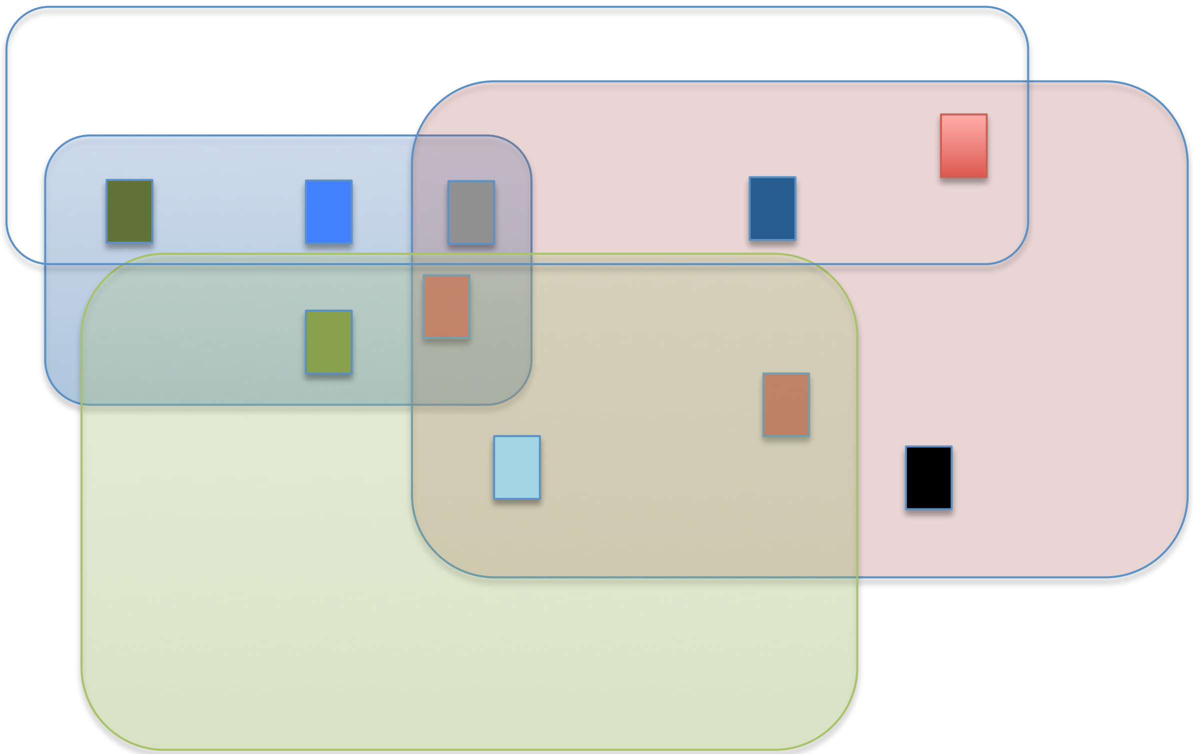
Set Cover Revisited: Hypergraph Covering with Hard Capacities

Barna Saha
AT&T Labs
Samir Khuller
U. Maryland

ICALP 2012

Set Cover

- Central problem with MANY applications.

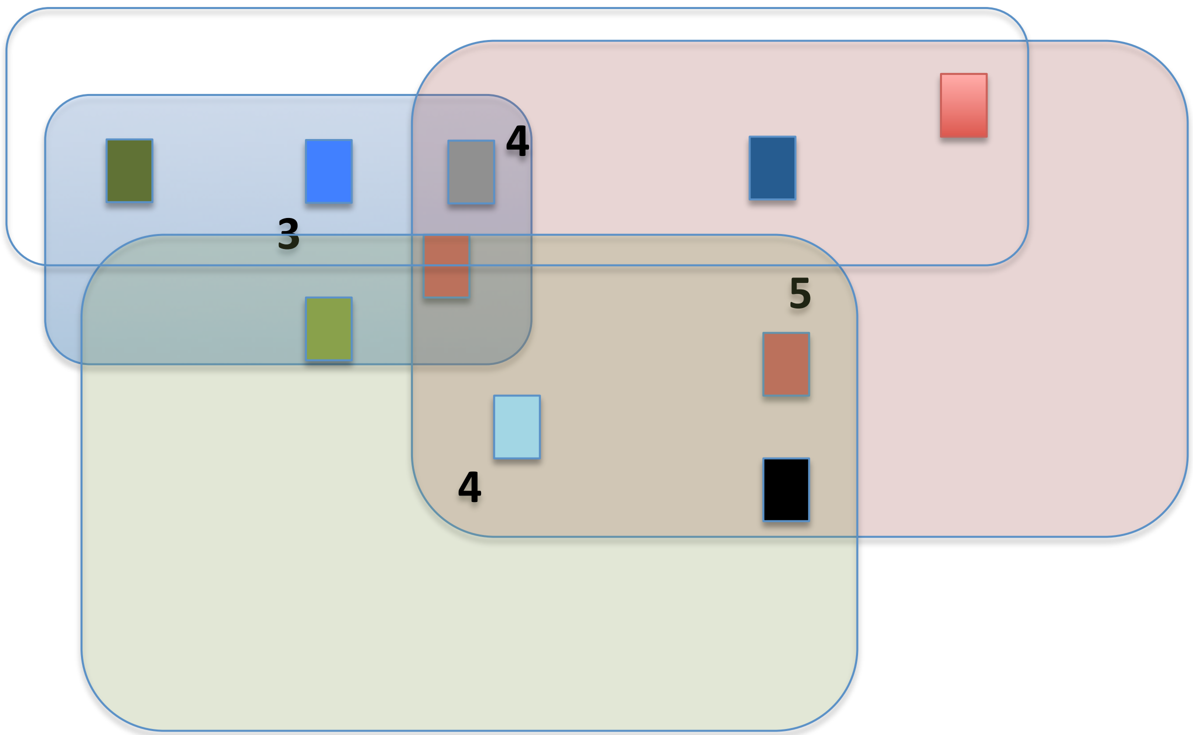


Basic Set Cover

- Extremely well studied, with O(log n) approximations using greedy (Chvatal, Lovasz, Johnson).
- Special case of Vertex Cover (choose nodes to cover all edges), has factor 2 approximation.
- Extends to hypergraphs with hyper edges of size at most f, giving f approximation.

Set Cover with Capacities

- Suppose in addition, each edge has a capacity on the number of elements it can cover.

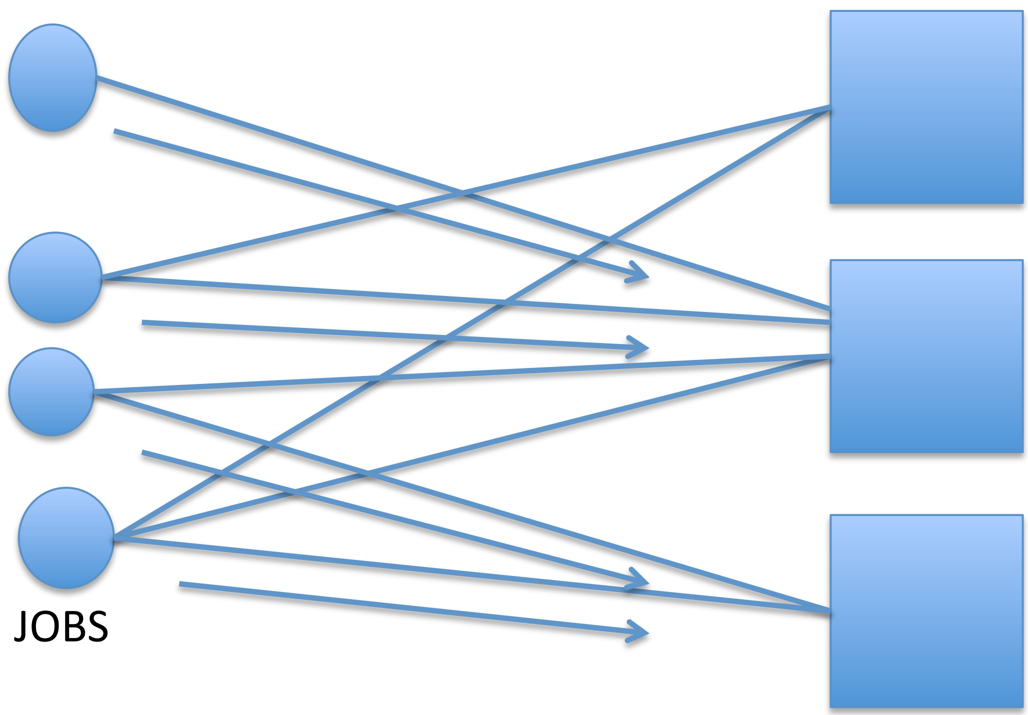


One Motivation: Saving Energy

Need to turn on machines to assign jobs……

Keep load on machines low.

Well studied problem for a given set of machines.

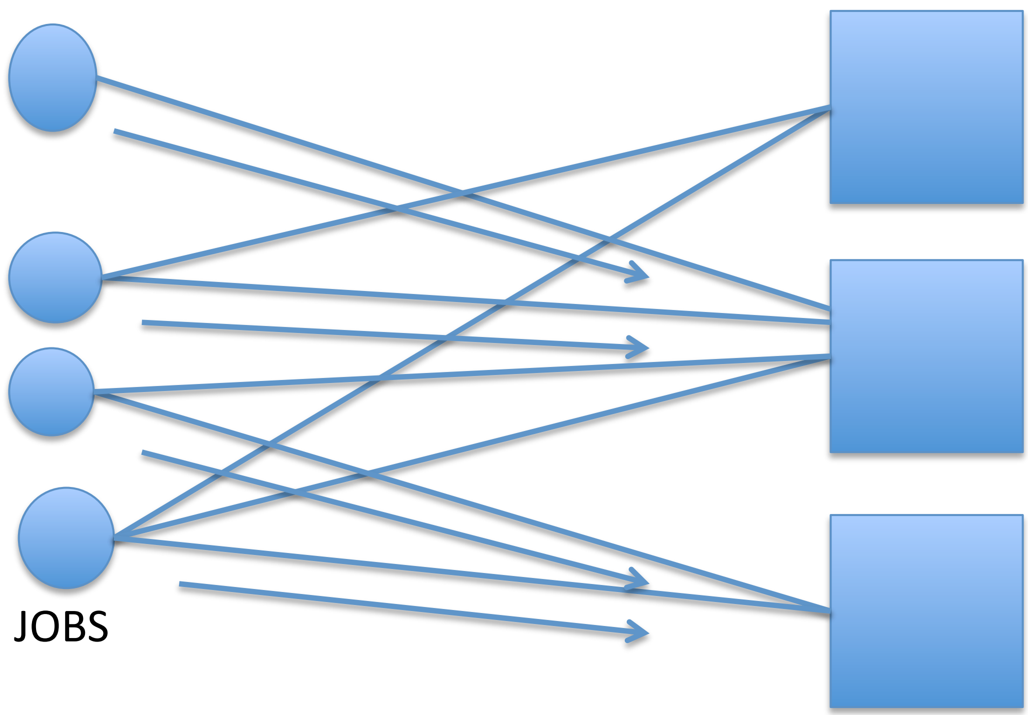


Set Cover with Capacities

Machines = Sets

Jobs = Elements

Load = Capacity



Machine Activation Problem [KLS SODA 2010]

- Related to Scheduling Jobs on Unrelated Machines [Lenstra, Shmoys, Tardos 90]
- We have N jobs and M machines and a p(i,j) is the processing time of job j on machine i.
- Objective: Minimize Makespan (largest processing time of jobs assigned to a machine).
- Our Problem: In addition we incur cost ci to buy machine i.
- Objective: Minimize makespan by spending C units to buy machines.

Main Result [KLS 2010]

- Suppose there is a cost C solution which assigns all jobs, with max load at most T.
- Our algorithm finds a solution with cost C.log n and max load at most 2T.
- However, if a job can only be done on a small number of machines, can we get a better approximation?

A greedy approach [KLS 2010]

- Given a set of machines S that are open, and a time bound T, let f(S) be the maximum number of jobs that can be done on machines in S.
- What is the incremental benefit of opening a new machine j?
- Not easy to compute this, since the problem of scheduling is NP-hard!
- Let f(S) be the "fractional benefit" instead!

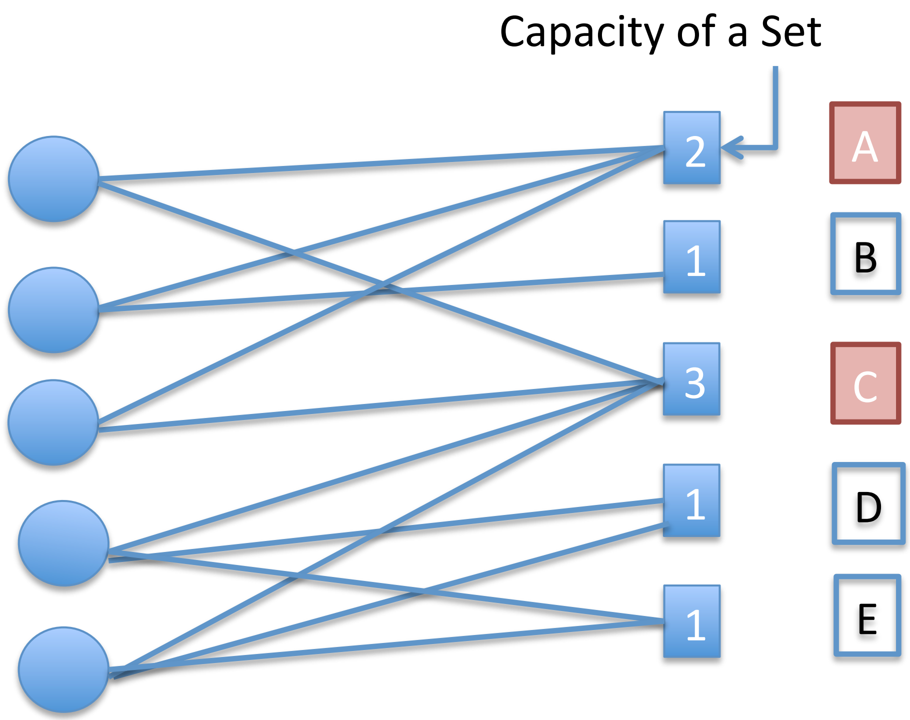
At each step make a greedy choice!

- Initially S={ }
- In each step, choose the machine j that maximizes f(S U j).
- Repeat until f(S)>(n-1)
- Works when all machines have same activation cost, look at f(S)/ci otherwise.

When >n-1 jobs are fractionally scheduled, can use [ST93] to convert it into an integral schedule!

Assigning N unit jobs

- Pick as few sets as possible, to cover all elements so that each set covers a small number of elements.



Wolsey's Approach (1982)

- Let f(S) be the largest number of elements that can be covered if we choose a collection of sets S (use NETWORK flow to compute f)
- Initially S={}
- At each set pick a new set Si that would increase f(S+Si) by the largest possible value.
- This gives a O(log n) approximation.
- Can we do better?

Vertex Cover with (soft) Capacities

- Given G=(V,E), and a capacity function k(v) pick the smallest collection of vertices to cover all edges (covering by stars). Nodes have weights.
- If we can pick multiple copies of a node, a 2-approximation exists [GHKO SODA 02] (works for hyper-graphs).
- NOTE: each element belongs to at most 2 sets.
- NOTE: With k(v) unbounded, easy 2 approx.

Weighted (hard) Capacitated VC is Set-Cover Hard [Chuzhoy, Naor 2002]!

- However, they show a factor 3 approximation for the unweighted case, separating the two problems in difficulty.
- Improved to a 2 approx [Gandhi, Halperin, Khuller, Kortsarz, Srinivasan ICALP 03]

LP Rounding for Unweighted Vertex Cover

$$\sum_{v \in V} x(v)$$

$$y(e,u) + y(e,v) = 1 \qquad \forall\, e = (u,v)$$

$$y(e,v) \leq x(v),\ y(e,u) \leq x(u) \qquad \forall e = (u,v)$$

$$\sum_{e=(u,v)} y(e,v) \leq k(v)x(v) \qquad \forall v$$

$$0 \leq x(v), y(e,v), y(e,u) \leq 1 \qquad \forall\, v \in V, \forall e = (u,v)$$

Rounding the LP solution

- Pick *v* with probability *2x(v)*.
- Add more vertices to cover remaining uncovered edges.
- Note that for each edge, at least one end point is chosen, but may not have available capacity…….
- Proof is quite difficult.
- However it is easy to bound the expected cost of the solution vs the LP cost.

Back to our Application

- We really have a hyper-graph since each job (element/edge) can be done on a small number of machines (typically 3).
- How do we decide which machines (set/node) to pick, so that all jobs can be assigned satisfying the load constraint?
- Main Difficulty: CN approach does not even work for multi-graphs (proof breaks down).
- Multi-graphs mentioned as an open problem.

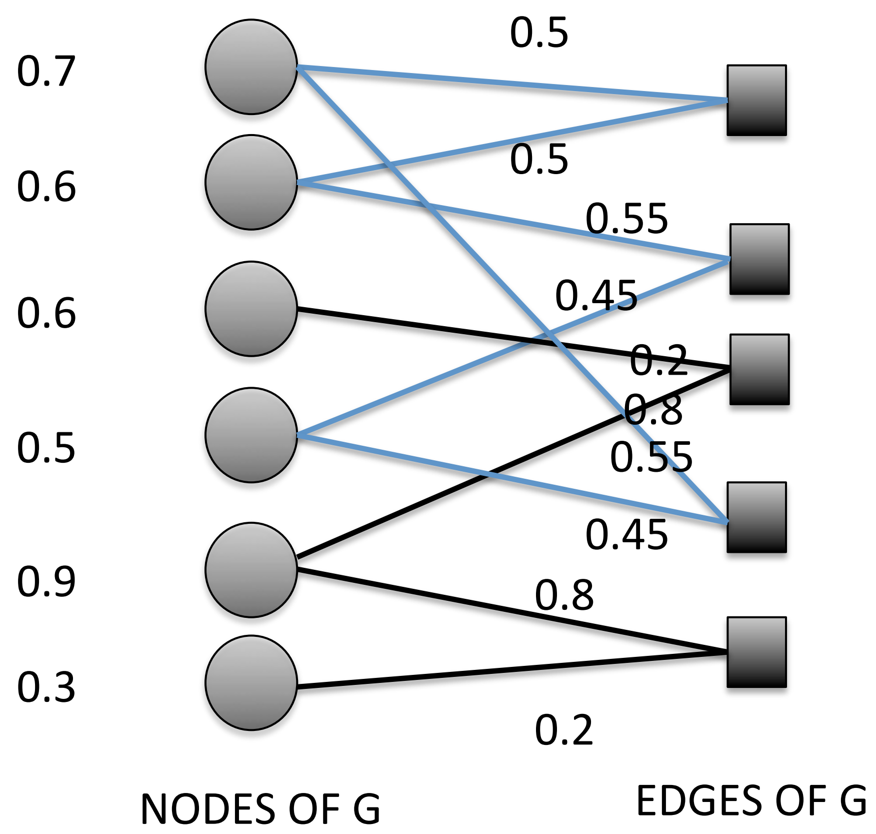
Hypergraph Cover with Capacities

- LP Rounding approach.
- Analyze the structure of an optimal fractional solution.
- Too many complications, lets focus on the structure first.
- In fact, we redesign a new algorithm for Capacitated VC using LProunding.
- This works for multi-graphs and hyper-edges of size f, and we get an O(f) approximation.

A Useful Property

- Partition edges of the graph into H1 and H2 based on whether y(e,v)=x(v) or not.
- H2: Only edges with y(e,v)=x(v)
- H1: rest
- In H1, we can perturb the y(e,v) values so that we “break” cycles, by either making the value 0 or by moving the edge into H2.

Example Graph



Rounding the LP

- H1 is acyclic, and this is very useful.
- Assigning edges in H2 is much easier

$$\sum_{e=(u,v)} y(e,v) \leq k(v)x(v)$$

- If the edge is in H2 and if we choose v, we can "scale up" y(e,v) and fully assign the edge.
- Example: x(v)=0.7 and k(v)= 6.
- Ex: 0.7+0.7+0.7+0.7+0.3+0.2+0.5+0.4 ≤ 6×0.7
- All v with x(v) larger than a fixed constant can be chosen and edges of H2 are dealt with!

Rounding the vertices

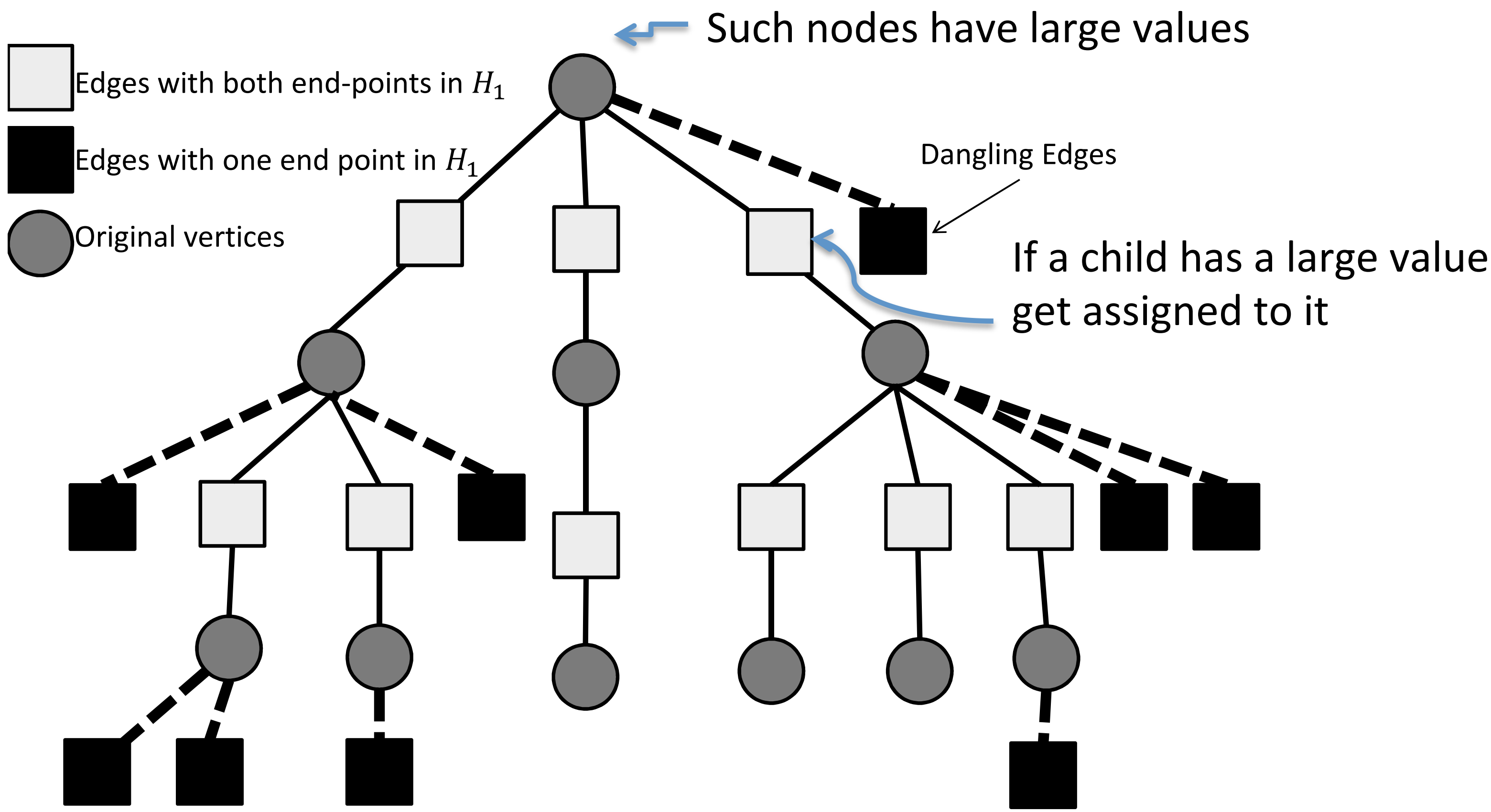


Fig 1a. Structure of H_1, dangling edges are colored black and connected by dashed lines, edges with both end-points in H_1

Main Difficulty

- Handling nodes with many dangling edges, whose other ends points have very small x(v) values.
- Use RANDOMIZED ROUNDING to handle this case, very involved proofs.
- Key is that we can cast it as a multi-set multi-cover problem with no capacities and take advantage of the fact that each node with many dangling edges have large x(v) values.

Main Difficulty

- Reduction to Multi-set Multi-cover (MSMC) Problem

  Each such node v is an element and each dangling edge (u,v) is a multi-set S_u containing v, $m(v)$ times where $m(v)$ is the multiplicity of the edge (u,v). If v has $L(v)$ dangling edges incident on it and can cover $l(v)$ dangling edges without violating the capacity, then in the multi-set multi-cover problem, v needs to be covered $L(v)-l(v)$ times.

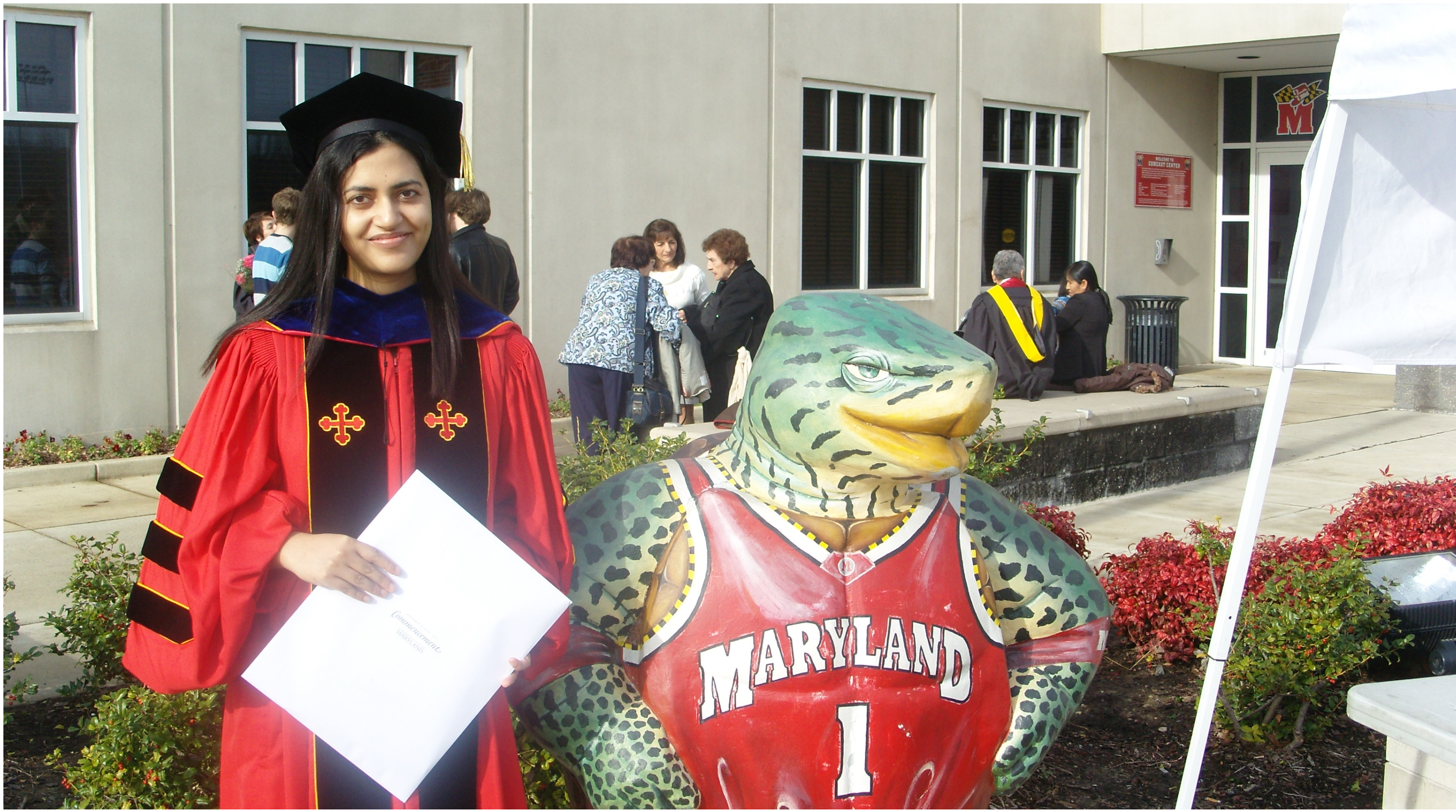
- Partially rounded fractional solution is feasible for the natural LP relaxation of MSMC.

- However, the randomized rounding algorithm cannot bound the rounded solution in terms of the LP objective of MSMC, but can charge the cost to the nodes {v} since they have large x(v) values.

Conclusions

- We conjecture that the correct answer is an f approximation (true for f=2!).
- We do get a 2f approximation but not for small f……still trying to optimize the bounds, we think we can make them much better (journal version).
- Combinatorial Approximation Algorithm?
- Online versions of these problems?

Details in her Ph.D. thesis!



Why isnt Barna here?