CS 270 Combinatorial Algorithms and Data Structures, Spring 2015

Lecture 13: Matching polytope

We will use a rank argument (last time we used convex combination) to show that the LP for bipartite matching LP is integral. Then, we will see how the technique can be extended to design approximation algorithms for two related NP-hard problems, the 3-dimensional matching problem and the general assignment problem. Finally, time permitting, we discuss the Edmonds LP for matching in general graphs.

Bipartite matching

Consider the maximum bipartite matching problem (last time it was maximum perfect bipartite matching).

For a subset of edges $F \subseteq E$, we use the notation $x(F)$ for $\sum_{x \in F} x_e$.

So the LP can be written as:

$$\max \sum_{e \in E} w_e \cdot x_e$$

$$x(\delta(v)) \leq 1 \quad \text{for } v \in V$$

$$x_e \geq 0$$

Recall that a basic solution is defined by $|E|$ linearly independent <u>tight</u> constraints.

Notice that there are only $|V|$ degree constraints, so if $|E| > |V|$ there must be edges with $x_e = 0$.

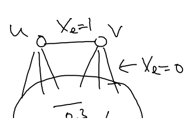
The high level idea is the following: We first argue that if there exist e with $x_e \in \{0,1\}$, then we can reduce the problem into a smaller one. This is intuitively possible because the LP is doing what we want on those edges. So, we iteratively reduce the problem until we could not do so, i.e. $0 < x_e < 1$ for all e in the remaining graph. but then we use the rank argument to derive a contradiction. To conclude, we must find e with $x_e \in \{0,1\}$ and the reduction works all the way.

Reduction

The reduction is based on the <u>induction hypothesis</u> that there is an integral optimal solution for all graphs with fewer edges.

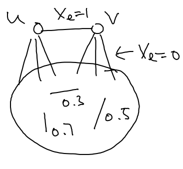
① If $x_e = 0$ for some e, we simply delete this edge e from the graph. It doesn't change the LP value and the graph is smaller, so by induction there is an optimal integral solution for the LP.

② If $x_e = 1$ for some $e = uv$. We remove u and v from the graph and all the incident edges to u, v.



Call the remaining graph $G' = G - \{u, v\}$.

Note that the LP solution restricted to G' is a feasible solution with value



Call the remaining graph $G' = G - \{u,v\}$.

Note that the LP solution restricted to G' is a feasible solution with value

$opt(G) - w_e$, where $opt(G)$ is the optimal value in G.

By induction, there is an integral optimal solution in G' (i.e. an integral matching M') with value $opt(G')-w_e$.

Then, obviously, $M = M' + uv$ is an integral optimal matching in G with value $opt(G)$.

Intuitively speaking, we just follow the LP to choose an edge with $x_e=1$.

Rank argument

Suppose $0 < x_e < 1$ for all $e \in E$. We are going to show that x is not a basic solution, by contradiction.

Recall that there are $|E|$ linearly independent tight constraints for a basic solution x.

Since $x_e > 0$, all the tight constraints are degree constraints.

Call W be the set of vertices with tight degree constraints, i.e. $x(\delta(w)) = 1 \quad \forall w \in W$.

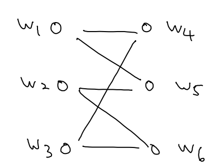
Note that in a basic solution x, we must have $|W| \geq |E|$, as there are $|E|$ linearly independent

For any $w \in W$, we must have $deg(w) \geq 2$ since $x(\delta(w)) = 1$ but $0 < x_e < 1 \quad \forall e \in E$.

So, $|W| \geq |E| = \frac{1}{2}\sum_{v \in V} deg(v) = \frac{1}{2}\left(\sum_{v \in W} deg(v) + \sum_{v \notin W} deg(v)\right) \geq \frac{1}{2}\left(\sum_{v \in W} 2 + \sum_{v \notin W} 0\right) = |W|$.

The inequalities must achieve as equalities.

In particular, we have $deg(v) = 2 \quad \forall v \in W$ and $deg(v) = 0 \quad \forall v \notin W$.



Now, we claim that these tight constraints are not linearly independent,

and hence # of tight $\underset{\wedge \text{independent}}{\text{constraints}}$ $\leq |W| - 1$, while # variables $\geq |W|$,

contradicting that x is a basic solution.

Why the constraints are not linearly independent? Let's do it slowly.

Each constraint is a vector in $\mathbb{R}^{|E|}$.

Given $F \subseteq E$, call χ_F the characteristic vector if $\chi_F(i) = \begin{cases} 1 & \text{if } i \in F \\ 0 & \text{if } i \notin F \end{cases}$.

Note that each constraint in the bipartite matching LP is just $\chi_{\delta(v)}$ for some $v \in V$.

Let W_1 be the subset of W on one side of the bipartite graph and $W_2 \subseteq W$ on the other side.

$deg(v) = 0$ for $v \notin W$, all the edges have one endpoint in W_1 and thus $\sum_{w \in W_1} \chi_{\delta(w)} = \chi_E$, all-one vector. By the same argument, $\sum_{w \in W_2} \chi_{\delta(w)} = \chi_E$.

Therefore, we have $\sum_{w \in W_1} \chi_{\delta(w)} = \sum_{w \in W_2} \chi_{\delta(w)}$, showing that the tight constraints are linearly dependent.

This completes the proof that x is not a basic solution.

Extensions

Okay, so this is an alternate proof that the bipartite matching LP is integral.

It turns out that this argument can be extended to design approximation algorithms for NP-hard variants.

The high level idea is to show that the basic solutions (although not integral) are "close" to integral:

Sometimes it means that there are variables with value close to one, sometimes it means that there are constraints almost satisfied by integral solutions.

We will see an example for each of the above "closeness" to integrality in the following.

LP-rounding

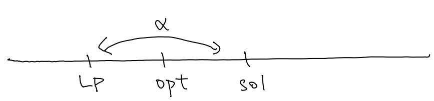
A common approach to design approximation algorithms is to use LP or SDP (a generalization of LP).

For a minimization problem, we say an algorithm is an α-approximation $(\alpha \geq 1)$ if $\frac{\text{sol}}{\text{opt}} \leq \alpha$, where sol denotes the value of the solution our algorithm returns and opt denotes the optimal value.

For a maximization problem, we say an algorithm is an α-approximation $(\alpha \geq 1)$ if $\text{opt}/\text{sol} \leq \alpha$.

The difficulty of designing approximation algorithms is that we can't compute opt efficiently, so it seems difficult to bound the approximation ratio.

Instead, we use LP/SDP to compute a lower bound (for minimization) or an upper bound (for maximization), and if we could show that our integral solution is within a factor of α to the optimal LP/SDP solution, then it is an α-approximation solution.

minimization 

How to prove such a result?

A common strategy is to start from an optimal LP solution, and design a "rounding" algorithm to turn it into an integral solution in a way that we can compare the objective values.

There are many different approaches to design rounding algorithms (see the book by Williamson and Shmoys).

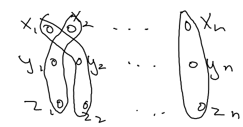
We are going to see one way by analyzing the extreme point solutions for LP.

3-dimensional matching

Let $X=\{x_1,\dots,x_n\}$, $Y=\{y_1,\dots,y_n\}$ and $Z=\{z_1,\dots,z_n\}$ be the 3-dimensions.

We are given m triples (x_i, y_j, z_k), and the task is to find a maximum number of disjoint triples (i.e. no two triples have the same "coordinate").

Equivalently, we can think of it as a maximum matching problem in a 3-uniform tri-partite hypergraph, where the goal is to find a maximum set of disjoint hyperedges.



The following is a simple and natural LP-relaxation of the problem.

$$\max \sum_{e\in E} x_e$$

$$x(\delta(v)) \le 1 \quad \forall v\in V, \text{ where } \delta(v) \text{ is the set of hyperedges containing } v$$

$$x_e \ge 0 \quad \forall e\in E, \text{ where } E \text{ is the set of hyperedges}$$

We are going to show that there is an LP-rounding algorithm with approximation ratio 2.

The idea is to analyze the LP and to show that there is a hyperedge e with $x_e \ge \frac{1}{2}$, in any basic solution.

Again, we can apply the same reduction as in bipartite matching to assume $0 < x_e < 1$ in the following.

<u>Lemma</u> For any basic solution x with $0<x_e<1$ $\forall e\in E$, there exists e with $x_e \ge \frac{1}{2}$.

<u>Proof</u> In a basic solution x, there are $|E|$ linearly independent tight constraints.

Suppose, by contradiction, that all edges have $x_e < \frac{1}{2}$.

Let W be the set of vertices with tight degree constraints, i.e. $x(\delta(v)) = 1$ $\forall v\in W$.

Since $x_e < \frac{1}{2}$ $\forall e$, we must have $\deg(v) \ge 3$ $\forall v\in W$.

We have $|W| \ge |E| = \frac{1}{3}\sum_{v\in V}\deg(v) = \frac{1}{3}\left(\sum_{v\in W}\deg(v) + \sum_{v\notin W}\deg(v)\right) \ge \frac{1}{3}\left(\sum_{v\in W} 3 + \sum_{v\notin W} 0\right) = |W|$.

So, inequalities must hold as equalities, and in particular $\deg(v)=0$ $\forall v\notin W$.

We are going to show that the tight constraints cannot be all linearly independent, and thus the number of linearly independent tight constraints is at most $|W|-1$, while the above argument shows that $|E|\ge|W|$, and this implies that x is not a basic solution.

To see why the tight constraints are linearly dependent, let $W_1 \subseteq W$ be the subset of W in the first part, $W_2 \subseteq W$ in the second part, and $W_3 \subseteq W$ in the third part.

Since $\deg(v)=0$ $\forall v\notin W$, we have $\sum x_e = x_E$ and similarly $\sum x_e = x_E$, in the first part, $W_2 \subseteq W$ in the second part, and $W_3 \subseteq W$ in the third part.

Since $\deg(v) = 0 \ \forall v \notin W$, we have $\sum_{e \in W_1} x_e = x_E$ and similarly $\sum_{e \in W_2} x_e = x_E$,

and thus $\sum_{e \in W_1} x_e = \sum_{e \in W_2} x_e$, showing linear dependence of the constraints. $\square$

Now, we use the lemma to give an LP-rounding algorithm with approximation ratio 2.

The idea is very simple, just keep picking a hyperedge with $x_e \geq \frac{1}{2}$.

More precisely, first we compute a basic solution x.

If $x_e \in \{0,1\}$, we reduce the problem by deleting an edge of $x_e = 0$ or picking an edge with $x_e = 1$ and delete all the hyperedges intersecting with e.

If $x_e \geq \frac{1}{2}$, again we pick e and delete all the intersecting hyperedges.

We get one edge in the integral solution.

How would the LP solution change?

Let $e = (x,y,z)$. Since $x_e \geq \frac{1}{2}$ and $x(\delta(x)) = 1$, the total fractional value of edges intersecting e at x is at most $1 - x_e$. Similarly for y and z.

So, the total fractional value removed is at most $x_e + 3(1-x_e) = 3 - 2x_e \leq 2$ as $x_e \geq \frac{1}{2}$.

Therefore, in the remaining graph, there is an LP solution with objective value opt -2.

By induction, there is an integral solution M' with $\frac{\text{opt}-2}{2} = \frac{\text{opt}}{2} - 1$ in the remaining graph.

So, $M = M' + e$ is an integral solution with value $\frac{\text{opt}}{2} - 1 + 1 = \frac{\text{opt}}{2}$, proving the approximation ratio.

Question: For bipartite matching, last time we saw a simpler "convex combination" argument. Can you apply that argument to 3-dimensional matching and get a simpler proof?

Remark: With an additional idea called "local ratio", one can obtain a 2-approximation algorithm for the weighted problem using this approach. See chapter 9.2 of "iterative methods in combinatorial optimization" if you are interested.

General assignment

In this problem, we'll see the idea of relaxing a "good" constraint if there are no integral variables.

Problem definition: There are m machines $M = \{M_1, M_2, \ldots, M_m\}$ and n jobs $J = \{J_1, \ldots, J_n\}$.

There is a total available time T_i for each machine M_i.

If job j is processed on machine i, the cost is c_{ij} and the processing time is p_{ij}.

Given the above input, the goal is to assign all the jobs to machines, to minimize the total cost while satisfying the time constraints (i.e. total processing time on machine i is at most T_i).

<u>LP relaxation</u>: There is a variable x_{ij} to indicate whether job j is assigned to machine i.

Let E be the set of possible pairs ij; initially every pair is possible, but we will delete pairs.

$$\min \sum_{i\in M, j\in J} c_{ij} x_{ij}$$

$$\sum_{i\in M: ij\in E} x_{ij} = 1 \quad \forall j \in J \quad \text{(each job is assigned to one machine)}$$

$$\sum_{j\in J: ij\in E} x_{ij}\cdot p_{ij} \le T_i \quad \forall i \in M' \subseteq M \quad \text{(total processing time at most } T_i\text{)}$$

$$x_{ij} \geq 0 \qquad \forall ij \in E$$

It is easy to see that this is a relaxation of the problem (i.e. integral solutions are what we want).

<u>Assumption:</u> We assume $p_{ij} \le T_i \;\; \forall ij$, as this pair cannot be in any feasible solution.

<u>Theorem</u>. Suppose there is an assignment with total cost C satisfying all the time constraints. There is a polynomial time algorithm to find an assignment with total cost C while every time constraint constraint is violated by at most T_i (i.e. $\sum_{j:\text{ job assigned to } i} p_{ij} \le 2T_i$).

(One may wonder why we don't insist on the constraints being satisfied while violating the cost by some factor. But finding an assignment satisfying all the constraints (without caring the cost) is NP-hard.)

<u>Iterative algorithm</u> Let $M' = M$. (M' is the set of machines with time constraints; initially every machine.)

While $J \neq \phi$ do

1. Compute a basic optimal solution x. If infeasible, return "impossible".
2. Delete all pairs with $x_{ij} = 0$.
3. If $x_{ij} = 1$, assign job j to machine i. Update $T_i \leftarrow T_i - p_{ij}$ and $J \leftarrow J - \{j\}$.
4. If there is a machine i with degree 1 (i.e. there is only one j with $x_{ij} > 0$), update $M' \leftarrow M' - \{i\}$, that is, remove the time constraint for machine i.

⑤ If there is a machine i with degree 2 and $\sum_j x_{ij} \geq 1$, update $M' \leftarrow M' - \{i\}$.

Return the assignment.

Informally, the algorithm finds integral variables as far as possible: delete a pair if $x_{ij}=0$ and follow the LP to assign job j to machine i if $x_{ij}=1$ and reduce the problem accordingly.

If we could always do the above, then we could solve the problem exactly.

If we couldn't find an integral variable, then we will show that either case ④ or ⑤ must happen.

Note that the machine in case ④ or ⑤ is almost settled with at most two undecided jobs.

The new idea then is to remove the time constraints on those machines, and we are going to show that the violation in these time constraints would be bounded by T_i.

The key of the analysis is to show without ④ or ⑤ then we can always find an integral variable.

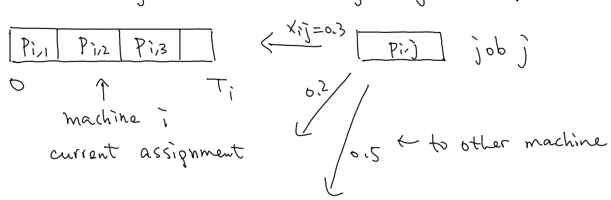
<u>Performance guarantee</u>

Here we assume the algorithm always succeeds in assigning all the jobs to machines (which we will prove later).

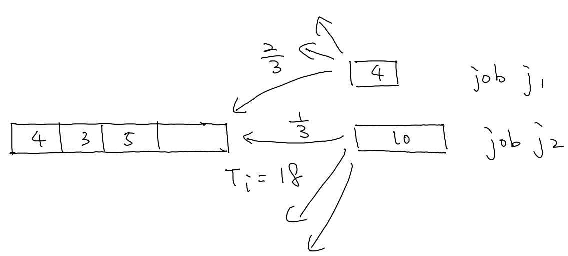
We only assign a job j to machine i when $x_{ij}=1$. By a simple inductive argument as we see before, this shows that the cost of the integral assignment is no more than the LP value.

It remains to consider the violation of the time constraint.

Consider case ④. The machine i was assigned some integral jobs before and is now only left with a fractional job, i.e.



Since machine i is only left with one job, even if we remove the constraint, the worst case is job j will be assigned to machine i in the new LP solution. Since $p_{ij} \leq T_i$ and the current integral assignment at machine i takes at most T_i time units. This implies that the time violation is at most T_i.



Case ⑤ is similar but a little more complicated.

Only two jobs are undecided at machine i,

but we know that $x_{ij_1} + x_{ij_2} \geq 1$ (i.e. machine i is fractionally assigned one job).

We now show that the violation is at most T_i.

The LP uses time $x_{ij_1} p_{ij_1} + x_{ij_2} p_{ij_2}$, while the integral assignment uses at most $p_{ij_1} + p_{ij_2}$.

So, the violation is at most
$$
\begin{aligned}
p_{ij_1} + p_{ij_2} - x_{ij_1} p_{ij_1} - x_{ij_2} p_{ij_2} &= (1-x_{ij_1}) p_{ij_1} + (1-x_{ij_2}) p_{ij_2} \\
&\le (2 - x_{ij_1} - x_{ij_2}) T_i \quad (\text{since } p_{ij} \le T_i) \\
&\le T_i \qquad (\text{since } x_{ij_1} + x_{ij_2} \ge 1).
\end{aligned}
$$

Therefore, assuming the algorithm terminates, it produces an assignment with cost at most the LP value, while the time constraint on each machine is violated by at most T_i.

Properties of basic solutions

To complete the proof, it remains to show that the algorithm must terminate successfully, i.e. in a basic feasible solution, one of ②, ③, ④, ⑤ must apply.

Suppose ② and ③ don't apply. Then there are $|E|$ linearly independent tight constraints, among the job constraints and the machine constraints.

Call the tight job constraints J^* and the tight machine constraints M^*.

Recall that $|J^*| + |M^*| \ge |E|$ in a basic solution.

Since $0 < x_e < 1 \ \forall e \in E$ (as ② and ③ don't apply), the degree of each job $j \in J^*$ has degree ≥ 2.

For a machine $i \in M^*$, if degree $= 1$, then ④ applies. So, assuming ④ doesn't apply, then each machine has degree at least two.

So,
$$
\begin{aligned}
|J^*| + |M^*| \ge |E| = \frac{1}{2}\Big(\sum_{j \in J} \deg(j) + \sum_{i \in M} \deg(i) \Big) &= \frac{1}{2}\Big(\sum_{j \in J^*} \deg(j) + \sum_{j \notin J^*} \deg(j) + \sum_{i \in M^*} \deg(i) + \sum_{i \notin M^*} \deg(i) \Big) \\
&\ge \frac{1}{2}\Big(\sum_{j \in J^*} 2 + \sum_{j \notin J^*} 0 + \sum_{i \in M^*} 2 + \sum_{i \notin M^*} 0 \Big) \\
&= |J^*| + |M^*|.
\end{aligned}
$$

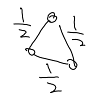
So, equalities hold throughout, and in particular $\deg(j) = 0 \ \forall j \notin J^*$ and $\deg(i) = 0 \ \forall i \notin M^*$, and furthermore, the remaining graph is a disjoint union of cycles (since $\deg = 2$).

Consider one such cycle C. Then $|J^* \cap C| = |M^* \cap C|$.

As each job $j \in J^* \cap C$ has $\sum_i x_{ij} = 1$ by the job constraint, the total fractional value in the cycle is at least $|J^* \cap C| = |M^* \cap C|$, and this implies that there must exist a machine i with $\sum_j x_{ij} \approx 1$ and $\deg(i) = 2$, hence case (b) must apply, and we're done.

To conclude, the algorithm must terminate successfully, and when it does, the performance guarantee is what we claimed.

Matching in general graphs

As you may observed by now, the LP for bipartite matching is not integral for general graphs. Of course, it has to do with an odd cycle, e.g. 

, while integral opt = 1.

Edmonds gave an exponential size LP for general matching.

$$\max \sum_e w_e x_e$$

$$x(\delta(v)) \le 1 \quad \forall v \in V$$

$$x(E(S)) \le \frac{|S|-1}{2} \quad \forall S \subseteq V$$ where $|S|$ odd and $E(S)$ is the set of edges with both vertices in S.

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

It is easy to see that the new constraints are valid, as an odd set S cannot contain more than $\frac{|S|-1}{2}$ edges.

It turns out that this LP is exact and also it can be solved in polynomial time (as there is a polytime separation oracle). Both are difficult to prove and we won't do it.

See chapter 9 of "iterative methods in combinatorial optimization" if you are interested.

You may wonder, as many researchers did, whether there is a polynomial size LP for general matching.

Recently, Rothvoss proved an exponential lower bound for the LP size for general matching in a very general setting (including all the possible LPs that we could imagine).

So, surprisingly, it has to be so complicated for general matching!

References

Materials are from chapter 3 and 9 of "iterative methods in Combinatorial optimization".

The general assignment's result is due to Shmoys and Tardos.