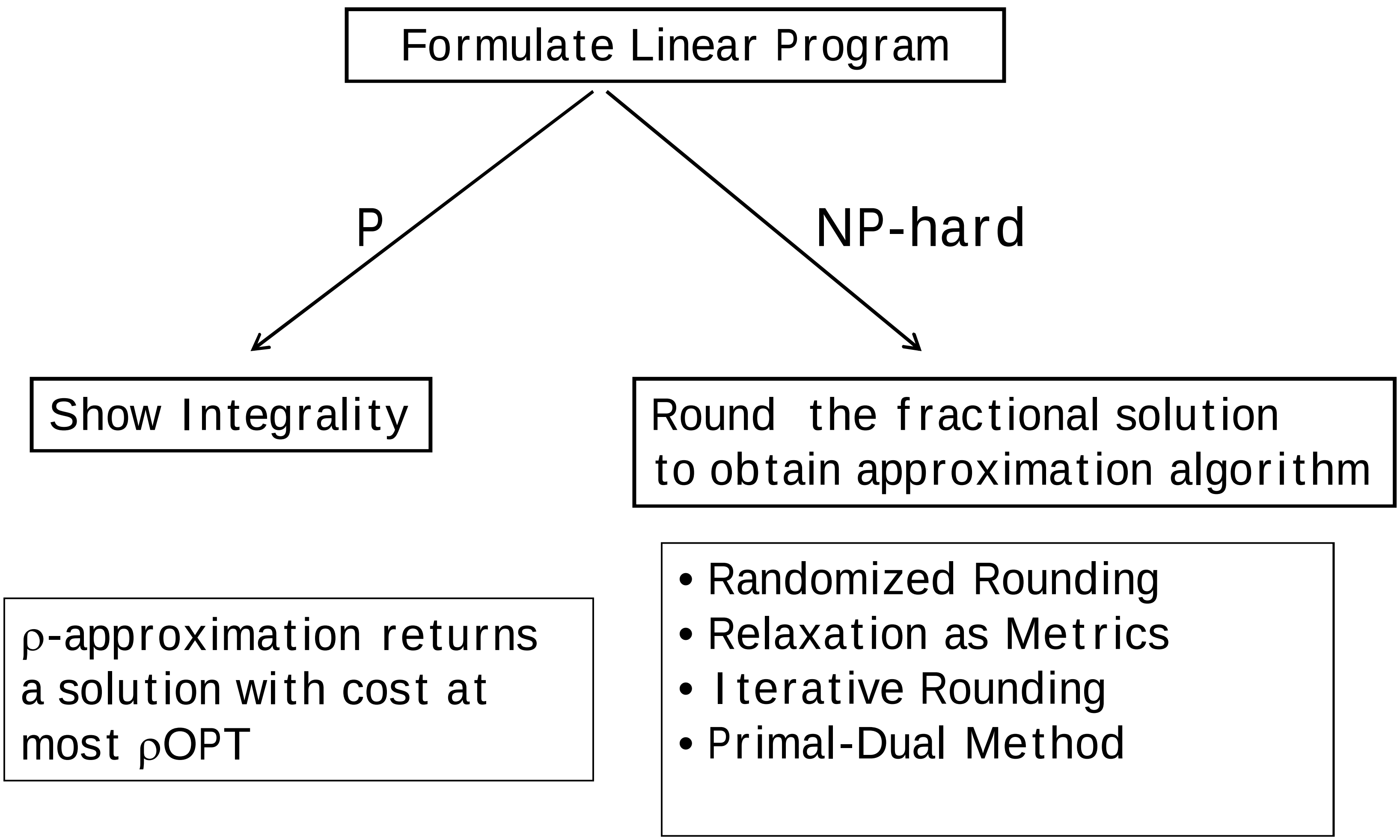
Iterative Methods in Combinatorial Optimization

R. Ravi
Carnegie Bosch Professor
Tepper School of Business
Carnegie Mellon University
ravi@cmu.edu

Combinatorial Optimization

- "Easy" Problems : polynomial time solvable (P)
  - Matchings
  - Spanning Trees
  - Matroid Basis
- "Hard" Problems : NP-hard
  - Survivable Network Design
  - Facility Location
  - Scheduling Problems

Linear Programming



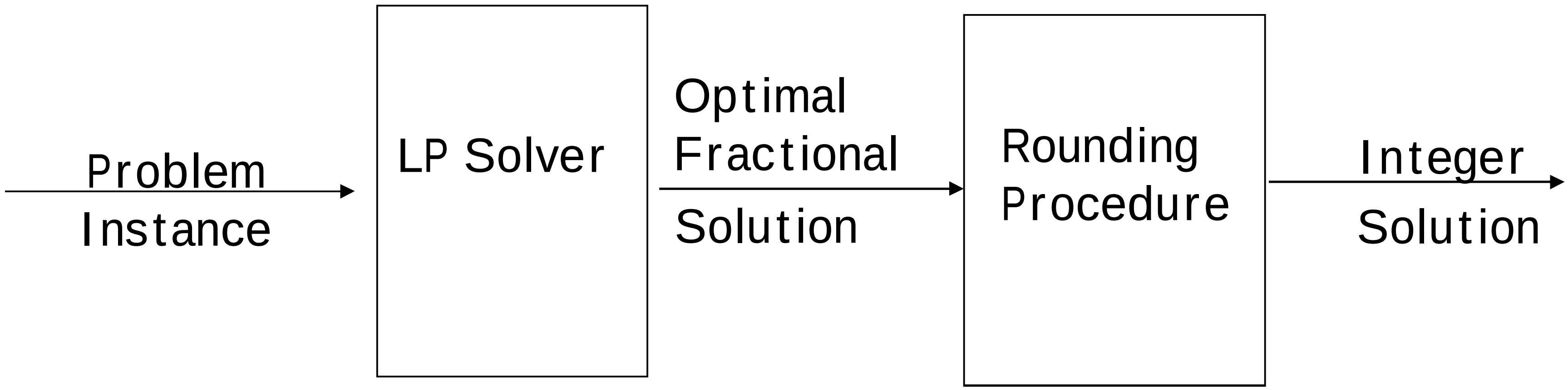
LP-based Methods for Approximation Algorithms

- Rounding large fractional variables
  - (Vertex Cover: Nemhauser-Trotter ‘75, BarYehuda-Even ‘81)
- Randomized Rounding
  - (Packing and Covering Programs: Raghavan-Thompson ‘88)
- Metric Embedding & Rounding
  - (Max Cut – Goemans-Williamson ’94; Multiway Cut: Calinescu-Karloff-Rabani ’98)
- Primal-Dual Methods
  - (Network Design: AKR ‘91, GW ‘92)

LP-based Methods for Approximation Algorithms

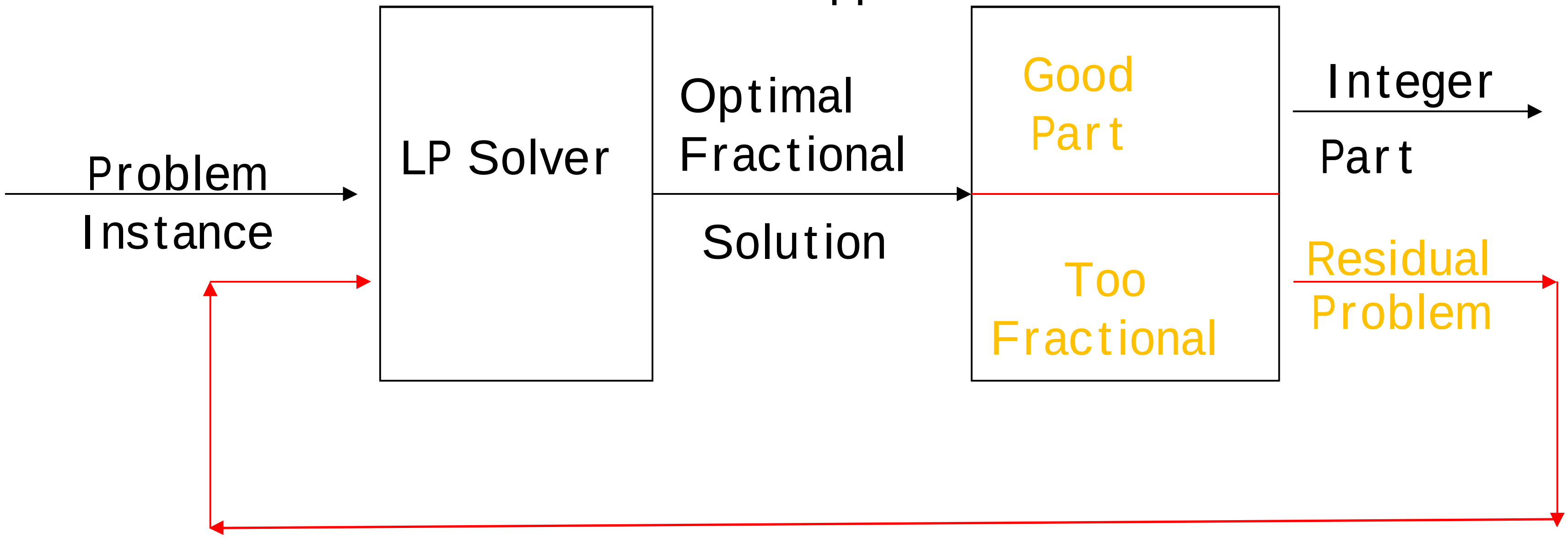
- Iterative Rounding
  - (Survivable Network Design Problem: Jain '98)
- Iterative Relaxation
  - (Degree bounded SNDP : Lau-Naor-Salvatipour-Singh '07, Degree bounded Spanning Trees: Singh-Lau '07)

Typical Rounding:



Iterative Rounding [Jain]:

e.g., Rounding ½-element gives 2-approximation



Iterative Rounding

1. Formulate NP-hard problem as covering LP
2. Argue existence of large-valued element in extreme point solution
3. Round up large-valued element to include in solution
4. Modify constraints to reflect residual problem
5. Repeat until no more constraints remain

Iterative Relaxation

1. Formulate base problem as LP with integral extreme points
2. Design a proof using an iterative method
3. Consider base problem with extra constraints
4. Iterative Relaxation Solution Framework
   - Follow iterative proof of integrality of base pblm
   - Add a step of relaxing (deleting) constraints that have “low” violation
   - Argue existence of either an integral element to include or a constraint to relax at each step
   - When all constraints are relaxed, remaining soln is integral with low violation

Iterative Method: Key Ingredients

1. Small number of independent tight constraints at extreme point solution implies large valued element
2. Bound number of independent tight constraints at extreme point

3. Incorporate side constraints into the argument for 1 with appropriate relaxation

Application

- Addition to toolkit of LP-based design of approximation algorithm
- New proofs of classical integrality results of easy problems
- Allows adaptation to designing approximation algorithm for NP-hard variant with side constraints

Easy Problems to Hard Problems

Base Problem	**Base problem with more constraints**
Spanning Tree	Bounded-degree spanning trees [Singh-Lau] Multi-criteria spanning trees [GRSZ11]
Matroids	Constrained Matroids [LKS08]
Submodular Flow	Constrained Submodular Flow [LKS08]
Bipartite Matchings	Scheduling in Unrelated Parallel Machines [Shmoys-Tardos]
SNDP	SNDP with degree constraints [LNSS]

Outline

- Preliminaries
- Iterative Relaxation (Global Argument)
  - Assignment
  - Generalized Assignment
- Iterative Relaxation (Local Argument)
  - Minimum Spanning Tree
  - Degree-bounded Min-cost Spanning Tree
- [Alternate proof of SNDP rounding]
- Extensions, Open problems

Preliminaries

Matrix of real numbers A

Row rank = dim (span (row vectors))

Column rank = dim (span (column vectors))

Key elementary fact:

Row rank = Column rank (= rank(A))

Column rank $\geq$ Row rank

Consider m X n matrix A

Take a basis for span(rows)

$x_1, x_2, \ldots, x_r$ (in R^n)

Note each Ax_i is in span(columns)

$Ax_1, Ax_2 \ldots Ax_r$ are linearly independent

For otherwise $(c_1 Ax_1 + c_2 Ax_2 \ldots c_r Ax_r) = 0$ implying $(c_1 x_1 + c_2 x_2 \ldots c_r x_r) = 0$

LP

Linear Program

$$\begin{aligned} \text{Min } & c^T x \\ \text{s.t. } & A x \geq b \qquad \text{(P)} \\ & x \geq 0 \end{aligned}$$

Defn: x is an extreme point solution to (P) if there is no nonzero vector y s.t. both x+y and x-y belong to (P)

Alt: x cannot be written as a y + (1-a) z for y and z both in (P)

Extreme point optimal solutions

If min $c^T x$: $Ax \geq b$, $x \geq 0$ has an optimum that is finite, there is an extreme-point solution achieving this value

Extreme point solutions

Let $P = \{x: Ax \geq b, x \geq 0\}$, and for x in P, $A^=$ be the rows that are tight at x, and $A^=_x$ be the submatrix of $A^=$ consisting of columns corresponding to nonzeros in x.

x is an extreme point iff $A^=_x$ has linearly independent columns (full column rank)

- If x is not an ext pt, for some y

$A^{=}(x - y) \geq b$ and $A^{=}(x + y) \geq b$

while $A^{=} x = b$, so $A^{=} y = 0$

- If $A^{=}$ has linearly dependent columns, start with $A^{=} y = 0$, extend y to all columns by adding 0's to show that $(x + \varepsilon y)$ and $(x - \varepsilon y)$ are both feasible

Rank Lemma

Let P = {x': Ax' $\geq$ b, x' $\geq$ 0}, and x be an extreme point solution in P with all positive entries. Then any maximal number of linearly independent constraints that are tight at x (rows obeying A_i x = b_i) equal the number of variables in x

Proof: $A^{=}_{x} = A^{=}$ since all entries positive

$A^{=}$ has full column rank = no. of vars

Row rank of $A^{=}$ = Its column rank

Outline

- Preliminaries
- Iterative Relaxation (Global Argument)
  - Assignment
  - Generalized Assignment
- Iterative Relaxation (Local Argument)
  - Minimum Spanning Tree
  - Degree-bounded Min-cost Spanning Tree
- [Alternate proof of SNDP rounding]
- Extensions, Open problems

Example: Assignment

- Complete bipartite graph $(A, B, A \times B)$ with $|A| = |B| = n$

$$\text{Min} \sum_{ij} c_{ij} x_{ij}$$

$$\sum_{i \in A} x_{ij} \geq 1 \quad \forall j \in B$$

$$\sum_{j \in B} x_{ij} \geq 1 \quad \forall i \in A$$

$$x_{ij} \geq 0 \quad \forall ij$$

Theorem: Extreme points x^* of the above relaxation are integral

Iterative Proof of Integrality

- Claim: At any extreme point x^*, there exists edge ij: $x^*_{ij} = 1$ *(1-edge)*

- To prove theorem, apply lemma repeatedly by deleting matched edge and its endpoints (including it in solution) and re-solving

- Note: we can also remove any edge with $x^*_{ij} = 0$ *(0-edge)* in the graph when re-solving

Proof Approach

- Support graph: graph of edges with nonzero value at extreme point x^*
- Suppose for contradiction there is no 1-edge
- (LB) Lower bound number of edges in support using property of no 1-edges
- (UB) Upper bound number of independent constraints tight at extreme point x^*
- Show LB > UB
- But at extreme point x^*, # support edges = # tight constraints (since column rank = row rank of nonsingular matrix defining it) !

Global Counting Argument

- (LB) Since every node in A has x^*-degree 1 and there are no 1-edges, there are at least 2n edges in support
- (UB) The following system is dependent

$$\sum_{i \in A} x_{ij} = 1 \quad \forall\ j \in B$$

$$\sum_{j \in B} x_{ij} = 1 \quad \forall\ i \in A$$

Hence maximum number of independent constraints tight at x^* is at most 2n -1

Outline

- Preliminaries
- Iterative Relaxation (Global Argument)
  - Assignment
  - Generalized Assignment
- Iterative Relaxation (Local Argument)
  - Minimum Spanning Tree
  - Degree-bounded Min-cost Spanning Tree
- [Alternate proof of SNDP rounding]
- Extensions, Open problems

Extension: Generalized Assignment

- Bipartite graph (Jobs J, Machines M) plus processing times p_{ij} and costs c_{ij} of job j on machine I
- Find min cost schedule with makespan at most P

$$\text{Min} \sum_{ij} c_{ij} \; x_{ij}$$

$$\sum_{i \in M} x_{ij} \geq 1 \quad \forall \; j \in J$$

$$\sum_{j \in J} p_{ij} \, x_{ij} \leq P \quad \forall \; i \in M$$

$$x_{ij} \geq 0 \; \forall \; ij$$

Preparation

- Prune edges with $p_{ij} > P$ (They can never be used in an integral feasible solution)

- If optimal solution x' has 1-edges, the problem can be reduced (delete the job, decrease makespan constraint rhs for the machine; include this assignment in solution)
- If x' has 0-edges, they can be removed

Relaxation

- If there is a machine with degree 1 in support, remove its makespan constraint
  - Single job using it fractionally cannot have $p_{ij} > P$, so final makespan is at most P
- If machine with degree 2 in support, remove its makespan constraint
  - Each of the two jobs potentially fully assigned to it cannot have $p_{ij} > P$, so its final makespan is at most 2P

Iterative Proof

- Claim: For any extreme point x', either there is a makespan constraint to relax, or there is a 1-edge
- Apply lemma repeatedly by either relaxing and re-solving or deleting matched edge and its endpoints (as well as removing 0-edges)
- Induction proof gives

Theorem (Shmoys-Tardos' 93): Generalized Assignment LP can be rounded to give solution with optimal cost and makespan at most 2P

Global Counting Argument

- Suppose for contradiction there is no 1-edge, no machine with degree 1 or 2
- No 1-edge, so degree of jobs at least 2
- No machine with degree 1 or 2, so degree of machines at least 3
- So # edges ≥ (2 # jobs + 3 # machines)/2
- But, number of tight constraints at most number of jobs plus number of machines
- (2 # jobs + 3 # machines)/2

  > # jobs + # machines

  (contradiction)

Iterative Method: Key Ingredients

1. Small number of independent tight constraints at extreme point solution implies large valued element
2. Bound number of independent tight constraints at extreme point
3. Incorporate side constraints into the argument for 1 with appropriate relaxation

Degree-bounded MSTs

- Given graph with edge costs, integer k, find a spanning tree with maximum degree at most k of minimum cost

- NP-hard: k=2 same as minimum cost Hamiltonian path
- With non-metric costs, no approximation of cost possible without violating the degree bound

Base Problem and Constrained Problem

Spanning Tree Problem

Deg-bounded MST problem

$$\min c(T)$$

s.t.

$$T \in SP(G)$$

$$\deg_T(v) \le k \quad \forall v \in V$$

Outline

- Preliminaries
- Iterative Relaxation (Global Argument)
  - Assignment
  - Generalized Assignment
- Iterative Relaxation (Local Argument)
  - Minimum Spanning Tree
  - Degree-bounded Min-cost Spanning Tree
- [Alternate proof of SNDP rounding]
- Extensions, Open problems

Spanning Tree Polyhedron

Linear Programming Relaxation

$$\min \sum_{e \text{ in } E} c_e x_e$$

$$\text{s.t. } \sum_{e \text{ in } E(V)} x_e = |V|-1$$ (Any tree has n-1 edges)

$$\sum_{e \text{ in } E(S)} x_e \leq |S|-1 \;\forall\; S \subseteq V$$

$$x_e \geq 0 \;\forall e \text{ in } E$$

Subtour elimination constraints

E(S): set of edges with both endpoints in S

- Equivalent compact formulations [Wong '80]
- Polynomial time separable [Cunningham '84]

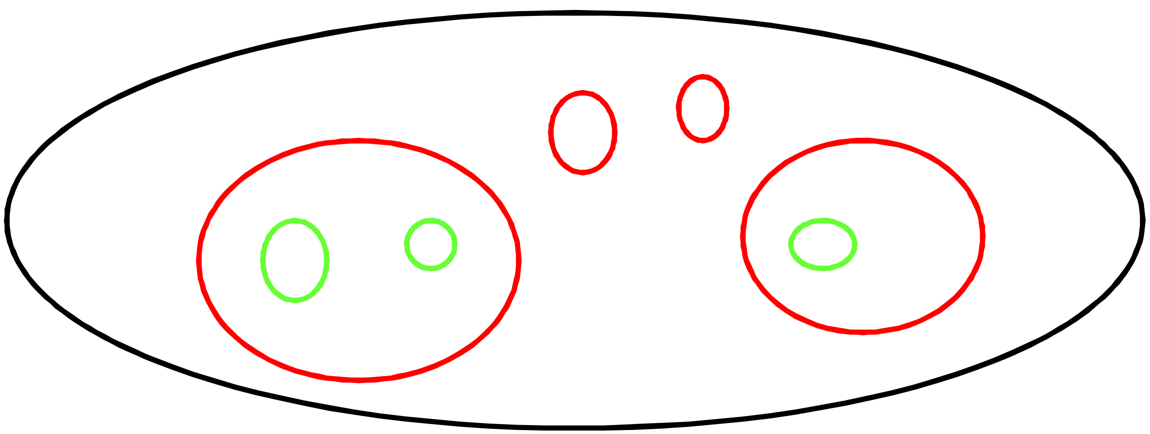
Theorem (Edmonds '71): Extreme points x^* of the above relaxation are integral

Extreme Points of Spanning Tree Polyhedron

Claim: Independent set of tight constraints defining x^* can be chosen s.t. corresponding subsets of vertices form a laminar family L [Cornuejols et al '88]

Follows from standard uncrossing arguments [Edmonds '71]

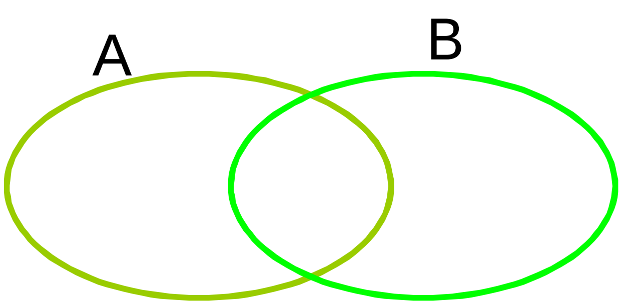
Tight Sets: $x(E(A)) = |A|-1$

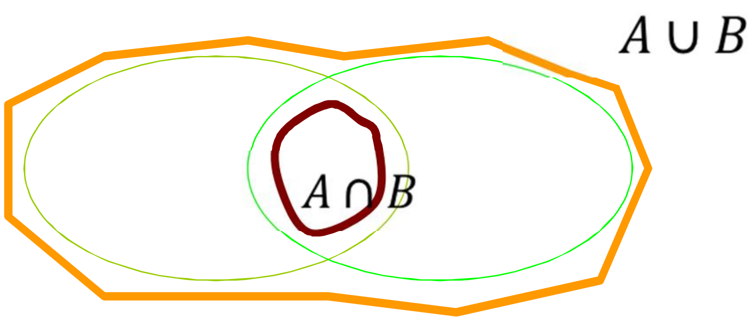


Extreme Points of Spanning Tree Polyhedron

Uncrossing Argument

$$
\begin{array}{ccccccc}
x(E(A \cap B)) & + & x(E(A \cup B)) & \geq & x(E(A)) & + & x(E(B)) \\
\leq & & \leq & & = & & = \\
|A \cap B| - 1 & + & |A \cup B| - 1 & = & |A| - 1 & + & |B| - 1
\end{array}
$$





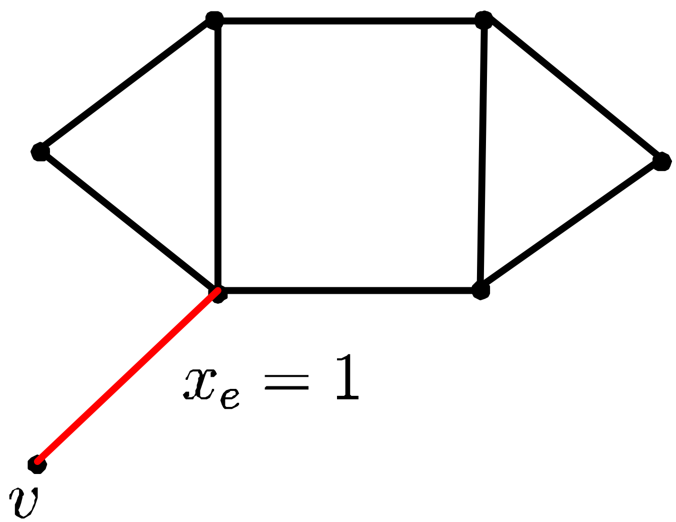
Iterative Proof of Integrality

$\sum_{e \text{ in } E} c_e x_e$

s.t. $\sum_{e \text{ in } E(V)} x_e = |V|-1$

$\sum_{e \text{ in } E(S)} x_e \leq |S|-1 \quad \forall S \subseteq V$

$x \geq 0$



While G is not a singleton

- Solve LP to obtain extreme point x*
- Remove all edges s.t. $x^*_e = 0$
- Contract all edges s.t. $x^*_e = 1$

Counting Argument

Claim: Support E of any extreme point x* of the LP has an edge with x*-value 1.

Proof Approach: Assume no such edge

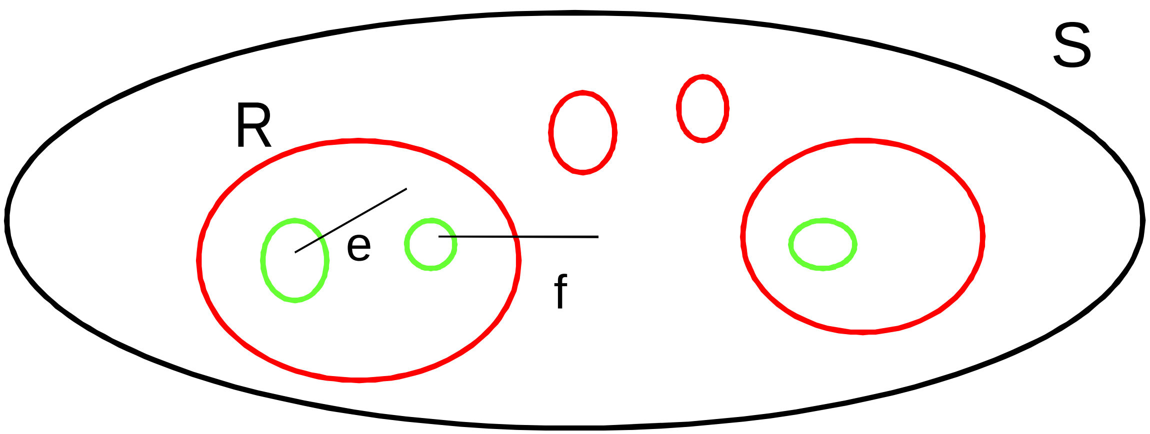
- x* is extreme implies independent tight constraints form a laminar family L
- Assign one token per edge in E and collect one per tight set in laminar family: show leftover token
- Contradicts row rank = column rank for tight linear subsystem defining extreme point x*

Fractional Token Redistribution

Definition: An edge in the support *belongs* to a set in L if it is the smallest set containing both ends of e

E.g., e belongs to R; f belongs to S

Edge e gives x_e *of its token to the set it belongs to*



Fractional Token Collection

Definition: An edge in the support *belongs* to a set in L if it is the smallest set containing both ends of e

Claim: The x_e tokens from edges that belong to a tight set in L can pay for it (i.e., give it one unit)

- Leaf sets S have $x(S) = |S| - 1 \geq 1$
- For others,

$x(S) - \sum_{\text{children } C} x(C) = (|S|-1) - \sum_{\text{children } C} (|C| - 1)$
$\neq 0$ by independence

- Every edge has $(1-x_e)$ left over for contradiction

Outline

- Preliminaries
- Iterative Relaxation (Global Argument)
  - Assignment
  - Generalized Assignment
- Iterative Relaxation (Local Argument)
  - Minimum Spanning Tree
  - Degree-bounded Min-cost Spanning Tree
- [Alternate proof of SNDP rounding]
- Extensions, Open problems

Bounded Degree MST

Extend spanning tree polyhedron

$$\min \sum_{e \text{ in } E} c_e \, x_e$$

$$\text{s.t.} \sum_{e \text{ in } E(V)} x_e = |V|-1$$

$$\sum_{e \text{ in } E(S)} x_e \leq |S|-1 \;\forall\, S \subseteq V$$

Spanning tree

$$\sum_{e \text{ in } \delta(v)} x_e \leq B_v \;\forall\, v \text{ in } W$$

Degree bounds

$$x_e \geq 0 \;\forall\, e \text{ in } E$$

(Note $W \subseteq V$)

Iterative Relaxation Algorithm

Initialize F = ϕ.

While W $\neq \phi$

1. Solve LP to obtain extreme point x^*.
2. Remove all edges e s.t. $x^*_e=0$.
3. (Relaxation) If there exists a vertex v in W such that $deg_E(v) \leq B_v+1$, then remove v from W (i.e., remove its degree constraint).

Key Claim: There is always a vertex to remove in (3.)

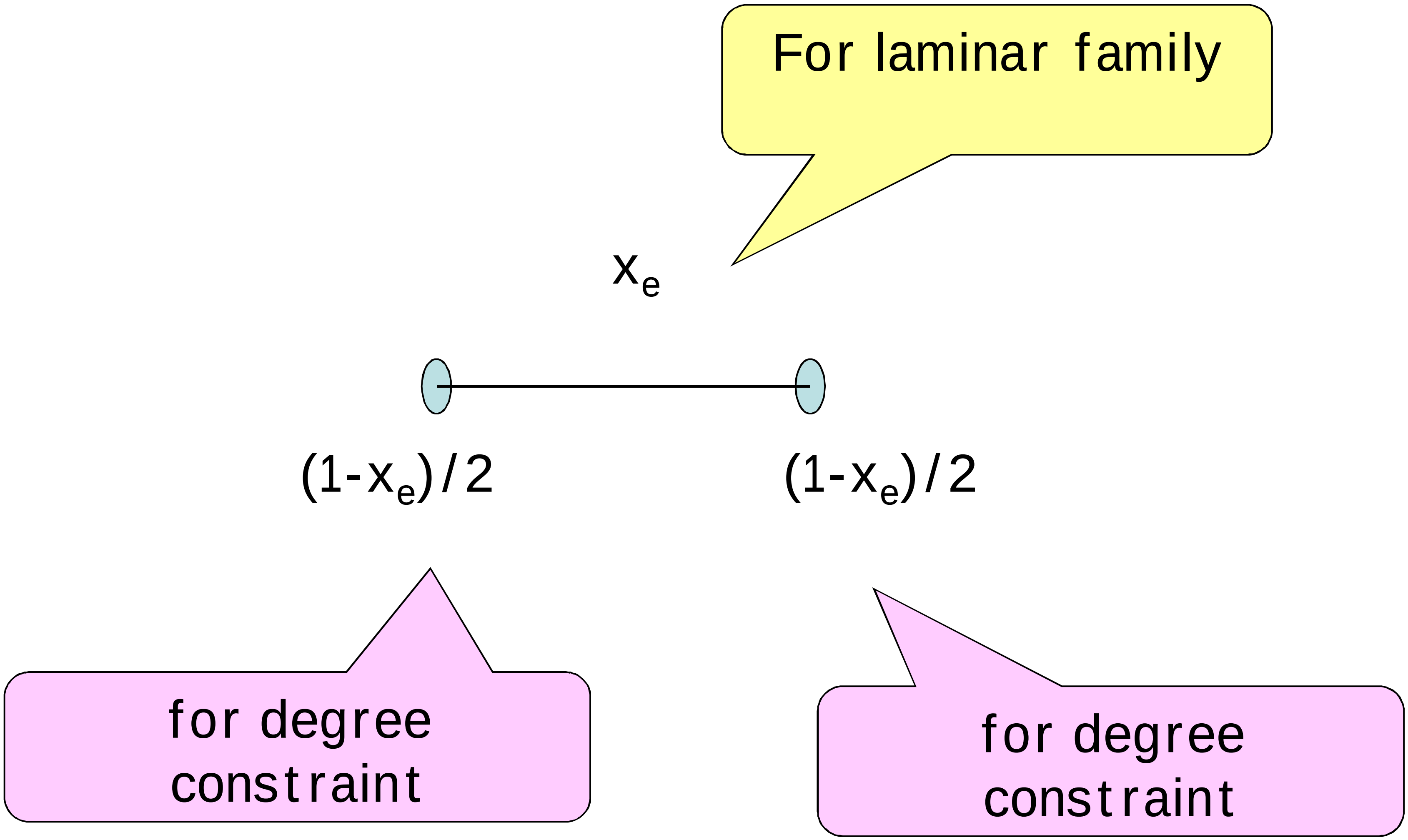
Local Counting Argument

Claim: x^* is extreme implies Independent tight constraints defining it form a laminar family L of subtour constraints and $T \subseteq W$ of tight degree constraints

Fractional Token Proof Outline:

- Assign 1 token per edge in support E
- Use x_e of each edge's token to "pay" for the laminar sets in L
- Use remaining $(1 - x_e)/2$ for each endpoint's degree constraint in T
- All edge tokens used contradicts independence of L U T constraints

Token Redistribution



Fractional Token Argument

Definition: An edge in the support *belongs* to a set in L if it is the smallest set containing both ends of e

Claim: Tokens from edges that belong to a tight set in L can pay for it (i.e., give it one unit)

(same proof as before)

Claim: Tokens from edges incident to node t in T can pay for its degree constraint

Tokens = $\sum_{e \text{ in } \delta(t)} (1 - x_e)/2 = (\deg_E(v) - B_v)/2 \geq 1$

(by relaxation condition)

Fractional Token Argument

If all tokens from E are "used up" in paying for sets in L and T, their constraints are dependent

All edge tokens x_e used up in laminar sets → $\sum_{e \text{ in } E} x_e$ = sum of constraints of maximal sets in L

All edges tokens $(1 - x_e)/2$ used up in T implies Every edge in E is incident to T or

$$\sum_{t \text{ in } T} \sum_{e \text{ in } \delta(t)} x_e = \sum_{e \text{ in } E} x_e$$

Maximal tight sets in L and degree constraints in T are dependent!

History: Degree-bounded MSTs

Reference	Cost Guarantee	Degree
Furer and Raghavachari '92	Unweighted Case	k+1
	Not possible	k
Konemann, R '01, '02	O(1)	O(k+log n)
CRRT '05 '06	O(1)	O(k)
R, Singh 06	MST	k+p (p=# distinct costs)
Goemans '06	1	k+2
Singh, Lau '07	1	k+1

Related Work

- First use of iterative relaxation with rounding for degree-bounded SNDP [Lau, Naor, Salvatipour and Singh STOC '07]
- Degree bounded matroids & submodular flows [Kiraly, Lau and Singh, IPCO'08]
- First use of fractional token argument [Bansal,Khandekar,Nagarajan STOC '08]
- Fractional token argument for LP extreme points for STSP and SNDP [Nagarajan, Ravi, Singh '08]

Outline

- Preliminaries
- Iterative Relaxation (Global Argument)
  - Assignment
  - Generalized Assignment
- Iterative Relaxation (Local Argument)
  - Minimum Spanning Tree
  - Degree-bounded Min-cost Spanning Tree
- [Alternate proof of SNDP rounding]

skip

- Extensions, Open problems

Survivable Network Design Problem

Given undirected graph with edge costs, find minimum cost subgraph with r_i edge disjoint paths between (s_i,t_i).

- Special cases
  - Shortest path, Spanning tree
  - Steiner tree
  - Generalized Steiner Forest
  - k-edge-connected subgraph

SNDP LP Relaxation

$$\text{Min} \sum_{e \in E} c_e \, x_e$$

$x(\delta(S)) \geq r_i \quad \forall \; S \subseteq V$ separating s_i and t_i

$x_e \geq 0 \quad \forall$ e in E

$\delta(S)$ = set of edges with exactly one end in S

Theorem (Jain): Any extreme point x of the above relaxation has an edge e with $x_e \geq ½$

Corollary: There is a polynomial-time 2-approximation algorithm for SNDP

Skew Supermodular function

f is skew supermodular iff

$$f(A \cup B) + f(A \cap B) \geq f(A) + f(B)$$

OR

$$f(A-B) + f(B-A) \geq f(A) + f(B)$$

E.g. $f(S) = \max r_i$ over (s_i, t_i) separated by S

LP Relaxation

$$\text{Min} \sum_{e \text{ in } E} c_e \; x_e$$

$$x(\delta(S)) \geq f(S) \quad \forall \; S \subseteq V$$

$$1 \geq x_e \geq 0 \quad \forall \; e \text{ in } E$$

Recall $f(S) = \max r_i$ over (s_i, t_i) separated by S

Theorem (Jain): Any extreme point x of the above relaxation for integral skew super-modular f(.) has an edge e with $x_e \geq ½$

Edge Boundaries are Strongly Submodular

$x(\delta(S))$ is strongly submodular iff

$$x(\delta(A \cup B)) + x(\delta(A \cap B)) \leq x(\delta(A)) + x(\delta(f(B))$$

AND

$$x(\delta(A-B)) + x(\delta(B-A)) \leq x(\delta(f(A)) + x(\delta(f(B))$$

Note: f(.) – x(.) is also skew-supermodular if

x(.) is strongly submodular

Jain's Iterative Rounding Algorithm

Initialize $F = \phi$; $f' = f$

While $f' \neq \phi$

1. Solve LP to obtain extreme point x^*.
2. Remove all edges e s.t. $x^*_e = 0$.
3. (Rounding) If there exists an edge e in E such that $x^*_e \geq \frac{1}{2}$ add e to F, and update $f'(S) = f'(S) - |e \cap \delta(S)|$

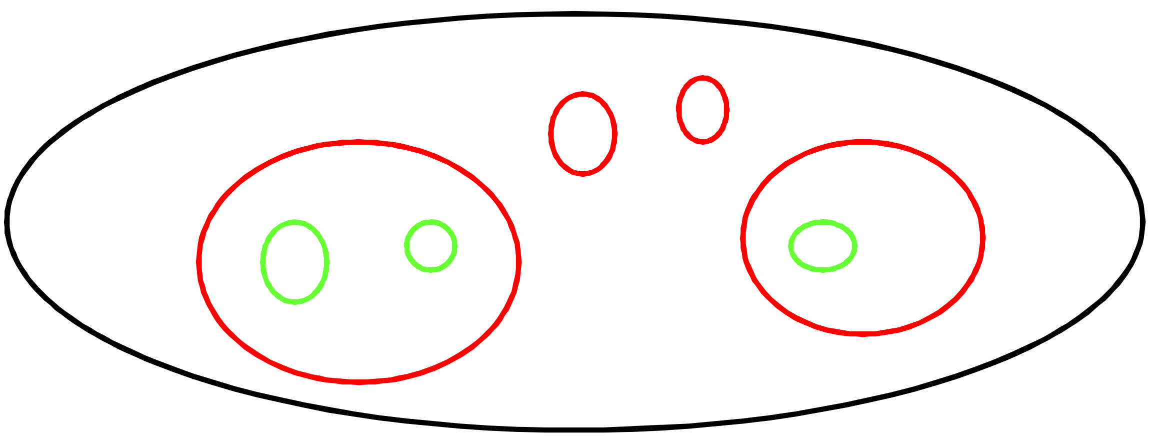
Remark: Updated f' is also skew-supermodular

Key Claim: There is always an edge to add to F in (3.)

Extreme Points

Claim: Independent set of tight constraints uniquely defining x* can be chosen s.t. corresponding subsets of vertices form a laminar family L [Jain '98]

Tight Sets: $x(\delta(S)) = f(S)$

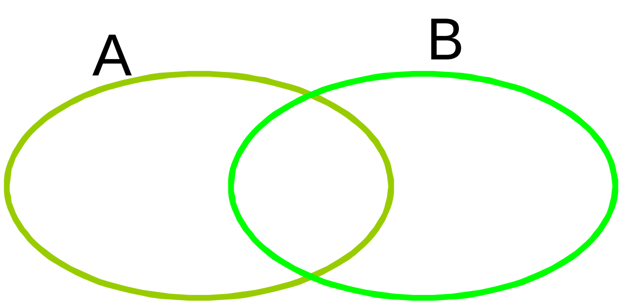


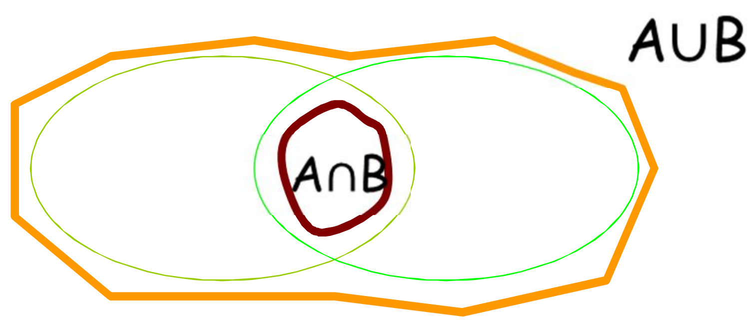
Uncrossing Argument

$$f(A \cup B) \quad + \quad f(A \cap B) \quad \geq \quad f(A) \quad + \quad f(B)$$

$$\leq \qquad\qquad \leq \qquad\qquad\qquad = \qquad\qquad = $$

$$x(\delta(A \cup B)) + x(\delta(A \cap B)) \leq x(\delta(A)) + x(\delta(f(B)))$$





Iterative Local Proof

Lemma: x* is extreme implies Independent tight constraints defining it form a laminar family L

Proof Approach:

- Assume for contradiction no $x_e \geq 1/2$
- Show number of nonzero variables is greater than number of tight constraints at extreme points
- Contradicts row rank = column rank for tight linear subsystem defining x*

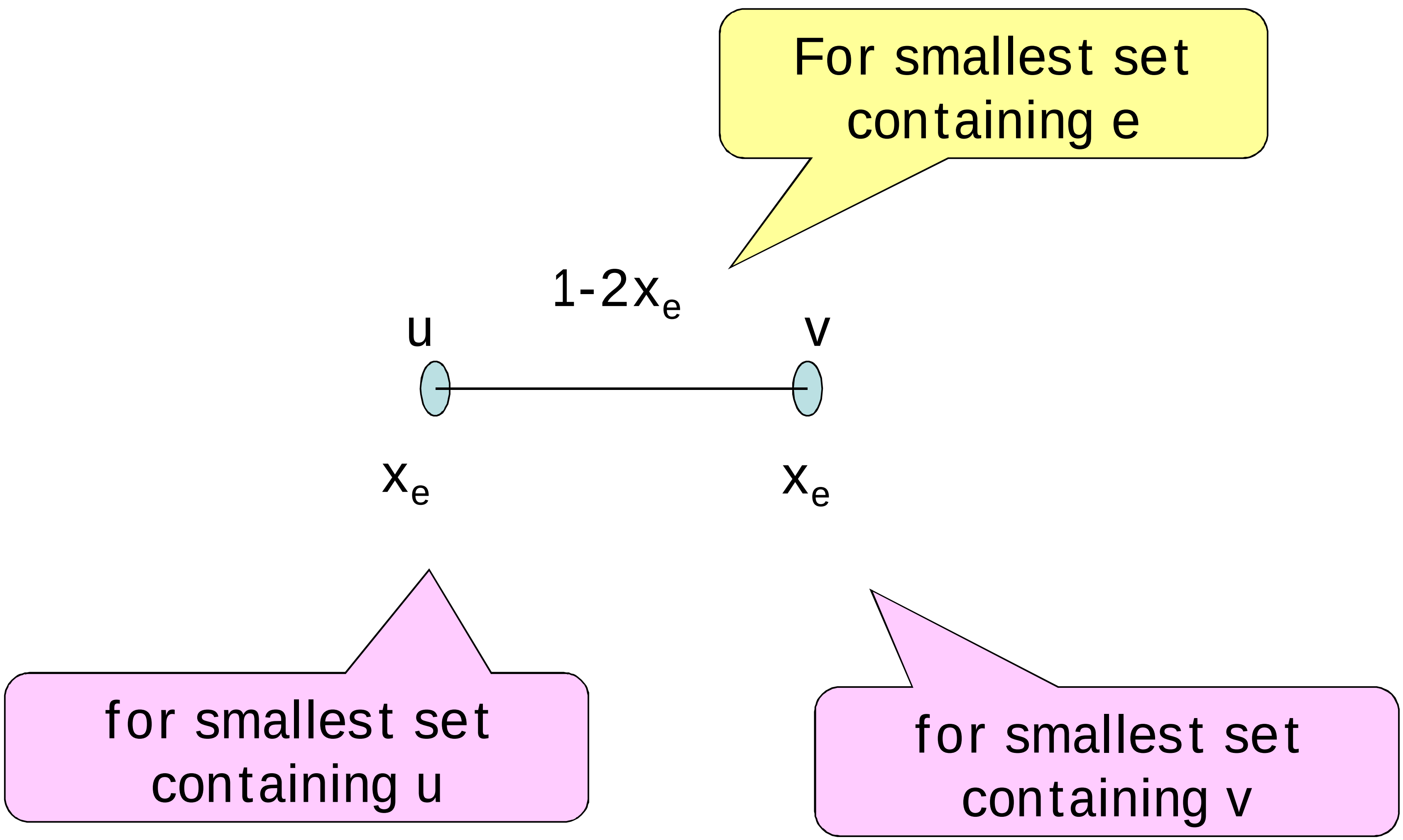
Iterative Local Proof

Lemma: x^* is extreme implies Independent tight constraints defining it form a laminar family L

Fractional Token Redistribution:

- Assign total of 1 token per edge (u,v) in E
- Assign x_e of its token to each of the smallest sets containing u and v
- Assign remaining $(1-2x_e)$ to the smallest set containing both u and v
- All of the above are nonzero if no $x_e \geq 1/2$

Token Redistribution

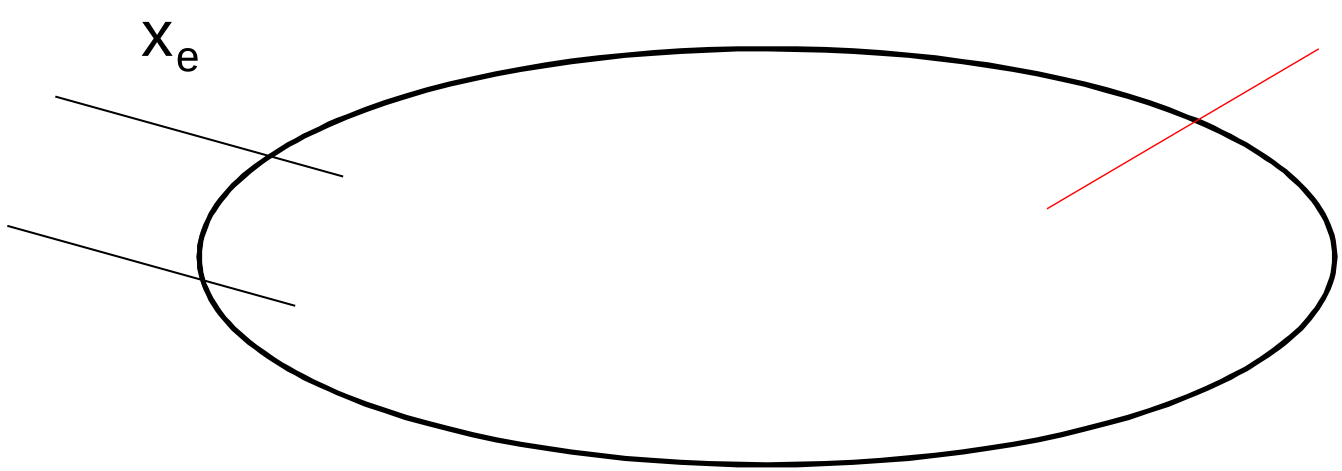


Iterative Local Proof

Claim: Every set in L receives at least one token

Consider leaf set S

$$x(\delta(S)) = f(S) \neq 0$$



Iterative Local Proof

Claim: Every set in L receives at least one token

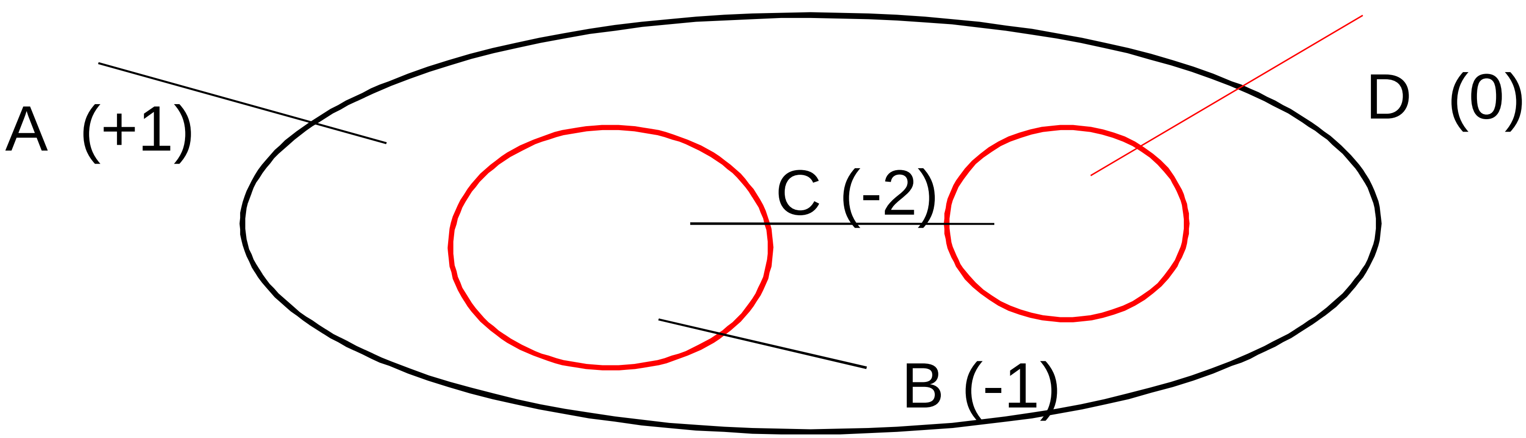
Consider set S with children R1, R2, ..., Rk in L

$x(\delta(S)) - x(\delta(R1)) - x(\delta(R2)) - \ldots - x(\delta(Rk))$

$= f(S) – f(R1) – f(R2) - \ldots - f(Rk)$

$\neq 0$ (else these sets are dependent)

$= x(A) – x(B) – 2x(C)$



Iterative Local Proof

Claim: Every set in L receives at least one token

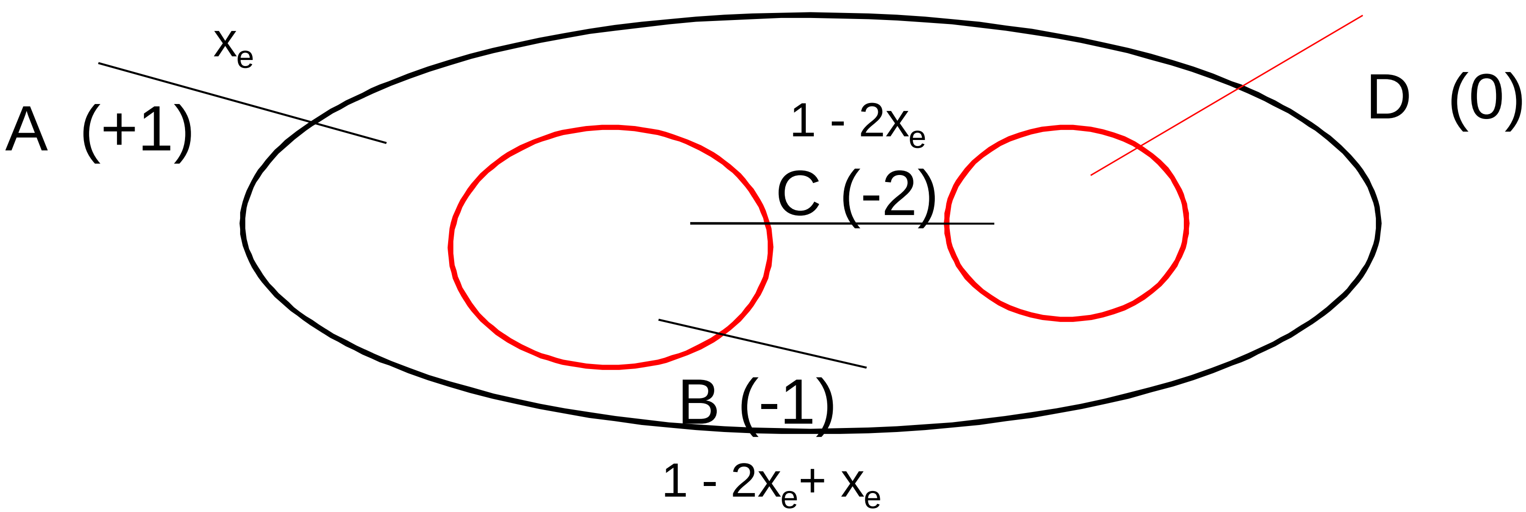
Consider set S with children R1, R2, ..., Rk in L

Tokens assigned to S

$= \sum_{e \text{ in } A} x_e + \sum_{e \text{ in } B} (1-x_e) + \sum_{e \text{ in } C} (1 - 2x_e)$

$= x(A) + |B| - x(B) + |C| - 2x(C)$

$= |B| + |C| + \text{nonzero integer}$



Iterative Local Proof

Not all tokens from E are "used up" in paying for sets in L

Consider a maximal set S in L

It has at least one edge (u,v) leaving it

No set in L contains both u and v, and hence its $(1 - 2x_{uv})$ token is unassigned

SNDP LP Relaxation

$$\text{Min} \sum_{e \text{ in } E} c_e \; x_e$$

$$x(\delta(S)) \geq f(S) \quad \forall \; S \subseteq V$$

$$1 \geq x_e \geq 0 \quad \forall \; e \text{ in } E$$

Denote $f(S) = \max r_i$ over (s_i, t_i) separated by S

Theorem (Jain): Any extreme point x of the above relaxation for integral skew super-modular f(.) has an edge e with $x_e \geq ½$

Iterative Method

- Inductive method for finding (near) optimal solutions from linear programming relaxations
- Overview
  - Formulate generic LP relaxation
  - Identify element with high fractional value to
    - **(Round**) Pick element in solution **or**
    - (**Relax**) Remove some constraints whose violation can be bounded
  - Formulate residual problem in generic form and iterate (Prove by induction)

Iterative Method: Key Ingredients

1. Small number of independent tight constraints at extreme point solution implies large valued element
2. Bound number of independent tight constraints at extreme point

3. Incorporate side constraints into the argument for 1 with appropriate relaxation

Examples of Base Problems

- Bipartite Matching and Vertex Cover
- Spanning trees (undirected and directed)
- Max weight matroid basis and 2-matroid intersection
- Rooted k-connected subgraphs
- Submodular Flows
- Network Matrices
- General Graph Matchings

Examples of Approximations

- Generalized Assignment, Maximum Budgeted Allocation
- Degree bounded variants of spanning trees, matroid bases, submodular flows and Survivable Network Design Problem
- Partial covering (e.g. vertex cover)
- Multi-criteria problems (spanning trees)
- Earlier results: Discrepancy, Unsplittable Flow, Bin packing
- Recent developments: Iterated randomized rounding for Steiner trees

Open Directions

- New proofs of integral polyhedra (e.g. TU matrices, TDI systems)
- Adding side constraints to other well behaved polyhedra (e.g. Network Matrices?)
- Traveling Salesperson Problems
- Packing problems (e.g. general unsplittable flow, degree bounded flow)
- Algorithms that avoid solving LP

Lap-chi Lau (CUHK) & Mohit Singh (McGill): co-authors on a monograph from Cambridge University Press. A non-printable copy is available on the web