Lovász Local Lemma: Application to Santa Claus Problem

COS 597B, Spring 2010 Scriber: Rong Ge

February 17, 2010

1 Santa Claus Problem

In the Santa Claus Problem (see Fig.1), or Max-Min Allocation Problem, there are m children (later called players) and n presents (items), each item can be given to only one player. Each player has an evaluation of the items, and the goal is to find an allocation that maximizes the min value of happiness of players. More formally it is defined as:

Definition 1 (Santa Claus Problem)**.** There are m players and n items, for every $i \in [m]$ and $j \in [n]$ the input specifies $v_{ij} \geq 0$, which is i-th player's evaluation of item j. An allocation is a function $f : [n] \to [m]$, $f(j) = i$ iff item j is given to player i. Player i's happiness is defined as $\sum_{j:f(j)=i} v_{ij}$, and the goal is to find an allocation that optimizes the value

$$\max_f \min_{i \in [m]} \sum_{j:f(j)=i} v_{ij}$$

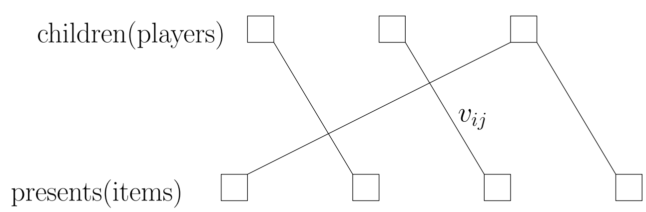

Figure 1: A Santa-Claus Instance

A similar problem is the Load-Balancing Problem (Min-Max Allocation), where there's a 2-approximation algorithm via LP. The Santa Claus Problem with general evaluation functions is difficult, and people are interested in special cases. In this lecture we'll focus on the case where $v_{ij} \in \{0, v_j\}$: each item j has a value v_j, and each player either want the item ($v_{ij} = v_j$) or do not want the item ($v_{ij} = 0$).

Bansal and Sviridenko [BS06] showed that this problem can be reduced to a combinatorial question. In the combinatorial question (see Fig. 2), there are m players in m/l groups of size l. Each group has $l - 1$ big items (that has value k). Each player is interested in k small items (that has value 1). Each small item is interested by at most l players (the bipartite graph in Fig. 2 only shows players and small items). Use S_i to denote the set of small items player i wants, the goal is to pick a subset of players $P \subseteq [m]$ where exactly one player is picked from each group, and then for each $i \in P$, pick a subset $T_i \subseteq S_i$, such that all T_i's are pairwise disjoint, and the min size of the sets T_i is maximized.

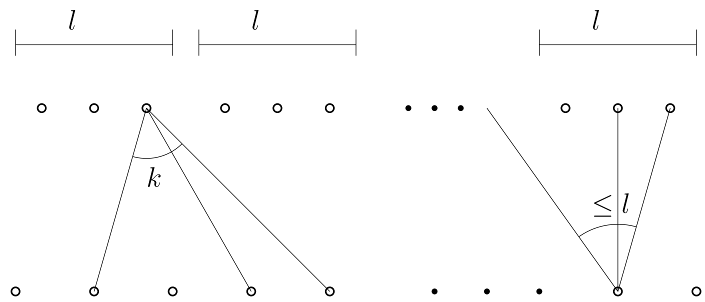

Figure 2: The Combinatorial Question

Theorem 1 ([BS06]). *If each T_i contains at least γk elements, the integrality gap of configuration LP is $O(1/\gamma)$. More over, if there's an efficient algorithm that can pick such sets T_i, there is an efficient rounding algorithm that gives approximation ratio $O(1/\gamma)$ for the problem.*

A naive way to choose the sets is: randomly choose a player from each group, then for each item randomly give it to a player who's interested in the item. It's easy to see that the expected number of requests for any item is at most 1.

However, like in the balls and bins problem, many items will be requested $\log m/\log\log m$ times, so this naive idea cannot give a constant approximation algorithm. [BS06] uses a similar idea to achieve approximation ratio $\log\log m/\log\log\log m$. In the next two sections we'll see two proofs showing γ is always at least some constant. The last proof can be made explicit and efficient by using explicit versions of Lovász Local Lemma.

2 Proof I: Hypergraph Matching

The first proof by Asadpour, Feige and Saberi [AFS08] uses *matchings in hypergraphs.* To explain the proof we need some definitions

Definition 2 (Hypergraph)**.** A hypergraph $H = (V, E)$ contains a vertex set V, and an (hyper-)edge set E, each $e \in E$ is a subset of V.

Definition 3 (Hypergraph Matching)**.** A matching of a hypergraph H is a subset M of edges, where no two edges share a common vertex, i.e., $\forall e_1, e_2 \in M$, $e_1 \cap e_2 = \Phi$.

Definition 4 (Bipartite Hypergraph)**.** A hypergraph H is bipartite if the vertex set is the disjoint union of two sets U and V, and for all $e \in E$, $|e \cap U| = 1$.

Definition 5 (Perfect Matching)**.** A perfect matching for a bipartite hypergraph $H = (U \cup V, E)$ is a matching M such that for any $u \in U$, there is an edge $e \in M$, $u \in e$.

Definition 6 (Traversal)**.** A traversal of a hyper graph H is a set of vertices that intersects every hyperedge in H. $\tau(H)$ is the minimum size of a traversal of H.

An extension of Hall's Theorem to hypergraphs is proved by Aharoni and Haxell [AH00],

Theorem 2. *For a bipartite hypergraph $H = (U \cup V, E)$, let C be a subset of U, define a new hypergraph $H_C = (V, E_C)$, where E_C contains all edges that has a vertex in C minus that vertex. More formally $E_C = \{e \cap V : e \in E \text{ and } e \cap C \neq \Phi\}$.*

If for all $e \in E$ $|e \cap V| \leq r - 1$ and for all $C \subseteq U$ $\tau(E_C) > (2r-3)(|C|-1)$ then there exists a perfect matching in H.

Notice that this theorem is Hall's theorem when $r = 2$.

When we apply this Theorem to the combinatorial question, construct a graph $H = (U \cup V, E)$ as follows: U is the set of groups ($|U| = m/l$), V is the set of small items $|V| = n$. For any subset T_i of S_i, if $|T_i| \geq \gamma k$, then there's an edge $e = T_i \cup \{\text{group of player } i\}$.

In this case $r = \gamma k + 1$. To bound the value of τ, take any subset $C \subseteq U$, look at the graph H_C, for any set S_i, since all its subsets with at least γk items form hyper-edges, at least $k - \gamma k + 1$ items must appear in traversal. There are $|C| \cdot l$ such sets, but an item may be counted at most l times, therefore

$$\tau(H_C) \geq \frac{l \cdot (k - \gamma k + 1) \cdot |C|}{l} = |C| \cdot (k - \gamma k + 1)$$

When $\gamma \leq 1/3$, $\tau(H_C) > (2r - 3)(|C| - 1)$ for any C. By Haxell's Theorem there's a perfect matching, and this matching correspond to a way to allocate the items to players. So when $\gamma \leq 1/3$ it's always possible to allocate the items where at least γk items are given to one of the players in each group.

3 Proof II: Lovász Local Lemma

3.1 Overview

This proof is given by Feige [Fei08]. The proof considers system of sets defined as follows.

Definition 7 ((k, l, β) system)**.** A (k, l, β) system is a system of sets, where each set (sometimes called player) has size k, sets are partitioned into groups of size l, each element (sometimes called item) is contained in at most βl sets. We say a (k, l, β) system is γ feasible if there's an allocation such that each group has a player who gets at least γk items.

Notice that (k, l, β) system do not specify the value of n and m. Some cases of (k, l, β) systems are easy: when k is a constant, finding a matching in the bipartite graph defined by groups and items will give a $\gamma = 1/k = \Omega(1)$ allocation. When l is a constant, use Hall's Theorem or pick a random set from each group will give a $\gamma = 1/\beta l = \Omega(1)$ allocation (in all the systems we are considering β is always bounded by a constant).

The proof relies on the following two lemmas to do reductions between (k, l, β) systems.

Lemma 3 (Reduce-l)**.** *There exists a constant C, when $l \geq k > C$, a (k, l, β) system can be reduced to a (k, l', β') system where if the new system is γ feasible, the original system is also γ feasible. $l' \leq \log^5 l$, $\beta' \leq \beta(1 + \frac{1}{\log l})$.*

Lemma 4 (Reduce-k)**.** *There exists a constant C, when $k \geq l > C$, a (k, l, β) system can be reduced to a (k', l, β) system where if original system is not γ feasible, then the new system is not γ' feasible. $k' \leq k/2$, $\gamma' = \gamma(1 + \frac{3 \log k}{\sqrt{\gamma k}})$.*

Using the two Lemmas it's easy to show the following theorem:

Theorem 5. *There exists $\epsilon > 0$ such that any $(k, l, 1)$ system is ϵ feasible.*

Proof. (sketch)

By induction, when $k < C$ or $l < C$ the problem falls into one of the easy cases. Otherwise if $l \geq k$ use Reduce-l, if $k > l$ use Reduce-k until one of the easy cases is reached. It's easy to prove the ratio between the original system and the new system is bounded by a constant (the overhead in Reduce-l only depends on l and the overhead in Reduce-k only depends on k, it's enough to prove that two product series converge to a finite value, for more detail see [Fei08]). ☐

3.2 Reduce-l

The basic idea in proving Reduce-l and Reduce-k is sampling. For Reduce-l, sample some sets (players) from each group; for Reduce-k, randomly throw out half of the items. However in both cases we need to make sure that no bad events happen. We do this by using Lovász Local Lemma

Lemma 6 (Lovász Local Lemma)**.** *For a collection $\mathcal{A}$ of (bad) events, if there exists function $x : \mathcal{A} \to \mathbb{R}^+$ that satisfies $\Pr[A] \le x(A) \prod_{B \in \Gamma(A)} (1 - x(B))$ ($\Gamma(A)$ is the set of events that event A depends on), then*

$$\Pr[\cap_{A \in \mathcal{A}} \overline{A}] \ge \prod_{A \in \mathcal{A}} (1 - x(A))$$

Now we prove Lemma Reduce-l.

Proof. (Reduce-l)
Sample $l' = \log^5 l$ sets from each group, let d be the degree of an arbitrary item. It's easy to see in the new system,

$$\mathrm{E}[d] \le \beta l'$$

Since the sampling only has negative correlations (between sets within the same group), Chernoff Bounds can be applied to get

$$\Pr[d > (1 + \frac{1}{\log l})\beta l'] \le e^{-\log^3 l}$$

For each item, the event that its degree is larger than $(1 + \frac{1}{\log l})\beta l'$ is a bad event. Each bad events depend on at most $\beta l \cdot l \cdot k \le l^4$ other events, because this event depends on the choice of βl groups, each group has l sets and each set affects k other events. By the symmetric version of Lovász Local Lemma, there always exists a way to find a new system and reduce l. If the new system is feasible, exactly the same solution will work for the original system because the new system is only a subset of the original system.

This application of Lovász Local Lemma can be made efficient by the algorithm of Moser and Tardos [MT09]. □

3.3 Reduce-k

Proof. (Reduce-k)
For reduce k, we remove each item with probability $1/2$, and we want to prove if the new system is γ' feasible the original system is γ feasible. This is not easy because the value of the new system is not directly related to the value of the old system.

To see the actual relationship, we fix a way of choosing sets from groups, then the problem reduces to a bipartite graph matching problem (one side of the graph is the groups, the other side is the items), and we can apply Hall's Theorem. Consider the contrapositive, if the old system is not γ feasible, then for any way of choosing sets, by Hall's Theorem, there is a set of players of size i where the number of neighbors is less than $\gamma k \cdot i$. Since we are removing each item with probability $1/2$ independently, we would expect the number of neighbors in the new system is also small ($< \gamma' k' i$), which by Hall's Theorem will show that the SAME WAY of picking players is not good for the new system. (this kind of analysis also shows the way to get explicit allocation for original system from explicit allocation for the new system: just choose the same sets and run matching/max flow algorithm)

We define two kinds of bad events. To analyze the degree of dependence, we consider the overlap graph on sets (players), two sets are connected in the graph if and only if they are not disjoint (i.e. contain the same item). B_1 is the event that some set has less than $k' = (1 - \frac{\log k}{\sqrt{k}})\frac{k}{2}$ items. B_i is the event for a connected collection of sets from distinct groups of size i, such that the number of neighbors in the original system is less than $\gamma \cdot k \cdot i$, but in the new system the number of neighbors is at least $(1 + \frac{\log k}{\sqrt{\gamma k}}) \cdot k/2 \cdot i$ neighbors.

If no bad events happen and the new system is γ' feasible, it means there's a choice of players (one from each group), such that for any subset C of these players the number of neighbors (items) is at least $(1+\frac{\log k}{\sqrt{\gamma k}}) \cdot k/2 \cdot |C|$, and since B_i does not happen, the number of neighbors in the original system is also large. By Hall's Theorem we know the same choice of players will give a γ allocation.

Again, because of negative correlations Chernoff Bounds can be applied, and we have

$$\Pr[B_1] \le e^{-\log^2 k}$$

$$\Pr[B_i] \le e^{-c \cdot i \cdot \log^2 k}$$

Number of B_j events that a B_i event depends on is bounded by $ik\beta l$ ways to choose one of the sets in B_j (this set must intersect with the sets of B_i), and then $\beta kl \le k^3$ choices for other sets in B_j (because B_j is connected and the degree in overlap graph is βkl). See Fig. 3 for illustration. In conclusion, the number is no more than $ik^{O(j)}$.

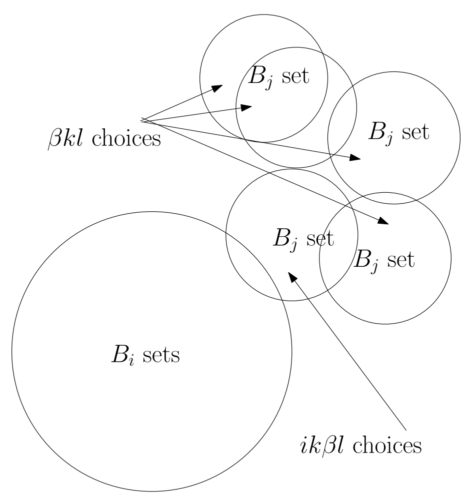

Figure 3: Counting the number of dependent sets

Set $x(B_i) = e^{-c \cdot i \cdot \log k}$ for some large enough c, then for any event i

$$x(B_i) \cdot \prod_j (1 - e^{-cj \log k})^{j \cdot k^{O(i)}} \ge e^{-cilogk} \ge \Pr[B_i]$$

therefore by Lovász Local Lemma there exists a way to reduce k.

Notice that in the new system the size of each set may be greater than k', but this can be handled easily by truncating each set to have only the first k' items. This operation will not affect the analysis for γ because it will only make the new system worse. $\square$

3.4 Explicit construction of Reduce-k

The algorithm by Moser and Tardos does not work for Reduce-k, because the number of bad events is super-polynomial. In a recent paper Haeupler, Saha and Srinivasan [HSS10] gave a new algorithm that works in this situation. Their algorithm relies on the following theorem

Theorem 7. *If the FIX procedure in Moser Tardos algorithm is applied to a subset of the events, and B is an event that FIX has not been applied to, then*

$$\Pr[B \textit{ happens in the output of the algorithm}] \leq \frac{\Pr[B]}{\prod_{C\in\Gamma B}(1-x(C))}$$

Intuitively this theorem says applying FIX to a subset of the events will not change the probability of other events by too much, and the proof follows from the induction in the proof for non-constructive version of LLL in previous lecture.

The next theorem shows that when there's slack in the assumption of LLL, the expected number of calls to FIX is almost linear to the number of variables (does not depend on the number of events)

Theorem 8. *If there exists assignments x such that for some $\epsilon > 0$*

$$\forall A \in \mathcal{A} \quad \Pr[A] \leq (1-\epsilon)x(A) \prod_{B\in\Gamma(A)} (1-x(B)),$$

then $T = \sum_{A\in\mathcal{A}} x(A) = O(n \log(1/\delta))$, where n is the number of variables and $\delta = \min_{A\in\mathcal{A}}\{\Pr[A]\}$. Let F be the number of calls to FIX, then

$$\mathrm{E}[F] = O(\frac{n}{\epsilon} \cdot \log \frac{T}{\epsilon}),$$

and

$$\Pr[F > \lambda \, \mathrm{E}[F]] \leq e^{-O(\lambda)}$$

For the bound on $T = \sum x(A)$, consider each variable v, let A be an event that depends on v. Then since

$$\delta \leq \Pr[A] \leq x(A) \prod_{B\in\Gamma(A)} (1-x(B))$$

The sum of $x(B)$ cannot be much larger than $\log(1/\delta)$, that is $\sum_{B\in\Gamma(A)} x(B) \leq O(log(1/\delta))$. Notice $\Gamma(A)$ contains all events that depend on v, take the sum over all variables v we get the bound of T. The expected number of FIX calls follows from "witness tree" analysis in Moser-Tardos algorithm in the previous lecture, the intuition is since there's a slack of $(1-\epsilon)$, larger trees become exponentially unlikely to happen in the algorithm.

If we only apply FIX to the events whose probability is larger than some inversed polynomial, then by the previous theorem the algorithm will be efficient, we still need to show that this will give a good assignment of the variables with high probability.

Theorem 9. *Suppose* $\log 1/\delta \leq \mathrm{poly}(n)$ *(δ is still the minimum value of probabilities). Suppose further that there is a fixed constant $\epsilon \in (0,1)$ and an assignment x such that*

$$\forall A \in \mathcal{A} \quad \Pr[A]^{1-\epsilon} \leq x(A) \prod_{B\in\Gamma(A)} (1-x(B)),$$

then for any constant c, let $p = n^{-(c+c')/\epsilon}$ (c' is some fixed constant), $\mathcal{A}'$ be the set of all events with probability at least p, running Moser-Tardos algorithm on all events in $\mathcal{A}'$ gives a good assignment with probability at least $1 - n^{-c}$.

Proof. Clearly none of the events in $\mathcal{A}'$ will happen. For event $B \in \mathcal{A}\backslash\mathcal{A}'$, by Theorem 7,

$$\Pr[\text{B happens in the output}] \leq \frac{\Pr[B]}{\prod_{C \in \Gamma(B)}(1 - x(C))},$$

but we also know $\Pr[B] \leq p = n^{-(c+c')/\epsilon}$, therefore $\Pr[B]^{-\epsilon} > n^{c+c'}$

$$\frac{\Pr[B]}{\prod_{C \in \Gamma(B)}(1 - x(C))} \leq x(B)\Pr[B]^{\epsilon} \leq x(B)n^{-c-c'}.$$

By union bound,

$$\begin{aligned}
\Pr[\text{assignment is good}] &\geq 1 - \sum_{B \in \mathcal{A}\backslash\mathcal{A}'} \Pr[\text{B happens in output}] \\
&\geq 1 - \sum_{B \in \mathcal{A}\backslash\mathcal{A}'} x(B)n^{-c-c'} \\
&\geq 1 - \sum_{B \in \mathcal{A}} x(B)n^{-c-c'} \\
&\geq 1 - n\log(1/\delta)x(B)n^{-c-c'} \\
&\geq 1 - n^{-c}.
\end{aligned}$$

(we take c' to be larger than $1 + \log(1/\delta)/\log n$) □

Finally we apply this new algorithm to Reduce-k, clearly the two kinds of slacks are satisfied, and we can enumerate events in $\mathcal{A}$ easily, so there's an algorithm that runs in polynomial time and produce a good assignment with high probability. Since both Reduce-l and Reduce-k are explicit and efficient now, there's an algorithm that produces γ allocation for some fixed $\gamma > 0$ with high probability in polynomial time, and the special case of Santa Claus Problem has a constant approximation algorithm.

References

[AFS08] Arash Asadpour, Uriel Feige, and Amin Saberi. Santa claus meets hypergraph matchings. In *APPROX '08 / RANDOM '08: Proceedings of the 11th international workshop, APPROX 2008, and 12th international workshop, RANDOM 2008 on Approximation, Randomization and Combinatorial Optimization*, pages 10–20, Berlin, Heidelberg, 2008. Springer-Verlag.

[AH00] Ron Aharoni and Penny Haxell. Hall's theorem for hypergraphs. *J. Graph Theory*, 35:83–88, October 2000.

[BS06] Nikhil Bansal and Maxim Sviridenko. The santa claus problem. In *STOC '06: Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 31–40, New York, NY, USA, 2006. ACM.

[Fei08] Uriel Feige. On allocations that maximize fairness. In *SODA '08: Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 287–293, Philadelphia, PA, USA, 2008. Society for Industrial and Applied Mathematics.

[HSS10] Bernhard Haeupler, Barna Saha, and Aravind Srinivasan. New constructive aspects of the lovasz local lemma. *CoRR*, abs/1001.1231, 2010.

[MT09] Robin A. Moser and Gábor Tardos. A constructive proof of the general lovasz local lemma. *CoRR*, abs/0903.0544, 2009.