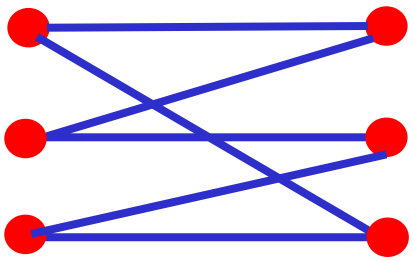
Various extensions in Approximation Algorithms.

Bipartite Matching

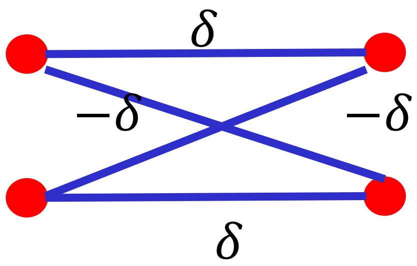
Given: $n \times n$ graph with edge costs

Min $c_e x_e$

$\sum_{e \in \delta(v)} x_e = 1$ (integer b_v in general)



Rounding: Fix Fractionality



Either $\delta > 0$ or < 0 good wrt cost.

$LP = 0.5\ LP(\delta) + 0.5\ LP(-\delta)$

Thm: Any LP = convex combination of matchings

Randomization

View $x_i \in [0,1]$ as probabilities

Randomized Rounding:

Independently round each x_i to 1 with probability x_i.

Suppose $\sum_i a_{ji} x_j = b_i$

Then upon rounding, $\mathrm{E}[\,\sum_i a_{ji}\,\tilde{x}_j\,] = b_i$

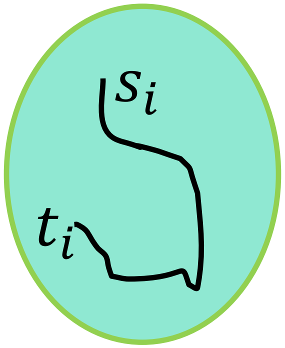
Sharp concentration (Chernoff): $\sum_i a_{ji}\,\tilde{x}_j = b_i$ + "small error"

Can handle several constraints + possibly several objectives

Randomization

Raghavan Thompson 84 (Routing with congestion)

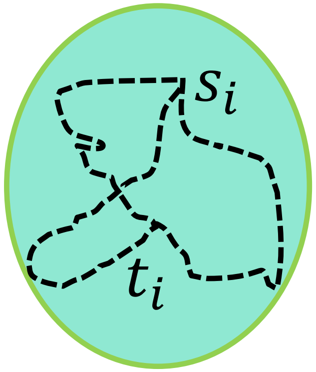
Goal: Given $s_i - t_i$ pairs. Find routes with congestion 1.



LP: Find unit flow from $s_i - t_i$ (s.t. edge capacity = 1)

Flow = prob. distribution over paths.

Pick one path at random for each flow.



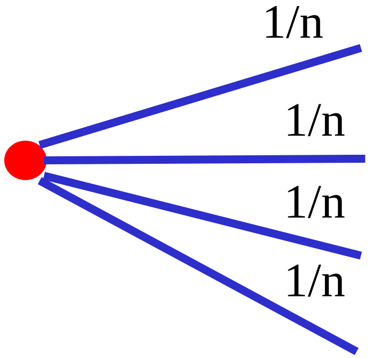
Expected congestion of any edge ≤ 1

Max. congestion = O(log n/log log n)

Often at odds

Matching: $\sum_{e \in \delta(v)} x_e = 1$

Picking x_e at random



Pr [vertex unmatched] = $\left(1 - \frac{1}{n}\right)^n \approx 1/e$

Randomization is not the right thing here.

Motivating Example

Several cost functions $c^1_{\{e\}},\ c^2_{\{e\}},\ \ldots, c^k_{\{e\}}$

Cost (M) $= \max_i\ c^i(M)$

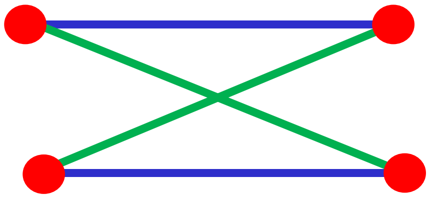
Suppose good fractional matching x

i.e. $C^i x \ \le C^*$ for all i =1 … k.

Two cost functions (blue and green)

M1: (2,0)   M2: (0,2)   So, Value = 2

Fractional Matching (0.5 M1 + 0.5 M2): (1,1)



Matching: Multiple Costs

Ideally: Pick each edge randomly.

Max cost (Chernoff) ≈ $C^* + O\left(\sqrt{\log\ k}\right) \cdot (std.\text{deviation})$

Naïve approaches:

1. Sample from Convex Combination (LP = Prob. over matchings)

LP = 1/k $(M_1 + M_2 + \cdots M_k)$ M_i : $(0, \ldots, 0, k\, C^*, 0, \ldots)$

Each individual M_i very bad (for some color)!

2. Pick each edge randomly

Good wrt all costs.

Must sample O(log n) times to ensure each vertex degree >= 1.

Combined Approaches

Very useful techniques developed in recent years.

For matching with several costs.

Can obtain Chernoff type bound [Arora Frieze Kaplan 96]

Outline

Introduction

Bipartite Graphs:

- Gandhi et al. Rounding
- Arora et al. Rounding

Spanning Trees (Matroids):

- Chekuri et al. Rounding

General Rounding (based on discrepancy)

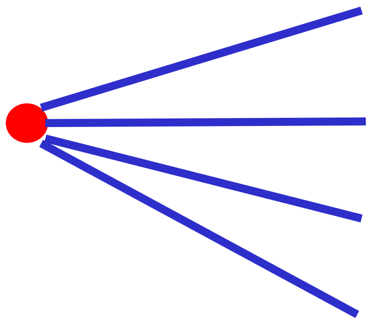
Dependent Rounding

[Srinivasan 01; Gandhi-Khuller-Parathasarathy-Srinivasan 04]:

Relevant for b-matchings (b>1)

Give a randomized algorithm s.t.

For any node



(i) Pick exactly b edges

(ii) Prob. Edge e picked = x_e

(iii) Cost of edges at each vertex concentrated around LP mean.

Extremely useful: E.g. in load balancing

Bipartite Rounding

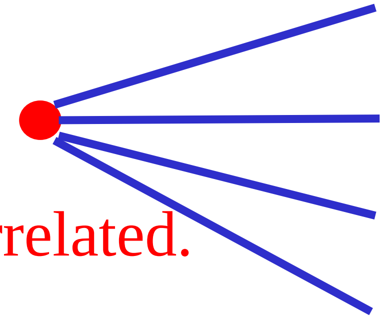
Alg: Take an even cycle, randomly round up or down.

Thm: (i) Pr [e chosen] = x_e

(ii) Degree b_v exactly preserved

(iii) Edges on a vertex are Negatively Correlated.

$$\Pr[\, \Pi_{e \in S} (X_e = 1)] \leq \Pi_{e \in S} \Pr[\, X_e = 1 \,]$$



Useful Fact: Negative Correlation implies Chernoff bounds.

Dependent Rounding

Proof of thm:

Inductively $E[\,\Pi_{e \in S} X_e]$ decreases over iterations.

At beginning t=0, x_e = LP value.

At end it is the 0-1 random variable.

In S, either update none, or 1 or 2 elements.

$X_1 X_2 \ldots X_k$

$\rightarrow$ ½ $(X_1 - \epsilon)(X_2 + \epsilon)\ X_3 \ldots\ X_k$ + ½$(X_1 + \epsilon)(X_2 - \epsilon) X_3 \ldots X_k$

Second Rounding

[Arora-Frieze-Kaplan 96]: Implies Chernoff like bounds

Guarantee: $(1+\epsilon)C^* + \tilde{O}(\sqrt{\mathrm{n}} \cdot c\_max)$ c_max: max edge cost

Alg: Consider LP $= \sum_i \ \alpha_i M_i$

Pick large k, view as $k\alpha_i$ matchings M_i (chosen to extent 1/k)

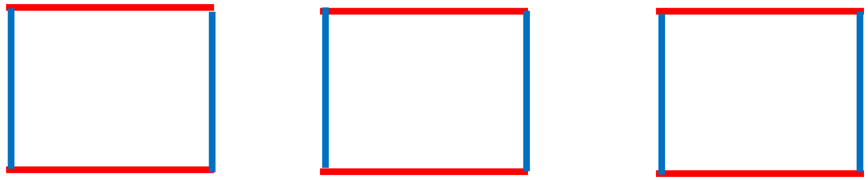
Will reduce to k/2 matchings (extent 2/k) and so on

until get one Matching.

Algorithm

Given matchings $M_1, \ldots M_k$, pair them arbitrarily.

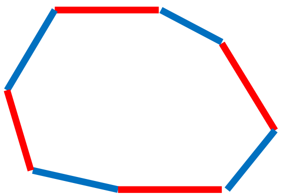
Consider $M_1 \cup M_2$. Union of alternating even cycles.



For each cycle,
pick one color randomly

If insufficient randomness (e.g. union = single cycle)

Break long cycles (make of length <= $\sqrt{n}$)

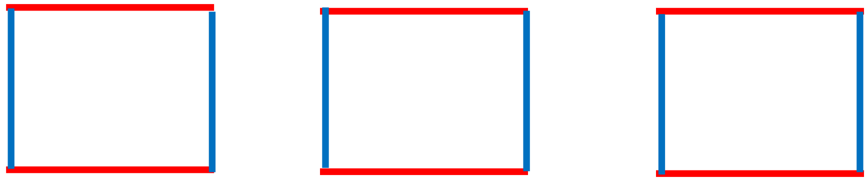


Proof: Whp $c(\text{Merge}) \approx$ ½ $\left(c(M_1) + c(M_2)\right)$ + tiny error

Algorithm

Given matchings $M_1, \ldots M_k$, pair them arbitrarily.

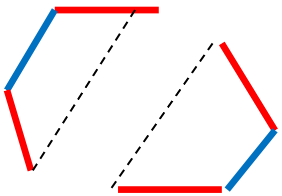
Consider $M_1 \cup M_2$. Union of alternating even cycles.



For each cycle,
pick one color randomly

If insufficient randomness (e.g. union = single cycle)

Break long cycles (make of length <= $\sqrt{n}$)



Proof: Whp $c(\text{Merge}) \approx$ ½ $\left(c(M_1) + c(M_2)\right)$ + tiny error

Outline

Introduction

Bipartite Graphs:

- Gandhi et al. Rounding
- Arora et al. Rounding

Spanning Trees (Matroids):

- Chekuri et al. Rounding

General Rounding (based on discrepancy)

Spanning Trees

Polyhedral Description

Min $\sum_e \; c_e x_e$

$\sum_e \; x_e \;=\; n-1$ (n-1 edges)

$\sum_{e\in E(S)} \; x_e \leq \; |S|-1$ (for each subset S of vertices)

Edmonds 65:

Integral (LP = convex combination of spanning trees)

Sometimes prefer convex combination with stronger properties.

General Degree Bounds

Arbitrary subset bounds: For subset S of edges, bound b_S.

LP: Spanning Tree constraints

+ Degree constraints

Convex combination is useless (the individual trees could have large maximum degrees)

General Degree Bounds

Thm: $(1+\epsilon)b_S \ + \ O_\epsilon(\log k)$ [exactly Chernoff bound]

k: number of degree bounds

Proof: Will sample spanning tree with marginals x_e,

s.t. edges are negatively correlated (can use Chernoff bounds)

1. Max. Entropy Sampling [Asadpour-Goemans-Madry-OveisGharan-Saberi 10]. Gave O(log n/log log n) for ATSP.

2. Randomized Matroid Rounding [Chekuri-Vondrak-Zenklusen 10]

CVZ Matroid Rounding

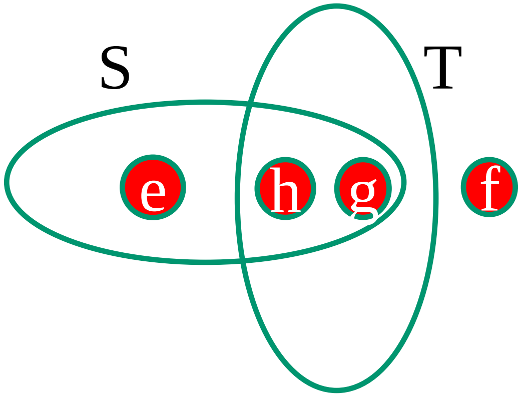
$\sum_{e \in E[S]} x_e \leq |S| - 1$

Apply +/- to candidate x_e, x_f

Trouble: Some tight set contains e but not f.

Let g be other fractional element in S.

May be some other set T complains.



Look at $S \cap T$.

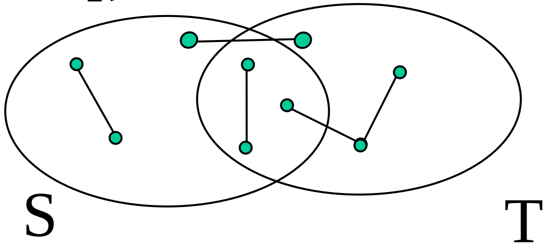
Claim (uncrossing): If S and T tight, $S \cap T$ also tight (so integral)

Repeat, until find minimal tight set.

Proof of Claim

Claim: If S,T tight, then $S \cap T$ and $S \cup T$ also tight

$$x(E[S]) + x(E[T]) \leq x(E[S \cap T]) + x(E[S \cup T])$$



LHS = |S|-1 + |T|-1

RHS $\leq$ $|S \cap T| - 1 + |S \cup T| - 1$

= |S| + |T| - 2

More Applications (TSP)

Christofedes (70's): 3/2 approximation

TSP $\leq$ Spanning Eulerian Subgraph

$\leq$ Min. Spanning Tree + Matching on odd degrees.

(at most OPT TSP) (at most ½ OPT TSP)

Thm [Oveis,Saberi,Singh 11]: 1.5 - ϵ for "graphic" TSP

Alg: Pick Random spanning Tree.

Directed Graphs: Reduce to finding "thin spanning tree"

Thm [Asadpour,Goemans,Madry,Oveis,Saberi 10]: O(log n/log log n)

Outline

Introduction

Bipartite Graphs:

Gandhi et al. Rounding

Arora et al. Rounding

Spanning Trees (Matroids):

Chekuri et al. Rounding

General Rounding (based on discrepancy)

A General Question

Given Ax = b say with fractional x, say x $\in$[0,1].

Want to round x to integer $\tilde{x}$ s.t. error $|Ax - A\,\tilde{x}|_\infty$ is as small as possible.

$$\begin{array}{c} x_1 x_2 \ldots x_n \\ \begin{bmatrix} & & \\ & A & \\ & & \end{bmatrix} \end{array} = b$$

Natural Idea: Do +/- updates on columns, s.t. low error added

Discrepancy

Hereditary Discrepancy: λ

If for any subset S of columns, there is some (clever) +/- coloring of S, s.t. each row sum <= λ

Thm [Lovasz Spencer Vestzergombi 86]:

There always exists rounding with error < $2 \cdot$ Herdisc(A)

Structure often implies low herdisc.

(TU matrices = 1, geometric set systems, …) [Discrepancy Theory]

Existential result: How do we efficiently find the +/- update?

(in matchings: even cycles, in spanning trees: two elements in a minimal set)

Making the result Algorithmic

Thm [Bansal 10]: Given any x, can round in polynomial time with error $\mathrm{Herdisc}(A) \cdot O(\log n)$

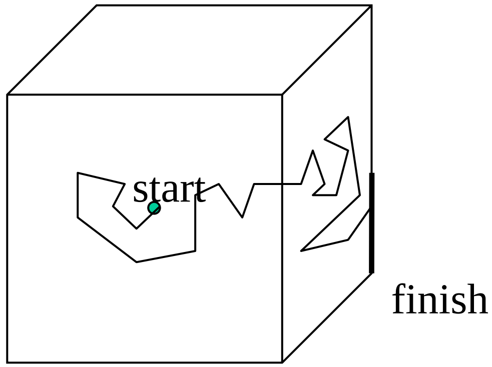
Nothing previously known.

And, instead of +/- updates, we will do Gaussian Updates.

We will use SDPs to find these updates.

Algorithm (at high level)

Cube: $\{0,1\}^n$



Each dimension: A variable
Each vertex: A rounding

Algorithm: At step t, update $x_i(t) = x_i(t-1) + \delta_i(t)$
Fix variable if reaches 0 or 1.

Each $\delta_i(t)$ distributed as a Gaussian
But the δ_i's are correlated (e.g. $\delta_1(t) + \delta_2(t) + \delta_3(t) = 0$)
so that low error added to rows.

SDP relaxations

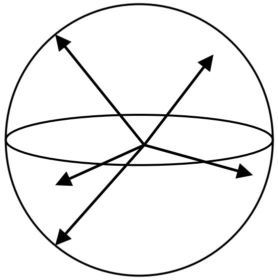
SDPs (LP on $v_i \cdot v_j$)

$$| \sum_i a_{ji} v_i |^2 \leq \lambda^2 \quad \forall j$$

$|v_i|^2 = 1$ Intended soln. $v_i =$ (+1,0,…,0) or (-1,0,…,0).

Low hereditary discrepancy guarantees feasibility.

But get vectors v_i.



Pick random Gaussian g: Set $\delta_i = g \cdot v_i$

Why Projection?

Say if $v_1 + v_2 + v_3 = 0$

Then $\delta_1 + \delta_2 + \delta_3 = g \cdot (v_1 + v_2 + v_3) = 0$

Properties of Rounding

Lemma: If $g \in R^n$ is random Gaussian. For any $v \in R^n$,

$g \cdot v$ is distributed as $N(0, |v|^2)$

Pf: $N(0,a^2) + N(0,b^2) = N(0,a^2+b^2)$ $\quad g \cdot v = \sum_i v(i)\, g_i \sim N(0, \sum_i v(i)^2)$

If $\delta_i = g \cdot v_i$

SDP:

$|v_i|^2 = 1$

$|\sum_i a_{ji} v_i|^2 \leq \lambda^2$

1. Each $\delta_i \sim N(0,1)$
2. For each row j,

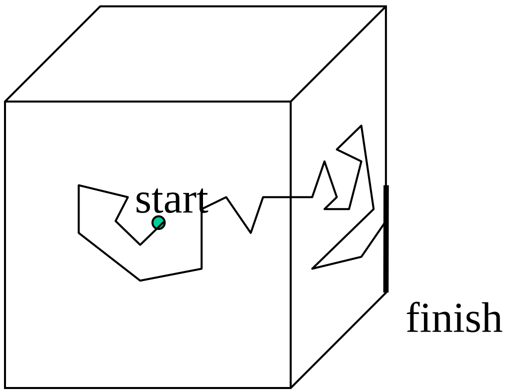
   $\sum_i a_{ji} \delta_i = g \cdot (\sum_i a_{ji} v_i) \sim N(0, \leq \lambda^2)$

   (std deviation $\leq \lambda$)

δ's mimic a = +/- update with low error

Analysis (at high level)

Cube: $\{0,1\}^n$



Each dimension: An Element
Each vertex: A Coloring

Algorithm: Solve SDP. Take a small step. Repeat.

Analysis:

Progress: Few steps to reach a vertex (walk has high variance)

Low Discrepancy: For each equation, discrepancy random walk has low variance

Analysis

Consider time $T = O(1/\gamma^2)$ γ = Scaling factor

Claim 1: With prob. ½, at least n/2 elements reach -1 or +1.

Pf: Each element doing random walk with size $\approx \gamma$.

Recall: Random walk with step ± 1, is $\approx O(t^{1/2})$ away in t steps.

Claim 2: Each row has $O(\lambda)$ discrepancy in expectation.

Pf: For each row j, $x_t(R_j)$ doing $\pm$ random walk with step size $\approx \gamma\,\lambda$

T log n steps suffices to color everything,

Chernoff: Max Error = $O(\lambda \log n)$

Conclusions

Classical Rounding + Randomization

Has turned out to be very fruitful idea.

Matchings, Spanning Trees, Discrepancy, …

Probably several new results still to be discovered.

Thank You!