Approximation Algorithms via LP rounding

Recipe: 1. Encode problem as integer LP
2. Relax to fractional LP and use an LP solver to obtain an
   optimal fractional solution $x^*$
3. ROUND fractional solution to an integer solution $x_{\text{INT}}^*$

$$\min \ c^T x$$
$$A x \geq b \quad \Leftrightarrow \quad x^* \leq x_{\text{opt}} \quad \text{Optimal} \quad \Rightarrow \quad x_{\text{INT}}^* \text{ is an}$$
$$x \geq 0 \quad \Rightarrow \quad x_{\text{INT}}^* \leq x^* \quad x_{\text{INT}}^* \text{ is an} \alpha-\text{approximation}$$

Examples

1. Vertex cover

$$\min \ \sum_{v \in V} x(v)$$
$$\text{s.t.} \quad \forall (u,v) \in E \quad x(u) + x(v) \geq 1$$
$$x(v) \geq 0 \quad \forall v \in V$$

$$x_{\text{INT}}(v) = \begin{cases} 1 & \text{if } x^*(v) > \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}$$

Approximation factor = 2

2. Set cover

$$\min \ \sum_{S \in S} c(S) x(S)$$
$$\sum_{S \in S} x(S) \geq 1 \quad \forall S \in S$$
$$x(S) \geq 0 \quad \forall S \in S$$

$$x_{\text{INT}}(S) = \begin{cases} 0 & \text{if } x^*(S) > \frac{1}{2} \\ 1 & \text{otherwise} \end{cases}$$

$$\text{Approximation factor} = \frac{1}{\text{OPT}}$$

Problem: $\Pr \left[ \sum_{S \in S} x(S) \geq 1 \right] \geq 1 - \frac{1}{e} \left(1 - x(S)\right) > 1 - \frac{1}{e}$

i.e. in expectation, a constant fraction of elements are not covered by $x_{\text{INT}}^*$

Solution: Repeat $\log n$ times (or boost rounding probability by $\log n$).
Monte Carlo algorithm with approximate log.n.

How do we convert to Las Vegas? For each element that is not covered, add cheapest set covering the element.

Load balancing on Unrelated Machines [Lenstra-Shmoys-Tardos]

\[
\begin{align*}
\min \ t \\
\text{s.t.} \ \sum_{j \in J} x_{ij} p_{ij} &\leq t + \epsilon \ i \in M \\
\sum_{i \in M} x_{ij} &\geq 1 \ \forall j \in J \\
x_{ij} &\geq 0 \\
t^* = 1; \ \text{Integer OPT} = m
\end{align*}
\]

Define: The integrality gap of an LP is the worst-case ratio (over all instances) of the integer optimal to the fractional optimal objective.

The above example shows that this LP has an integrality gap of \(m\).

Stronger LP: Let \(t_{\text{err}}\) be optimal makespan (can be guessed up to \((1+\epsilon)\) error). Define

\[
\tilde{p}_{ij} = \begin{cases} 
  p_{ij} & \text{if } p_{ij} \leq t_{\text{err}} \\
  \infty & \text{if } p_{ij} > t_{\text{err}} 
\end{cases}
\]

New LP: Check feasibility of

\[
\sum_{j \in J} \tilde{x}_{ij} \geq 1 \ \forall i \in M \ \text{guaranteed for } t_{\text{err}}
\]

\[
\sum_{i \in M} \sum_{j \in J} \tilde{p}_{ij} x_{ij} \leq t_{\text{err}} \ \forall i \in M
\]

If infeasible, update \(t_{\text{err}}\) to \((1+\epsilon)\) \(t_{\text{err}}\).

If feasible, we need to round the solution.

Rounding: In a basic feasible solution, at most \((m+n)\) variables.
Rounding: In a basic feasible solution, at most \((m+n)\) variables \(x_{ij}\) are non-zero. If \(n > m\), at least \(n-m\) jobs are assigned integrally. Let \(H\) be the graph of edges with \(0 < x_{ij} < 1\). This is an \(n'\times m\) graph containing \(\leq n'+m\) edges, but each job has degree \(\geq 2\) and \(n' \leq m\).

Lemma: \(H\) is a pseudo-tree, i.e., each connected component on \(k\) vertices has at most \(k\) edges.

Proof: \(x^*\), restricted to the jobs and machines in a connected component, must be a BFS. If not, it can be written as a convex combination of feasible points which can be extended to the full LP.

Lemma: A pseudo-tree in a bipartite graph has a perfect matching.

Proof: Leaves are machines since each job has degree \(\geq 2\). Match leaves to their unique neighbors and recurse.
If there are no leaves, then the pseudo-tree is a cycle, which being bipartite, must be even and hence contain a perfect matching. Thus, each machine gets at most one job more than those it get integrally in the fractional solution. By parametric pruning, this is a 2-approximation.