Approximation Algorithms:
LP Relaxation, Rounding, and Randomized Rounding Techniques

My T. Thai

Overview

An overview of LP relaxation and rounding method is as follows:

1. Formulate an optimization problem as an *integer program* (IP)
2. Relax the integral constraints to turn the IP to an LP
3. Solve LP to obtain an optimal solution x^*.
4. Construct a feasible solution x^I to IP by rounding x^* to integers.

Rounding can be done deterministically or probabilistically (called *randomized rounding.*

Let us define some notations:

- Let $cost(x^I)$ and $cost(x^*)$ be the objective values of x^I and x^* respectively
- Let $OPT(IP)$ and $OPT(LP)$ be the optimal values of the IP and LP respectively. Note that $OPT(IP)$ is also a cost of an optimal solution for the optimization problem whereas the $OPT(LP) = cost(x^*)$.

Hence the approximation ratio using the rounding techniques can be done as follows (suppose the optimization is a minimization problem):

$$cost(x^I) = \frac{cost(x^I)}{OPT(LP)} \times OPT(LP) \leq \frac{cost(x^I)}{cost(x^*)} \times OPT(IP)$$

Therefore, any upper bound of $\frac{cost(x^I)}{cost(x^*)}$ is an approximation ratio of the construction algorithm (which is to round x^* to x^I).

If we follows this method as mentioned about, the best approximation ratio is $\frac{cost(x^I)}{cost(x^*)} \leq \frac{OPT(IP)}{OPT(LP)}$

Note that the supremum of $\frac{OPT(IP)}{OPT(LP)}$ is called *integrality gap.*

Linear Integer Program and Examples

Roughly speaking, linear integer program is similar to an LP with an additional constraint, that is, variables are integer. The integer program is the problem of determining if a given integer program has a feasible solution.

Let's see how we can use IP to formulate many discrete optimization problems.

VERTEX COVER (VC)

Recall that the vertex cover is defined as follows:

Definition 1 *Given a graph $G = (V, E)$ where $|V| = n$ and $|E| = m$, find a subset $C \subseteq V$ such that every edge $e \in E$ has at least one endpoint in C and C has a minimum size.*

How to formulate VC as an IP?

For each vertex $i \in V$ ($V = \{1, 2, ..., n\}$, let $x_i \in \{0, 1\}$ be variables such that $x_i = 1$ if $i \in C$; otherwise, $x_i = 0$. We have:

$$\begin{array}{lll} \min & \sum_{i=1}^{n} x_i & \\ \text{st} & x_i + x_j \geq 1 & \forall (i, j) \in E \\ & x_i \in \{0, 1\} & \forall i \in V \end{array} \tag{1}$$

Hence, solving the VC problem is equivalent to solving the above IP.

Exercise 1 *Formulate the weighted vertex cover problem as an IP.*

WEIGHTED SET COVER

Definition 2 *Given a universe $U = \{1, ..., m\}$, a collection $\mathcal{S}$ of subsets of U, $\mathcal{S} = \{S_1, ..., S_n\}$, and a weight function $w : \mathcal{S} \to \mathbb{Q}^+$, find a minimum weight sub-collection $\mathcal{C} = \{S_j \mid 1 \leq j \leq n\}$ such that $\mathcal{C}$ covers all elements of U.*

Let w_j be the weight of subset S_j. Let x_j be a binary variables such that $x_j = 1$ if $S_j \in \mathcal{C}$; otherwise, $x_j = 0$. We have the corresponding IP:

$$\begin{array}{lll} \min & \sum_{j=1}^{n} w_j x_j & \\ \text{st} & \sum_{i \in S_j} x_j \geq 1 & \forall i \in \{1, ..., m\} \\ & x_j \in \{0, 1\} & \forall j \in \{1, ..., n\} \end{array} \tag{2}$$

LP Relaxation and Rounding

VERTEX COVER

The corresponding LP of IP (1) is as follows:

$$\begin{array}{lll} \min & \sum_{i=1}^{n} x_i & \\ \text{st} & x_i + x_j \geq 1 & \forall (i,j) \in E \\ & 0 \leq x_i \leq 1 & \forall i \in V \end{array} \tag{3}$$

Note that we can relax the last constraint $0 \leq x_i \leq 1$ to $0 \leq x_i$ without changing the solutions of LP (3).

A main observation of this LP is that:

Theorem 1 *Given a graph G, any vertex x of the polyhedron defined by (3) has coordinates x_i being either 0, 1/2, or 1. (This property is called* half integrality*)*

(Note that any vertex x of the polyhedron is called a basic feasible solution (or extreme point solution). Also note that a vertex of the polyhedron cannot be expressed as a convex combination of two feasible solutions.)

Based on this theorem, we can construct a feasible solution x^I to the IP as follows:

Let $x_i^I = 1$ if $x_i^* \geq 1/2$; otherwise, $x_i^I = 0$.

This technique of constructing a feasible to IP from LP is called rounding. Note that after the relaxation and rounding techniques, x^I is a feasible solution to IP.

What is the approximation ratio? Recall that the approximation ratio is equal to $cost(x^I)/cost(x^*)$. We have:

$cost(x^*) \geq \frac{1}{2}(x_1^I + ... + x_n^I) = \frac{1}{2}cost(x^I)$.

Hence, this algorithm has an approximation ratio of 2. Recall that by using the matching approach, we also obtained a 2-approximation algorithm.

Question: Can we use this technique to obtain a 2-approximation for weighted vertex cover problem?

LP Relaxation and Rounding

WEIGHTED SET COVER

First, let obtain an LP from IP (2) as follows:

$$\begin{array}{lll} \min & \sum_{j=1}^{n} w_j x_j & \\ \text{st} & \sum_{i \in S_j} x_j \geq 1 & \forall i \in \{1, ..., m\} \\ & 0 \leq x_j \leq 1 & \forall j \in \{1, ..., n\} \end{array} \tag{4}$$

How do we obtain x^I from x^*?

Note that when we round x^* to x^I, we must make sure that x^I is a feasible solution to IP. In addition, we must also not "over round" it, i.e, round all fraction solutions x^* to 1. Of course, the over rounding still yield a feasible solution to IP, however, we will obtain too many subsets in $\mathcal{C}$.

Let rewrite (4) as min $c^T x$ subject to $Ax \geq b$. Now, consider matrix A. What can we say about each entry a_{ij}. a_{ij} is equal to either 0 or 1. $a_{ij} = 1$ if element $i \in S_j$; otherwise, $a_{ij} = 0$. Denote $f = max_i \sum A[i, :]$. So f represent the frequency of the most frequent element. In other words, f denotes the maximum number of subset S_j that cover an element.

Why we want to define f? Notice that during the rounding process, we round some x_j^* to 1 and the rest to 0. The rounding must satisfy the inequalities and consistencies. That is, for example, if in the first inequality, we round x_1^* to 1 and the rest to 0. Then in the second inequality, we must also round x_1^* to 1.

Let consider this example. We have the first inequality as follows: $x_1^* + 0x_2^* + x_3^* + 0x_4^* + x_5^* \geq 1$. The sum of all coefficients of the x_j^* is 3. If all x_j^* were at most 1/3, then the left hand side is at most 1. Thus there must be some x_j^* which are at least 1/3. If $x_3^* \geq 1/3$, we round x_3^* to 1 then we'd be fine. Now, you might ask what if only $x_2^*, x_4^* \geq 1/3$. The answer is that: this cannot be happened. Why?

Therefore, we have the following rounding algorithm:

Let $x_j^I = 1$ if $x_j^* \geq 1/f$; otherwise, $x_j^I = 0$.

Note that when $x_j^I = 1$, then $S_j \in \mathcal{C}$.

Theorem 2 *This algorithm achieves an approximation factor of f for the weighted set cover problem*

Proof. First, we need to solve that x^I is a feasible solution to weighted set cover, that is, $\mathcal{C}$ covers all elements.

Assume that there exists one element i that is not covered by $\mathcal{C}$. Then there exists a row i such that:

$$\sum_{j: x_j^* \geq 1/f} a_{ij} = 0$$

However,

$$\sum_{j=1}^{n} a_{ij} x_j^* = \sum_{j: x_j^* \geq 1/f} a_{ij} x_j^* + \sum_{j: x_j^* < 1/f} a_{ij} x_j^* < \sum_{j: x_j^* \geq 1/f} a_{ij} x_j^* + 1 \leq 1$$

Contradicting to the fact that $\sum_{j=1}^{n} a_{ij} x_j^* \geq 1$

As for the approximation ratio, it's easy to see that:

$$cost(x^I) = \sum_{j=1}^{n} w_j x_j^I \leq \sum_{j=1}^{n} w_j (f x_j^*) = f \cdot OPT(LP)$$

□

GENERAL COVER

Definition 3 *Given a collection of multisets of U (rather than a collection of sets). A* multiset *contains a specified number of copies of each element. Let a_{ij} denote the multiplicity of element i in multiset S_j. Find a sub-collection $\mathcal{C}$ such that the weight of $\mathcal{C}$ is minimum and $\mathcal{C}$ covers each element i b_i times. (rather than just one time)*

The corresponding IP:

$$\begin{array}{ll} \min & \sum_{j=1}^{n} w_j x_j \\ \text{st} & a_{i1} x_1 + \cdots + a_{in} x_n \geq b_i \quad \forall i \in \{1, ..., m\} \\ & x_j \in \{0, 1\} \quad \forall j \in \{1, ..., n\} \end{array} \tag{5}$$

The corresponding LP:

$$
\begin{array}{ll}
\min & \sum_{j=1}^{n} w_j x_j \\
\text{st} & a_{i1}x_1 + \cdots + a_{in}x_n \geq b_i \quad \forall i \in \{1, ..., m\} \\
& 0 \leq x_j \leq 1 \quad \forall j \in \{1, ..., n\}
\end{array}
\tag{6}
$$

Question: Can we use the same approached as in the weighted set cover problem to obtain an f-approximation algorithm where $f = max_i \sum_{j=1}^{n} a_{ij}$?

Answer: Yes

SCHEDULING ON UNRELATED PARALLEL MACHINES

Definition 4 *: Given a set J of n jobs, a set M of m machines, and for each $j \in J$ and $i \in M$, $p_{ij} \in \mathbb{Z}^+$ be the time taken to process job j on machine i. Find a schedule (for these jobs on these machines) so that the maximum completion time of any machine is minimized.*

Let x_{ij} be a variable indicating whether job j is to be processed on machine i, that is, $x_{ij} = 1$ if j is to be processed on i; otherwise, $x_{ij} = 0$.

Let t be the *makespan*, i.e., the maximum processing time of any machine. Clearly, the objective is to minimize t. The following IP is equivalent to our problem:

$$
\begin{array}{ll}
\min & t \\
\text{st} & \sum_{i=1}^{m} x_{ij} = 1 \qquad \forall j \in J \\
& \sum_{j=1}^{n} x_{ij} p_{ij} \leq t \qquad \forall i \in M \\
& x_{ij} \in \{0, 1\} \qquad \forall i \in M, \forall j \in J
\end{array}
\tag{7}
$$

Remarks:

- The first constraint ensures that all jobs are scheduled
- The second constraint ensures that the maximum completion time of any machine is at most t

The corresponding LP:

$$
\begin{array}{ll}
\min & t \\
\text{st} & \sum_{i=1}^{m} x_{ij} = 1 \qquad \forall j \in J \\
& \sum_{j=1}^{n} x_{ij} p_{ij} \leq t \qquad \forall i \in M \\
& x_{ij} \geq 0 \qquad \forall i \in M, \forall j \in J
\end{array}
\tag{8}
$$

Exercise 2 *Give an example showing that the integrality gap of (7) is at least m*

An Example: Suppose $|J| = 1$, i.e, we have only one job, and $p_{ij} = m$ for all $i \in M$. Then the minimum makespan is m, i.e, $OPT(IP) = m$. However, the optimal solution to (8) is to schedule the job to extent of $1/m$ on each machine, thus $OPT(LP) = 1$. Hence, the integrality gap is m.

Questions: What do we "learn" from the above example?

- In (7), if $p_{ij} > t$, then the IP automatically sets $x_{ij} = 0$
- In (8), if $p_{ij} > t$, the relaxation allows us to set x_{ij} to nonzero "fractional" values, thereby raising this question: Is there anyway that we can set $x_{ij} = 0$ if $p_{ij} > t$ in the LP?

Recall that an *extreme point solution* of a set of linear inequalities is a feasible solution that cannot be expressed as convex combination of two other feasible solutions. Let polyhedron P be a feasible region defined by a set of linear inequalities. Then an *extreme point solution* is a vertex of P.

Suppose that we can find a schedule in which the integral schedule's makespan is at most $T \in \mathbb{Z}^+$, and the fractional variables x_{ij} all correspond to $p_{ij} \leq T$. For all these jobs (corresponding to fractional variables x_{ij}, we can match to machines in a one-to-one manner. If such a T exists, we have the following system, called polyhedron $P(T)$.

$$\begin{array}{lll} \sum_{i:(i,j)\in S_T} x_{ij} & = 1 & j \in J \\ \sum_{j:(i,j)\in S_T} x_{ij}p_{ij} & \leq T & i \in M \\ x_{ij} \geq 0 \quad (i,j) \in S_T & & \end{array} \tag{9}$$

where $S_T = \{(i,j) \mid p_{ij} \leq T\}$

Lemma 1 *Any extreme point solution x^* to $P(T)$ has at most $n+m$ positives (non-zero) variables*

Lemma 2 *For any extreme point solution x^* to $P(T)$, the number of fractionally assigned jobs (according to x^*) is at most m.*

Proof. Let a and b be the number of jobs that ar integrally and fractionally set by x^*. Each fractional job is assigned to at least 2 machines, therefore, results in at least 2 positive (non-zero) entries in x^*. Thus we have:

$a + b = n$ and $a + 2b \leq n + m$. The second inequality dues to Lemma 1. It follows that $b \leq m$ and $a \geq n - m$.

□

Corresponding to x^*, define a bipartite graph $G = (A, B; E)$ as follows:

- A is the set of fractionally assigned jobs
- B is the set of machines to which some fractional jobs were assigned
- Edge $(i, j) \in E$ iff x^*_{ij} is fractional

Lemma 3 *Given such a G, there exists a perfect matching from A into B*

Proof. We will prove this lemma using Hall's Theorem. (Given a bipartite graph $G = (A, B; E)$, G has a perfect matching that matches all vertices of A iff for every set $S \subseteq A$, the neighborhood of S, i.e, $N(S) = \{y \in B \mid \exists x \in S, (x, y) \in E\}$, is at least as large as S.)

Consider any subset $S \subseteq A$ (note that S is a subset of fractionally assigned jobs), we only need to check that $|N(S)| \geq |S|$.

Let $H = (S, N(S); E(H))$ be the subgraph of G induced by S and $N(S)$. Now, we will prove that $|E(H)| \leq |S| + |N(S)|$. Consider the polyhedron P which is defined in the same way as $P(T^*)$ restricting to variables corresponding to edges in $E(H)$. Let y^* be x^* restricted to H. Then y^* is an extreme solution (vertex) in P. Based on Lemma 1, y^* has at most $|S| + |N(S)|$ positive variables. Since the number of positive variables is exactly the number of edges in H. Thus $|E(H)| \leq |S| + |N(S)|$.

Additionally, each fractionally assigned job must be assigned to at least 2 machines. Consequently, $2|S| \leq |E(H)| \leq |S| + |N(S)|$. Hence $|S| \leq |N(S)|$.

□

A matching in G is called a perfect matching if it matches every job $j \in A$.

Now, we see that if $P(T)$ has a feasible solution, we are able to find a schedule in which the integral schedule's makespan is at most T, and for the fractional jobs, we can find an one-to-one matching between these jobs and machines. Hence, we can find the schedule with a factor of $2T$.

How do we find a range of T?

Let $\alpha = \max_j \min_i p_{ij}$ and β be the makespan of the greedy schedule which assigns each job to a machines with minimum processing time. Then we have the following algorithm:

Algorithm 1 Scheduling on unrelated parallel machine

Use binary search to find the least value $T^* \in [\alpha, \beta]$ for which $P(T^*)$ is feasible
Find an extreme solution x^* of $P(T^*)$
Assign all integrally set jobs to machines as in x^*
Construct the bipartite graph $G = (A, B; E)$ as described above and find a perfect matching $\mathcal{M}$
Assign fractionally set jobs to machines according to $\mathcal{M}$

Theorem 3 *The above algorithm achieves an approximation guarantee of factor 2*

Proof. Let OPT be the optimal solution of our problem. Note that the jobs are assigned in two steps. In the first step, all integrally set jobs are assigned based on x^*. Hence the makespan in this step is at most $T^* \leq OPT$.

In the second step, we assign all fractional set jobs. Note that each edge (i, j) of G satisfies $p_{ij} \leq T^*$. The perfect matching $\mathcal{M}$ found in G schedules at most one extra job on each machine. Thus their makespan is at most T^*.

Therefore, both both steps, the total makespan is $\leq 2OPT$.

$\square$

Filtering and Rounding

Overview of Filtering and Rounding techniques:

- Formulate the optimization problem as an IP
- Formulate the corresponding LP
- Use an optimal solution to the LP to construct a "filtered" version of the optimization problem (creating a restricted version of the IP.) This step often involve with fixing a parameter ϵ and try to define a new set of feasible solution to LP with a cost within $(1+\epsilon)$ of the LP
- Using rounding technique to produce a good integral solution to the filtered problem, which is also a solution for an original IP

Let's see how to use this technique to solve the metric uncapacitated facility location problem.

METRIC UNCAPACITATED FACILITY LOCATION

Definition 5 *Given a set F of facilities and a set C of cities. Let f_i be the cost of opening facility $i \in F$, and c_{ij} be the cost of connecting city j to (opened) facility i where c_{ij} satisfies the triangle inequality. The problem is to find a subset $O \subseteq F$ of facilities that should be opened, and an assignment function $\theta : C \to O$ assigning every city j to open facilities $\theta(j)$ in such a way that the total cost of opening facilities and connecting cities to open facilities is minimized, that is, to minimize the objective function:*

$$\sum_{i \in O} f_i + \sum_{j \in C} c_{\theta(j)j}$$

Let x_i be an indicator variable denoting whether facility i is opened and y_{ij} be an indicator variable denoting whether city j is assigned to facility i. We have the following IP:

$$
\begin{array}{ll}
\min & \sum_{i\in F} f_i x_i + \sum_{i\in F, j\in C} c_{ij} y_{ij} \\
\text{st} & \sum_{i\in F} y_{ij} = 1 \quad j\in C \\
& x_i - y_{ij} \geq 0 \quad i\in F, j\in C \\
& x_i, y_{ij} \in \{0,1\} \quad i\in F, j\in C
\end{array} \tag{10}
$$

Remarks:

- The first constraint ensures that each city is connected to a facility
- The second constraint ensures this facility must be opened

We have the corresponding LP as follows:

$$
\begin{array}{ll}
\min & \sum_{i\in F} f_i x_i + \sum_{i\in F, j\in C} c_{ij} y_{ij} \\
\text{st} & \sum_{i\in F} y_{ij} = 1 \quad j\in C \\
& x_i - y_{ij} \geq 0 \quad i\in F, j\in C \\
& x_i \geq 0 \quad i\in F \\
& y_{ij} \geq 0 \quad i\in F, j\in C
\end{array} \tag{11}
$$

Let (x^*, y^*) be an optimal solution to the LP. Note that we interpret a fractional x_i^* as a partially open facility and a fractional y_{ij}^* as a partial assignment of city j to facility i.

Let $F(x) = \sum_{i\in F} f_i x_i$ and $\Theta(y) = \sum_{i\in F, j\in C} c_{ij} y_{ij}$.

The Filtering Step

Consider LP (11). Note that when $y_{ij} > 0$, the city j can be assigned to candidate facility i. However, the cost c_{ij} might be too large.
Question: How do we control it? How do we filter out these facilities?

Fix a parameter $\epsilon > 0$ (we will determine ϵ later). For each city j, we will filter out all facilities i such that c_{ij} is greater than $(1+\epsilon)$ of the optimal assignment cost (in an LP).

Let the current optimal assignment cost for city j be

$$\Theta_j^* = \sum_{i \in F} c_{ij} y_{ij}^*$$

Note that y_{ij}^* is an optimal solution to LP (11).

Define F_j as a set of possible (candidate) facilities such that we can assign j to as follows:

$$F_j = \{i \mid y_{ij}^* > 0, c_{ij} \leq (1+\epsilon)\Theta_j^*\}$$

Using the filtering, we construct a feasible (x', y') to LP (11) so that the following two conditions must be held:

- The cost of (x', y') is not too far from the cost of (x^*, y^*)
- $y'_{ij} > 0$ implies that $c_{ij} \leq (1+\epsilon)\Theta_j^*$

Note that the second condition also implies that when $c_{ij} > (1+\epsilon)\Theta_j^*$, set $y'_{ij} = 0$.

Now, how do we define y'_{ij} based on y_{ij}^*? Note that we set $\sum_{i \in F} y'_{ij} = 1$. Therefore, define:

$$y'_{ij} = \frac{y_{ij}^*}{\sum_{i \in F_j} y_{ij}^*} \qquad \text{if } i \in F_j$$

Otherwise, $y'_{ij} = 0$

Now, how do we define x'_i. Recall that x' must satisfy the second constrain, which is $x'_i - y'_{ij} \geq 0$. So first, let us compare the cost of y' and y^*.

$$\Theta_j^* = \sum_{i \in F} c_{ij} y_{ij}^* > \sum_{i \notin F_j} c_{ij} y_{ij}^* > (1+\epsilon)\Theta_j^* \sum_{i \notin F_j} y_{ij}^* = (1+\epsilon)\Theta_j^* \left(1 - \sum_{i \in F_j} y_{ij}^*\right)$$

Therefore,

$$\frac{1}{\sum_{i \in F_j} y_{ij}^*} < \frac{1+\epsilon}{\epsilon}$$

Which means $y'_{ij} \leq (\frac{1+\epsilon}{\epsilon})y^*_{ij}$.

Thus, define

$$x'_i = min\{1, (\frac{1+\epsilon}{\epsilon})x^*_i\}$$

Hence, we have a filter solution (x', y') for LP (11).

The Rounding Step

We will round (x', y') to the feasible integer solutions to IP (10) as follows:

1. Pick an unassigned city j with the smallest assignment cost Θ^*_j.
2. For this city j, we open the facility $\theta(j) \in F_j$ with the smallest opening cost $f_{\theta(j)}$, that is, round $x'_{\theta(j)}$ to 1 and $y'_{\theta(j)j}$ to 1.
3. For all other cities j' such that $F'_j \cap F_j \neq \emptyset$, round $y'_{\theta(j)j'} = 1$, that is assign cities j' to open facility $\theta(j)$
4. Repeat this process until all cities are assigned

Theorem 4 *The above algorithm has an approximation ratio of* $max\{3(1+\epsilon), (1+1/\epsilon)\}$. *Set* $\epsilon = 1/3$, *we obtain an approximation algorithm within the ratio of 4.*

Proof. Let $(\hat{x}, \hat{y})$ be a solution obtained from our algorithm. Let $Z(\hat{x}, \hat{y})$ be the cost of the objective function. Recall that $F(x) = \sum_{i \in F} f_i x_i$ and $\Theta(y) = \sum_{i \in F, j \in C} c_{ij} y_{ij}$. Hence $Z(\hat{x}, \hat{y}) = F(\hat{x}) + \Theta(\hat{y})$. Likewise, let $Z(x', y') = F(x') + \Theta(y')$ and $Z(x^*, y^*) = F(x^*) + \Theta(y^*)$. So, $Z(x^*, y^*)$ is the optimal cost of LP (11). Let OPT be the optimal cost of our optimization problem.

First, let compare $Z(x', y')$ with $Z(x^*, y^*)$. We have:

$$F(x') = \sum_i f_i x'_i \leq \sum_i f_i (1 + 1/\epsilon) x^*_i = (1 + 1/\epsilon) F(x^*)$$

$$\Theta(y') = \sum_j \sum_i c_{ij} y'_{ij} \leq \sum_j (1+\epsilon)\Theta^*_j \sum_i y'_{ij} \leq (1+\epsilon)\Theta(y^*)$$

Note that in the rounding procedure, for the least cost unassigned city j, we open only one facility $\theta(j)$ with the least opening cost $f_{\theta(j)}$ in F_j. We have:

$$f_{\theta(j)} = f_{\theta(j)} \sum_{i \in F_j} y'_{ij} \leq f_{\theta(j)} \sum_{i \in F_j} x'_i \leq \sum_{i \in F_j} f_i x'_i$$

So the cost of opening facility $\theta(j)$ is at most the fractional cost of all facilities that can "cover" j. Additionally, for any other j', the facility opening cost is not increased (since we don't open any facility). Therefore,

$$F(\hat{x}) \leq F(x') \leq (1 + 1/\epsilon) F(x^*)$$

However, for each j', we have increased the assignment cost to $\theta(j)$. Note that the cost c_{ij} satisfies the triangle inequality, we have:

$$c_{\theta(j)j'} \leq c_{ij'} + c_{ij} + c_{\theta(j)j} \leq (1+\epsilon)\Theta^*_{j'} + (1+\epsilon)\Theta^*_j + (1+\epsilon)\Theta^*_j \leq 3(1+\epsilon)\Theta^*_{j'}$$

Thus,

$$\Theta(\hat{y}) = \sum_j c_{\theta(j)j} \leq 3(1+\epsilon) \sum_j \Theta^*_j = 3(1+\epsilon)\Theta(y^*)$$

In total, we have:

$$Z(\hat{x}, \hat{y}) = F(\hat{x}) + \Theta(\hat{y})$$
$$\leq (1 + 1/\epsilon) F(x^*) + 3(1+\epsilon)\Theta(y^*) \leq max\{3(1+\epsilon), (1+1/\epsilon)\} OPT$$

Hence the approximation ratio is $max\{3(1+\epsilon), (1+1/\epsilon)\}$. When $\epsilon = 1/3$ (we choose such ϵ to minimize this maximum), we have the ratio of 4.

$\square$

Randomized Rounding

Overview of randomized rounding techniques:

- Similar to the rounding techniques, except that at the rounding step, we round an optimal fractional solution x^* *randomly* according to some probability distribution
- Goal of the rounding: Obtain a good approximation ratio with high probability (some positive constant, e.g., $> 1/2$)
- Independently run the algorithm many times to increase this probability

WEIGHTED SET COVER (WSC)

As above, we have the LP (after using the relaxation techniques) of the WSC as follows:

$$\begin{array}{ll} \min & \sum_{j=1}^{n} w_j x_j \\ \text{st} & \sum_{j: i \in S_j} x_j \geq 1 \qquad \forall i = 1 \dots m \\ & x_j \geq 0 \qquad \forall j = 1 \dots n \end{array} \tag{12}$$

Recall that $x_j = 1$ iff $S_j \in \mathcal{C}$; otherwise, $x_j = 0$.

Suppose we have an optimal solution x^* of LP (12). To obtain x^I, we use the randomized rounding as follows:

Round x_j^* to 1 with probability x_j^*, that is $\mathbf{Pr}[x_j^I = 1] = x_j^*$. (Also round x_j^* to 0 with probability $1 - x_j^*$)

Note that $\mathbf{Pr}[x_j^I = 1]$ represents the *probability* that the set S_j is selected in the sub-collection $\mathcal{C}$.

Then the expected cost of our solutions is:

$$\mathbf{E}[cost(x^I)] = \sum_{j=1}^{n} \mathbf{Pr}[x_j^I = 1] \cdot w_j = \sum_{j=1}^{n} w_j x_j^* = OPT(LP)$$

Note that x^I should be feasible and $cost(x^I) \leq \rho \cdot OPT$ where OPT is the cost of the optimal solutions to IP and ρ is some approximation ratio. Therefore, what we really want is to find the probability that x^I satisfies the above requirements. If this probability is at least some positive constant, then we say that ρ is an approximation ratio of this algorithm.

We have:

$$\begin{aligned} & \mathbf{Pr}[x^I \text{ is feasible and } \mathrm{cost}(x^I) \leq \rho \cdot OPT] \\ = \ & 1 - \mathbf{Pr}[x^I \text{ is not feasible or } \mathrm{cost}(x^I) > \rho \cdot OPT] \\ \geq \ & 1 - \mathbf{Pr}[x^I \text{ is not feasible}] - \mathbf{Pr}[\mathrm{cost}(x^I) > \rho \cdot OPT] \end{aligned}$$

Now, we need to find $\mathbf{Pr}[x^I \text{ is not feasible}]$ and $\mathbf{Pr}[\mathrm{cost}(x^I) > \rho \cdot OPT]$.

Let's find the $\mathbf{Pr}[x^I \text{ is not feasible}]$ first.

Consider an element $i \in U$. Suppose that i occurs in k sets of $\mathcal{S}$, we have:

$$x_{j1} + x_{j2} + \cdots + x_{jk} \geq 1$$

(Note that the above inequality is the inequality in the matrix A corresponding to element i (i^{th} rows)).

The probability that this element is not covered is:

$$\mathbf{Pr}[i \text{ is not covered}] = (1 - x^*_{j1}) \cdots (1 - x^*_{jk})$$

$$\leq \left(\frac{k - (x^*_{j1} + \cdots + x^*_{jk})}{k} \right)^k \leq (1 - \frac{1}{k})^k \leq \frac{1}{e}$$

Therefore,

$$\mathbf{Pr}[x^I \text{ is not feasible}] \leq \sum_{i=1}^{m} \mathbf{Pr}[i \text{ is not covered}] \leq \frac{m}{e}$$

This bound is very bad since m is large. We can obtain a better one by independently running the above strategy $t > 0$ (to be determined) times and round x_j^* to 0 with the probability $(1 - x_j^*)^t$ (instead of $(1 - x_j^*)$), that is, set x_j^I to 0 when $x_j^I = 0$ in all t rounds. Then we have:

$$\mathbf{Pr}[i \text{ is not covered}] \leq (\frac{1}{e})^t$$

Thus, $\mathbf{Pr}[x^I \text{ is not feasible}] \leq m(\frac{1}{e})^t$. When t is logarithmically large, i.e., $t = \theta(\log m)$, then $m(\frac{1}{e})^t \leq 1$.

Now, we consider the $\mathbf{Pr}[\text{cost}(x^I) > \rho \cdot OPT]$.

We have already proved that $\mathbf{E}[cost(x^I)] = OPT(LP) \leq OPT$ in one time. So when we run this strategy in t times, we have $\mathbf{E}[cost(x^I)] \leq t \cdot OPT$. Based on the Markov's Inequality, we have:

$$\mathbf{Pr}[\text{cost}(x^I) > \rho \cdot OPT] \leq \frac{\mathbf{E}[cost(x^I)]}{\rho \cdot OPT} \leq \frac{t \cdot OPT}{\rho \cdot OPT} = \frac{t}{\rho}$$

Therefore,

$$\mathbf{Pr}[x^I \text{ is feasible and cost}(x^I) \leq \rho \cdot OPT] \geq 1 - m(\frac{1}{e})^t - \frac{t}{\rho}$$

Now, set $t = \theta(\log m)$ and $\rho = 4t$ so that $1 - m(\frac{1}{e})^t - \frac{t}{\rho} \geq 1/2$. We conclude that this algorithm has an approximation ratio within a factor of $O(\log m)$ with probability at least 1/2.

Remark: Let X be a nonnegative random variable with a known expectation and a positive number $p \in \mathbb{R}^+$, Markov's Inequality says that

$$\mathbf{Pr}[X \geq p] \leq \frac{\mathbf{E}[X]}{p}$$

MAXIMUM SATISFIABILITY (MAX-SAT)

Definition 6 *Given a conjunctive normal form formula f on Boolean variables $X = \{x_1, \ldots, x_n\}$, consisting of m clauses $C_1, \ldots, C_m$ weighted $w_1, \ldots, w_m \in \mathbb{Z}^+$. Find a truth assignment to the Boolean variables that maximizes the total weight of satisfied clauses.*

Remarks:

- *Literals*: Boolean variables or their negations. For example, if $x \in X$, then x and $\bar{x}$ are literals over X.
- *Clause*: a disjunction of literals
- *Conjunctive normal form*: a conjunction of clauses
- *Truth assignment*: A truth assignment for X is a function $t : X \to \{T, F\}$. If $t(x) = T$, we say x is true under t, otherwise, x is false. The literal x is true under t iff the variable x is true under t.
- *Satisfied clause*: A clause is *satisfied* by a truth assignment iff at least one of its members is true under that assignment. For example, we have a clause $C_1 = \{x_1, \bar{x}_2, x_3\}$. We say C_1 is satisfied by t *unless* $t(x_1) = F$, $t(x_2) = T$, and $t(x_3) = F$.
- *Satisfied collection*: A collection $\mathcal{C}$ of clauses over X is satisfiable iff there exists some truth assignment for X that simultaneously satisfies all the clauses $C_j \in \mathcal{C}$.

An Example: Let $X = \{x_1, x_2\}$ and $\mathcal{C} = \{C_1, C_2\}$ where $C_1 = \{x_1, \bar{x}_2\}$ and $C_2 = \{\bar{x}_1, x_2\}$. Define t as $t(x_1) = T$ and $t(x_2) = T$, then $\mathcal{C}$ is satisfiable under t.

If $\mathcal{C} = \{\{x_1, x_2\}, \{x_1, \bar{x}_2\}, \{\bar{x}_1\}\}$, then $\mathcal{C}$ is not satisfiable (under any truth assignment t).

1. A Simple Randomized Algorithm for MAX-SAT

1: For each variable $x_i \in X$
2: independently set $t(x_i) = T$ with probability 1/2

Theorem 5 *Let W be the cost (weight) of a random assignment t and OPT be the cost of an optimal assignment, then* $\mathbf{E}[W] \geq \frac{1}{2}OPT$

Proof. Let l_j denote the length of clause C_j and W_j denote the random variable indicating the event $\{C_j$ is satisfied$\}$, that is: $W_j = 1$ if C_j is satisfied; otherwise, $W_j = 0$. We have:

$$\mathbf{E}[W] = \sum_{j=1}^{m} w_j \mathbf{Pr}[W_j = 1] = \sum_{j=1}^{m} w_j[1 - (1/2)^{l_j}] \geq \frac{1}{2}\sum_{j=1}^{m} w_j \geq \frac{1}{2}OPT$$

□

Hence, we conclude that the above algorithm can obtain an (expected) approximation ratio 2. Now, instead of converting this into a high probability statement, we will show how to derandomize this algorithm by a method known as *conditional expectation*. With this method, the algorithm will deterministically computes a truth assignment such that the weight of satisfied clauses is $\geq \mathbf{E}[W]$

The important point of this method is that for setting x_1 to truth (or false), we can find a formula f' in polynomial time on the remaining $n-1$ variables by removing all clauses that are satisfied (where $x_1 = T$). The solution of f' is the solution of f having $x_1 = T$.

The basic ideas of the conditional expectation is as follows.

Consider a fixed $k \in [n]$. Let $a_1, \ldots, a_i, 1 \leq i \leq k$ be a truth assignment to $x_1, \ldots, x_i$. Let f' be a formula obtained by setting $x_1 = a_1, \ldots, x_i = a_i$ and removing a set C' of all clauses that are already satisfied. Then,

$$\mathbf{E}[W_f \mid x_i = a_i, 1 \leq i \leq k] = \mathbf{E}[W_{f'}] + \sum_{c \in C'} w_c$$

Thus we can compute $\mathbf{E}[W_f \mid x_i = a_i, 1 \le i \le k]$ in polynomial time. Also notice that for $k \ge 1$, we have

$$\begin{aligned}\mathbf{E}[W \mid x_i = a_i, 1 \le i \le k-1] = \\ &\mathbf{E}[W \mid x_i = a_i, 1 \le i \le k-1, x_k = T]/2+ \\ &\mathbf{E}[W \mid x_i = a_i, 1 \le i \le k-1, x_k = F]/2\end{aligned}$$

It follows that the larger value between the two expectations on the right hand side is at least as large as that of the left hand side. Therefore, we can set x_1 to be T or F, compute the corresponding $\mathbf{E}$, then following the path that leads to the larger $\mathbf{E}$. Eventually, we will get a truth assignment such that the total weight of clauses satisfied by this assignment by at least $\mathbf{E}[W]$.

2. A Randomized Rounding (LP Rounding) Algorithm for MAX-SAT

Notations:

- $y_i = 1$ iff $x_i =$ True; otherwise, $y_i = 0$
- $z_j = 1$ iff C_j is satisfied; otherwise, $z_j = 0$

We can formulate the MAX-SAT as the following IP:

$$\begin{array}{ll} \max & \sum_{j=1}^{m} w_j z_j \\ \text{st} & \sum_{i:x_i \in C_j} y_i + \sum_{i:\bar{x}_i \in C_j} (1 - y_i) \ge z_j \quad \forall j = 1 \cdots m \\ & y_i, z_j \in \{0, 1\} \quad \forall i = 1 \cdots n, \forall j = 1 \cdots m \end{array} \tag{13}$$

Its equivalent LP (after the relaxation technique) is:

$$\begin{array}{ll} \max & \sum_{j=1}^{m} w_j z_j \\ \text{st} & \sum_{i:x_i \in C_j} y_i + \sum_{i:\bar{x}_i \in C_j} (1 - y_i) \ge z_j \quad \forall j = 1 \cdots m \\ & 0 \le y_i \le 1 \quad \forall = 1 \cdots n \\ & 0 \le z_j \le 1 \forall j = 1 \cdots m \end{array} \tag{14}$$

Let (y^*, z^*) be the optimal solution to the LP (14). Similar to the rounding strategy for SET COVER, round x_i to True with probability y_i^*. We have:

Theorem 6 *The above algorithm has an $e/(e-1)$ approximation ratio.*

Proof. Again, let W be the cost (weight) of the obtained solution, we have:

$$\begin{aligned}
\mathbf{E}[W] &= \sum_{j=1}^{m} w_j \mathbf{Pr}[W_j = 1] \\
&= \sum_{j=1}^{m} w_j \left(1 - \prod_{i:x_i \in C_j} (1 - y_i^*) \prod_{i:\bar{x}_i \in C_j} y_i^*\right) \\
&\geq \sum_{j=1}^{m} w_j \left(1 - \left[\frac{\sum_{i:x_i \in C_j} (1 - y_i^*) + \sum_{i:\bar{x}_i \in C_j} y_i^*}{l_j}\right]^{l_j}\right) \\
&= \sum_{j=1}^{m} w_j \left(1 - \left[\frac{l_j - \left(\sum_{i:x_i \in C_j} y_i^* + \sum_{i:\bar{x}_i \in C_j} (1 - y_i^*)\right)}{l_j}\right]^{l_j}\right) \\
&\geq \sum_{j=1}^{m} w_j \left(1 - \left[1 - \frac{z_j^*}{l_j}\right]^{l_j}\right) \\
&\geq \sum_{j=1}^{m} w_j \left(1 - \left[1 - \frac{1}{l_j}\right]^{l_j}\right) z_j^* \qquad (*) \\
&\geq \min_j \left(1 - \left[1 - \frac{1}{l_j}\right]^{l_j}\right) \sum_{j=1}^{m} w_j z_j^* \\
&\geq \left(1 - \frac{1}{e}\right) OPT
\end{aligned} \tag{15}$$

Note that at the step $(*)$, the function $g(z) = 1 - \left(1 - \frac{z}{l_j}\right)^{l_j}$ is a concave function with $g(0) = 0$ and $g(1) = 1 - \left(1 - \frac{1}{l_j}\right)^{l_j}$. Therefore, for $z \in [0, 1], g(z) \geq \left(1 - \left(1 - \frac{1}{l_j}\right)^{l_j}\right) z$.

Hence the above algorithm has an (expected) approximation ratio $e/(e-1)$.

Again, we can derandomize this algorithm and obtain a deterministic $e/(e-1)$ ratio.

$\square$

3. A Combination of the Above Two Algorithms

Note that for the randomized algorithm, called $\mathcal{A}_1$, we have:

$$\mathbf{E}[W] = \sum_{j=1}^{m} w_j \mathbf{Pr}[W_j = 1] = \sum_{j=1}^{m} [1 - (1/2)^{l_j}] OPT$$

and for the LP randomized rounding algorithm, called $\mathcal{A}_2$, we have:

$$\mathbf{E}[W] \geq \left(1 - \left[1 - \frac{1}{l_j}\right]^{l_j}\right) OPT$$

Notice that when l_j increases, $[1-(1/2)^{l_j}]$ increases whereas $\left(1 - \left[1 - \frac{1}{l_j}\right]^{l_j}\right)$ decreases. It implies that when the length of each clause ≤ 2, $\mathcal{A}_2$ works well whereas $\mathcal{A}_1$ works better when the length of each clause ≥ 2. Therefore, it is natural to combine these two algorithms as follows:

Run both algorithms $\mathcal{A}_1$, $\mathcal{A}_2$ and report the better assignment. Call this algorithm $\mathcal{A}$.

Theorem 7 *The algorithm $\mathcal{A}$ has an approximation ratio 4/3.*

Proof. Let W^1 and W^2 represent the cost obtained from algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$ respectively. We have:

$$\begin{aligned}
\mathbf{E}[\max\{W^1, W^2\}] &\geq \mathbf{E}\left[\frac{\mathbf{E}[W^1] + \mathbf{E}[W^2]}{2}\right] \\
&\geq \sum_{j=1}^{m} w_j \left(\frac{1}{2}\left(1 - \frac{1}{2^{l_j}}\right) + \frac{1}{2}\left(1 - \left[1 - \frac{1}{l^j}\right]^{l_j}\right) z_j^*\right) \\
&\geq \sum_{j=1}^{m} \frac{\alpha_{l_j} + \beta_{l_j}}{2} \sum_{j=1}^{m} w_j z_j^* \\
&\geq \sum_{j=1}^{m} \frac{\alpha_{l_j} + \beta_{l_j}}{2} OPT
\end{aligned}$$

where $\alpha_{l_j} = \left(1 - \frac{1}{2^{l^j}}\right)$ and $\beta_{l_j} = \left(1 - \left[1 - \frac{1}{l^j}\right]^{l^j}\right)$

Note that when $l_j = 1$ or $l_j = 2$, we have $\alpha_{l_j} + \beta_{l_j} \geq 3/2$ and when $l_j \geq 3$, $\alpha_{l_j} + \beta_{l_j} \geq 7/8 + (1 - 1/e) \geq 3/2$. Therefore:

$$\mathbf{E}[\max\{W^1, W^2\}] \geq \frac{3}{4} OPT$$

$\square$