15-854: Approximations Algorithms **Lecturer:** Anupam Gupta
Topic: LP Rounding, continued **Date:** November 2, 2005
Scribe: Brian Taylor

16.1 Introduction

Today we continue our discussion of LP rounding. Last time, we applied the filtering and rounding techniques of Lin and Vitter[2, 3] to the metric uncapacitated facility location problem to obtain a 4-approximation[1]. In today's lecture, we look at three problems with similar ILP formulations:

- Minimum Cost Bipartite Matching
- Minimum Makespan Scheduling on Unrelated Parallel Machines
- Generalized Assignment

For each of these problems, the basic feasible solutions of the corresponding LP relaxation can be shown to have some convenient properties. These properties can be used to directly round optimal LP solutions into feasible integral solutions within a constant factor of optimum, with no need of an intermediate filtering phase.

16.2 Minimum Cost Bipartite Matching

Recall that a *matching* on a bipartite graph $G = (U, V, E)$ is a subset $M \subseteq E$ of edges such that no two edges in M are incident to the same vertex. In this problem, we allow partite sets U and V to differ in size, with $|U| = n_U \leq |V| = n_V$, and we define a cost c_{ij} on every edge (u_i, v_j). We seek a matching of minimum cost in which every vertex in U, the smaller of the partite sets, must be adjacent to an edge in the matching. (When $|U| = |V|$, this is a min-cost perfect matching.)

The min-cost bipartite matching problem can be formulated as an integer program as follows:

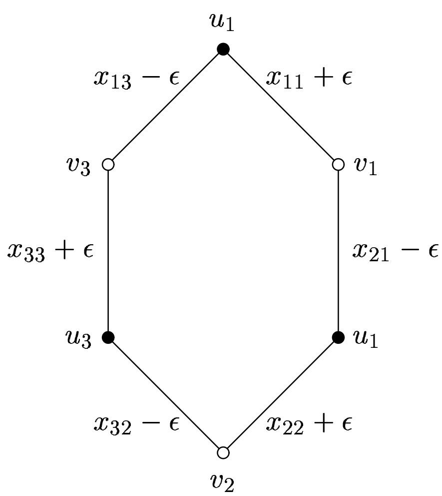
$$
\begin{aligned}
\min \quad & \sum_{(u_i,v_j)\in E} c_{ij} x_{ij} \\
\text{s.t.} \quad & \sum_{j=1}^{n_V} x_{ij} = 1 \qquad \forall\, i \in \{1,\ldots,n_U\} \quad (1)\\
& \sum_{i=1}^{n_U} x_{ij} \leq 1 \qquad \forall\, j \in \{1,\ldots,n_V\} \quad (2)\\
& x_{ij} \in \{0,1\}
\end{aligned}
$$

Here x_{ij} is 1 if and only if edge (u_i, v_j) is in the minimum cost matching. An LP relaxation can be obtained by replacing the integer constraint $x_{ij} \in \{0, 1\}$ with the constraint $x_{ij} \geq 0$. We will show that basic feasible solutions of this LP relaxation are integral, and hence an optimum basic feasible solution of the LP relaxation is optimal for the integer program.

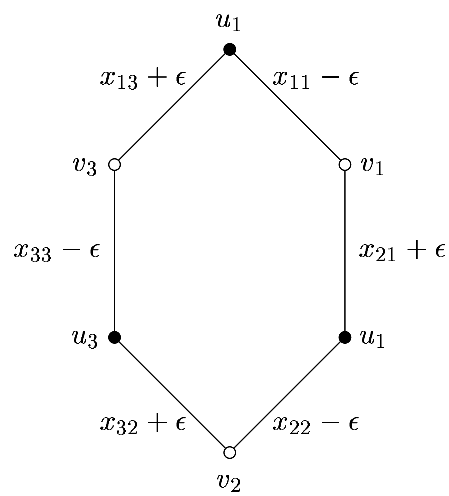
Lemma 16.2.1 *Let x be a basic feasible solution of the LP relaxation. There are no cycles among the edges $M = \{(u_i, v_j) : x_{ij} > 0\}$.*

Proof: Suppose x is a basic feasible solution of the LP relaxation, and there is a cycle among the set M of edges (u_i, v_j) with $x_{ij} > 0$. We will show that x can be written as an affine combination of two other solutions y and z, which contradicts the assumption that x is a basic feasible solution[1].

Let $C = \{u_1, v_1, u_2, v_2, \ldots, u_\ell, v_\ell\}$ be the cycle among the edges of M. Since G is bipartite, C is of even order. Define solution y as follows: for all edges $(u_{i'}, v_{j'})$ not on the cycle, let $y_{i'j'} = x_{i'j'}$. For the first edge (u_1, v_1) on the cycle, let $y_{11} = x_{11} + \epsilon$; for the second edge (u_2, v_1), let $y_{21} = x_{21} - \epsilon$. For the third edge, fifth edge, seventh edge, and so on, let $y_{kk} = x_{kk} + \epsilon$, and for the fourth edge, sixth edge, etc., let $y_{k(k-1)} = x_{k(k-1)} - \epsilon$. Define z similarly, but with the sign of ϵ reversed: $z_{kk} = x_{kk} - \epsilon$, and $z_{k(k-1)} = x_{k(k-1)} + \epsilon$. See below for an illustration of solutions y and z.



Definition of y_{ij}



Definition of z_{ij}

Let ϵ be the largest value such that $0 \le y_{ij} \le 1$ and $0 \le z_{ij} \le 1$ for all i and j. Note that y and z are feasible solutions: since $\sum x_{ij}$ is "conserved" at each u_i and v_j by adding and subtracting ϵ, solutions y and z satisfy constraints 1 and 2. Since $x = \frac{1}{2}y + (1 - \frac{1}{2})z$, it follows that x is not a basic feasible solution, contradicting our assumptions. ■

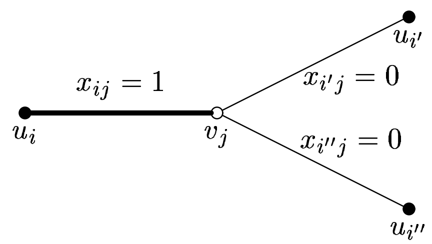
Observe that by our choice of ϵ, some y_{ij} or z_{ij} along the cycle must be zero. Effectively, the cycle in x is "broken" in either y or z, so we can think of the process of adding and subtracting ϵ as "cycle cancellation." The next lemma makes use of a similar "path cancellation" idea to argue that all paths in M are of length 1 (and hence, M is a matching).

[1]Recall that a solution is a basic feasible solution if and only if it is an extreme point of the polytope defined by the linear constraints. If solution x can be written as an affine combination of solutions y and z, that is, $x = \alpha y + (1-\alpha)z$ for some $\alpha \in (0, 1)$, then x lies along the line segment between y and z, and hence cannot be an extreme point.

Since M has no cycles, it is a forest.

Claim 16.2.2 *The leaves of all non-K_2 trees on the edges of M (i.e., those trees on M which are not a single edge) are in V, the larger partition.*

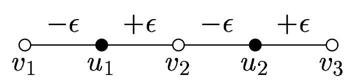
Proof: Suppose vertex u_i in U is a leaf of a non-K_2 tree in M. Let v_j be the neighbor of u_i in M. By constraint 1, $x_{ij} = 1$, which implies that $x_{i'j} = 0$ for all $u_{i'}$ adjacent to v_j, where $u_{i'} \neq u_i$.



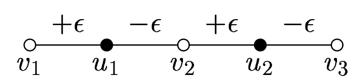
Thus, the tree in M of which u_i is a leaf is a single edge, which contradicts our assumptions. ■

Lemma 16.2.3 *Let x be a basic feasible solution of the LP relaxation. Then the edges $M = \{(u_i, v_j) \ : x_{ij} > 0\}$ form a matching.*

Proof: By Lemma 16.2.1, M is a forest. Consider a leaf-to-leaf path $P = \{v_1, u_1, \ldots, v_{\ell-1}, u_{\ell-1}, v_\ell\}$ in some non-K_2 tree in M. By Claim 16.2.2, both endpoints of P are in V. As in the proof of Lemma 16.2.1, we form two new solutions y and z. For all edges (u_i', v_j') not in P, set $y_{i'j'} = x_{i'j'}$. For the first, third, fifth, etc., edge (u_k, v_k) along the path, set $y_{kk} = x_{kk} - \epsilon$, and for all other edges in P, set $y_{(k-1)k} = x_{(k-1)k} + \epsilon$. Define z_{ij} similarly, but with the opposite sign on ϵ.



Values added to x_{ij} to get y_{ij}



Values added to x_{ij} to get z_{ij}

If we choose the largest ϵ such that $0 \leq y_{ij} \leq 1$ and $0 \leq z_{ij} \leq 1$ for all i and j, then y and z are feasible solutions, and $x = \frac{1}{2}y + (1 - \frac{1}{2})z$, which contradicts the assumption that x is a basic feasible solution. Thus, all trees in M must consist of a single edge, and hence M is a matching. ■

The preceding lemma and constraint 1 imply that in any basic feasible solution, if $x_{ij} > 0$, then $x_{ij} = 1$. Thus, we have the following theorem.

Theorem 16.2.4 *Any basic feasible solution of the LP relaxation is integral.*

Because the optimum $c^T x$ of the LP relaxation is a lower bound on the optimum of the integer program, and because x is integral, an optimal solution for the LP relaxation[2] is a feasible (and thus optimal) solution for the integer program as well.

[2] By the fundamental theorem of linear programming, we may assume without loss of generality that an optimal solution is a basic feasible solution.

16.3 Scheduling on Unrelated Parallel Machines

In the scheduling problem for unrelated parallel machines (denoted $R \,||\, C_{\max}$), we wish to schedule a set of n jobs $J = \{J_1, \ldots, J_n\}$ on a set of m machines $M = \{M_1, \ldots, M_m\}$. The processing time of job J_j on machine M_i is *unrelated* to its processing time on any other machine $M_{i'}$. Thus, for every machine M_i and job J_j, we denote by p_{ij} the processing time of J_j on M_i. We require that each job must be processed on one machine alone; it cannot be started on one machine, preempted, and completed on another machine. Our objective is to construct a schedule of minimum makespan, where the makespan of a schedule is the maximum completion time of any job. This problem can be formulated as the following ILP:

$$
\begin{aligned}
\min \quad & t \\
\text{s.t.} \quad & \sum_{j=1}^{n} p_{ij} x_{ij} \leq t \quad \forall\, i \in \{1, \ldots, m\} \quad (1) \\
& \sum_{i=1}^{m} x_{ij} = 1 \qquad \forall\, j \in \{1, \ldots, n\} \quad (2) \\
& x_{ij} \in \{0, 1\}
\end{aligned}
$$

Here $x_{ij} = 1$ if and only if job J_j is assigned to machine M_i. We can form an LP relaxation by replacing the constraint $x_{ij} \in \{0,1\}$ with the constraint $x_{ij} \geq 0$. Unfortunately, the integrality gap of this LP is not bounded by any constant.

Claim 16.3.1 *The integrality gap of the above LP relaxation is m.*

Proof: Suppose there is one job and m machines, with $p_{i1} = m$ for all machines M_i. The optimal fractional solution is $x_{i1} = 1/m$ for all i, which yields a fractional makespan $t = 1$. However, any ILP solution has a makespan of m. ■

To get around this problem, we will use the method of parametric pruning. We'll guess a makespan bound T, rather than attempt to directly minimize the makespan. If our guess T is too low, the LP will have no solution. By binary searching over a suitable set of values (to be discussed later), we can find the minimum value of T. Our revised LP, which we denote as LP_{BM}, has the following constraints:

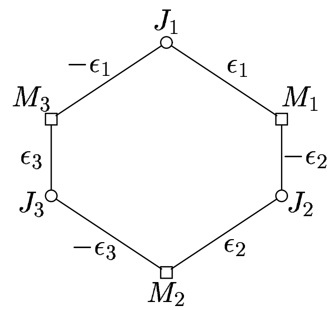
$$
\begin{aligned}
& \sum_{j=1}^{n} p_{ij} x_{ij} \leq T \quad \forall\, i \in \{1, \ldots, m\} \quad (1) \\
& \sum_{i=1}^{m} x_{ij} = 1 \qquad \forall\, j \in \{1, \ldots, n\} \quad (2) \\
& x_{ij} \geq 0
\end{aligned}
$$

Additionally, we will add a constraint to filter out processing times which are greater than the makespan T. Let $E_T = \{(i,j) : p_{ij} \leq T\}$. We require $x_{ij} = 0$ for all $(i,j) \notin E_T$.

It will be convenient in the following discussion to visualize a fractional solution x of LP_{BM} as a bipartite graph $G(x) = (J, M, E)$. Partite sets J and M represent the jobs and machines, respectively, and edge set $E = \{(M_i, J_j) : x_{ij} > 0\}$. We will show that given a fractional solution x to $\mathrm{LP}_{\mathrm{BM}}$, we can obtain an integral solution $\hat{x}$ with makespan $C_{\max} \leq 2T$.

Lemma 16.3.2 *Let x^* be an optimal solution of $\mathrm{LP}_{\mathrm{BM}}$. We can obtain from x^* a fractional solution x such that $G(x)$ is acyclic.*

Proof: If $G(x^*)$ is acyclic, we are done. Otherwise, let $C = \{J_1, M_1, \ldots, J_\ell, M_\ell\}$ be a cycle in $G(x^*)$. Form x from x^* in the following way: if $(M_{i'}, J_{j'})$ is not in C, let $x_{i'j'} = x^*_{i'j'}$. For each job J_j on the cycle, we assign $+\epsilon_j$ and $-\epsilon_j$ to the two cycle edges incident to J_j. For every (M_i, J_j) on the cycle, we set x_{ij} equal to x^*_{ij} plus the epsilon value on that edge.



ϵ values added to x^*_{ij} to get x_{ij}

Clearly, the solution x satisfies constraint 2. To ensure that it also satisfies constraint 1, we will enforce $\ell - 1$ linear constraints on the ϵ values. At each machine M_i for $i \in \{2, 3, \ldots, \ell\}$, we force the "load" $\sum_{j=1}^{n} p_{ij} x_{ij}$ on M_i to be unchanged. If the neighbors of M_i on the cycle are jobs J_i and J_{i+1}, and the corresponding epsilon values are ϵ_i and $-\epsilon_{i+1}$, then we can ensure that the load on M_i is unchanged by setting

$$p_{ii}\epsilon_i = p_{i(i+1)}\epsilon_{i+1}$$

for each $i \in \{2, 3, \ldots, \ell\}$. For the above example, this gives us the following system of equations:

$$p_{22}\epsilon_2 = p_{23}\epsilon_3$$
$$p_{33}\epsilon_3 = p_{31}\epsilon_1$$

Since we have ℓ variables and $\ell - 1$ equations, we can treat one of the epsilon variables (say ϵ_1) as a free variable. Starting with $\epsilon_1 = 0$, increase ϵ_1 until some x_{ij} hits 0. We have guaranteed by the above equations that the load at machines $M_2, M_3, \ldots, M_\ell$ is unchanged. There are two possibilities for the load L of machine M_1:

1. L hasn't increased. In this case, x is a feasible solution.

2. L has increased. Simply flip the sign on each ϵ_i to obtain a feasible solution.

Thus, we have a feasible solution x with one fewer cycle than x^*. We can iteratively apply the above procedure to x until x is acyclic. ■

Using an argument similar to the proof of Lemma 16.2.1, one can show the following result.

Exercise 16.3.3 *Suppose we augment* $\mathrm{LP_{BM}}$ *with an objective which minimizes the sum of the loads on all the machines; that is, we minimize* $\sum\sum p_{ij}x_{ij}$*. Show that any basic feasible solution of this new LP is acyclic.*

Lemma 16.3.4 *Given a feasible solution* x *such that* $G(x)$ *is acyclic, we can obtain an integral solution* $\hat{x}$ *corresponding to a schedule of makespan* $C_{\max} \leq 2T$*.*

Proof: For all $x_{ij} = 1$, assign job J_j to machine M_i (and delete J_j from $G(x)$). The resulting partial schedule S has makespan at most T. $G(x)$ is a forest; by reasoning similar to the proof of Claim 16.2.2, it's not hard to see that every leaf in $G(x)$ must be a machine. Root each tree in $G(x)$ at an arbitrary job, and assign each job J_j to one of its children machines M_i. For each such assignment, since the edge (M_i, J_j) is in $G(x)$, we know that $p_{ij} \leq T$. Thus, each machine from the partial schedule S gets at most one more job whose processing time is at most T, so we have a schedule of makespan $C_{\max} \leq 2T$. ■

By the above lemmas, we know that, given a lower bound T on the optimal makespan, we can construct a schedule of makespan no more than $2T$. We now show how to find a suitable value of T.

Claim 16.3.5 *To find the appropriate value of* T*, we need only search over the values* p_{ij}*.*

Proof: Let $p_1, \ldots, p_{nm}$ be the processing times in sorted order. Suppose $p_i < T < p_{i+1}$. If we set $T' = p_i$, the set of p_{ij} such that $p_{ij} \leq T'$ is equal to the set of $p_{ij} \leq T$, and the fractional makespan $\sum\sum p_{ij}x_{ij}$ of the new program $\mathrm{LP}(T')$ is no larger than that of $\mathrm{LP}(T)$. ■

The following theorem is immediate from Lemma 16.3.2, Lemma 16.3.4, and Claim 16.3.5.

Theorem 16.3.6 *There is a 2-approximation algorithm for the problem of minimum makespan scheduling on unrelated parallel machines.*

16.4 Generalized Assignment Problem

The problem of minimum makespan scheduling on unrelated parallel machines can be generalized by associating a cost c_{ij} with each job J_j and machine M_i, where c_{ij} is the cost of assigning J_j to M_i. The costs c_{ij} are not related to the processing times p_{ij}. In addition to finding a schedule of minimum makespan, we wish to minimize assignment costs as well. Given values C and T, Shmoys and Tardos[4] show how to find a schedule of cost at most C and makespan at most $2T$, assuming that there exists a schedule of cost at most C and makespan at most T.

16.5 Notes

We saw in Section 16.2 that the vertices of min-cost bipartite matching polytopes are integral. It turns out[5] that the vertices of general matching polytopes are half-integral (each variable is set to either 0, 1/2, or 1).

A 2-approximation for $R \,||\, C_{\max}$ was given by Lenstra, Shmoys, and Tardos[6]. In that paper, it is shown that approximating $R \,||\, C_{\max}$ within a factor of 3/2 is NP-hard. It is also shown that if all $p_{ij} \in \{1, 2\}$, then the problem can be solved in polynomial time. When the number of machines is fixed, Jansen and Porkolab[7] give a fully polynomial time approximation scheme whose time complexity is linear in n, the number of jobs.

References

[1] David B. Shmoys, Éva Tardos, and Karen Aardal. *Approximation algorithms for facility location problems.* In 29th Annual ACM Symposium on Theory of Computing, pages 265–274, 1997.

[2] Jyh-Han Lin and Jeffrey S. Vitter. *ϵ-approximations with minimum packing constraint violation.* In 24th Annual ACM Symposium on Theory of Computing, pages 771–782, 1992.

[3] Jyh-Han Lin and Jeffrey S. Vitter. *Approximation Algorithms for Geometric Median Problems.* Information Processing Letters, Vol. 44, pages 245–249, 1992.

[4] David B. Shmoys and Éva Tardos. *Scheduling unrelated machines with costs.* Proceedings of the fourth annual ACM-SIAM Symposium on Discrete Algorithms, pages 448–454, 1993.

[5] L. Lovász and M. D. Plummer. *Matching Theory.* North-Holland Publishing Co., Amsterdam, 1986. Annals of Discrete Mathematics, Vol. 29.

[6] Jan Lenstra, David B. Shmoys, and Éva Tardos. *Approximation algorithms for scheduling unrelated parallel machines.* Proceedings of the fourth annual ACM-SIAM Symposium on Discrete Algorithms, pages 448–454, 1993.

[7] Klaus Jansen and Lorant Porkolab. *Improved Approximation schemes for scheduling unrelated parallel machines.* Proceedings of the thirty-first annual ACM symposium on Theory of computing, pages 408–417, 1999.