LP Relaxation, Rounding, and Randomized Rounding

1 Cut Problems

1.1 Max-flow min-cut

A *flow network* is a directed graph $D = (V, E)$ with two distinguished vertices s and t called the *source* and the *sink*, respectively. Moreover, each arc $(u, v) \in E$ has a certain *capacity* $c(u, v) \geq 0$ assigned to it.

Let X be a proper non-empty subset of V. Let $\bar{X} := V - X$, then the pair $(X, \bar{X})$ forms a partition of V, called a *cut* of D. The set of arcs of D going from X to $\bar{X}$ is called an *edge cut* of G, denoted by $[X, \bar{X}]$.

A *source/sink cut* of a network D is a cut (S, T) with $s \in S$ and $t \in T$. (Note that, implicitly $T = \bar{S}$.) Given a source/sink cut (S, T), the *capacity* of the cut, denoted by cap(S, T) is the total capacity of edges leaving S:

$$\text{cap}(S, T) := \sum_{\substack{u \in S, v \in T, \\ (u,v) \in E}} c(u, v).$$

A cut with minimum capacity is called a *minimum cut*.

A *flow* for a network $D = (V, E)$ is a function $f : E \to \mathbb{R}$ which assigns a real number to each edge (u, v). A flow f is called a *feasible flow* if it satisfies the following conditions:

(i) $0 \leq f(u, v) \leq c(u, v), \forall (u, v) \in E$. These are the *capacity constraints*.

(ii) For all $v \in V - \{s, t\}$, the total flow into v is the same as the total flow out of v:

$$\sum_{u:(u,v) \in E} f(u, v) = \sum_{w:(v,w) \in E} f(v, w). \tag{1}$$

These are called the *flow conservation law*.

The *value* of a flow f for D, denoted by val(f), is the net flow out of the source:

$$\text{val}(f) := \sum_{u:(s,u) \in E} f(s, u) - \sum_{v:(v,s) \in E} f(v, s).$$

The evasive *max-flow min-cut* theorem states that the value of a maximum flow is equal to the capacity of a minimum s, t-cut. This theorem can be shown using linear programming duality as follows. Let $\mathcal{P}$ be the set of all paths from s to t. Let f_P denote the flow value sent along path P. It is easy to see that the following linear program is equivalent to the maximum flow problem:

$$\begin{array}{ll} \max & \displaystyle\sum_{P \in \mathcal{P}} f_P \\ \text{subject to} & \displaystyle\sum_{P : e \in P} f_P \leq c_e, \quad \forall e \in E, \\ & \quad f_P \geq 0, \quad \forall P \in \mathcal{P}. \end{array} \tag{2}$$

The dual of this program is

$$
\begin{array}{rll}
\min & \sum_{e \in E} c_e y_e & \\
\text{subject to} & \sum_{e \in P} y_e \geq 1, & \forall P \in \mathcal{P}, \\
& y_e \geq 0, & \forall e \in E.
\end{array} \tag{3}
$$

Note that, each 01-solution to (3) corresponds to a set of edges whose removal disconnect s from t and vice versa. In particular, each s, t-cut corresponds to a 01-solution of (3). Thus, to prove the max-flow min-cut theorem, we only need to show that there is an optimal integral solution to (3). (An optimal integral solution must be a 01-solution.)

Exercise 1. Show that the linear program (3) can be solved in polynomial time with the ellipsoid method.

We will use randomized rounding to obtain such an integral solution. Let $\mathbf{y}^*$ be an optimal solution to (3). Interpret y_e^* as the length of edge e. For each vertex v, let $d(s, v)$ be the distance from s to v, i.e. the length of a shortest path from s to v according to the distance function $\mathbf{y}^*$. Then, for each arc $e = (u, v)$ we have $d(s, v) \leq d(s, u) + y_e^*$. For each radius $r \geq 0$, let $B(r)$ be the set of vertices of distance at most r from s. Note that $t \notin B(r)$ if $r < 1$.

Now, choose r uniformly at random from $[0, 1)$. Consider the cut $C = [B(r), \overline{B(r)}]$ and an arbitrary arc $e = (u, v)$. The arc e belongs to C iff $d(s, u) \leq r < d(s, v)$. Thus,

$$
\text{Prob}[e \in C] = \frac{d(s, v) - d(s, u)}{1 - 0} \leq y_e^*.
$$

Thus,

$$
\text{E}[\text{cap}(C)] = \sum_{e \in E} c_e \text{Prob}[e \in C] \leq \sum_{e \in E} c_e y_e^* = \text{cost}(\mathbf{y}^*).
$$

Thus, there must be at least one (integral) cut C with capacity at most cost$(\mathbf{y}^*)$. That is the cut that we are looking for.

I personally found this result to be rather surprising (and obviously elegant). The argument is **very** typical of the probabilistic method. Let us delve a little more technically into this argument.

- Let $\mathcal{C}$ be the set of all s, t-cuts of the form $[B(r), \overline{B(r)}]$, for $r \in [0, 1)$. Even though number of possible values of r is infinite, there are only finitely many such cuts. By choosing r at random, each cut $C \in \mathcal{C}$ has a probability Prob$[C]$ of being chosen. (This is a slight abuse of notation, since Prob$[C]$ is often used to denote the probability that event C holds.) Then, cap(C) is a random variable defined on this finite sample space. We have, by definition of expectation,

$$
\mu = \text{E}[\text{cap}(C)] = \sum_{C \in \mathcal{C}} \text{cap}(C) \, \text{Prob}[C].
$$

  Thus, there must be a cut C with capacity at most μ. (Recall the basic probabilistic method discussed at the beginning of this note.)

- Secondly, I'd like to explain the relation

$$
\text{E}[\text{cap}(C)] = \sum_{e \in E} c_e \, \text{Prob}[e \in C]
$$

  that we used earlier. This fact does not come directly from the definition of expectation. For any $C \in \mathcal{C}$, let I_e be the 01-random variable indicating if e is in C or not, namely

$$
I_e = \begin{cases} 1 & e \in C \\ 0 & o.w. \end{cases}.
$$

Then, $\text{Prob}[I_e = 1] = \text{Prob}[e \in C]$. Moreover,

$$\text{cap}(C) = \sum_{e \in C} c_e = \sum_{e \in E} c_e I_e.$$

By *linearity of expectation*,

$$\text{E}[\text{cap}(C)] = \text{E}[\sum_{e \in E} c_e I_e] = \sum_{e \in E} c_e \text{E}[I_e] = \sum_{e \in E} c_e \,\text{Prob}[e \in C].$$

This is a very typical argument of the probabilistic method! The nice thing about the linearity of expectation is that it holds whether or not the variables I_e are independent.

- The above two bullets are not surprising. What is surprising is the following relation:

$$\sum_{C \in \mathcal{C}} \text{cap}(C)\,\text{Prob}[C] = \text{cost}(\mathbf{y}^*).$$

The characteristic vector $\mathbf{y}_C$ of any s,t-cut C is a feasible solution to the linear program. Thus,

$$\text{cost}(\mathbf{y}^*) \leq \text{cost}(\mathbf{y}_C) = \text{cap}(C).$$

Hence,

$$\sum_{C \in \mathcal{C}} \text{cap}(C)\,\text{Prob}[C] \geq \sum_{C \in \mathcal{C}} \text{cost}(\mathbf{y}^*)\,\text{Prob}[C] = \text{cost}(\mathbf{y}^*).$$

Equality holds iff $\text{cap}(C) = \text{cost}(\mathbf{y}^*)$ whenever $\text{Prob}[C] \neq 0$. In other words, all cuts in $\mathcal{C}$ are minimum cuts! **That** I found surprising! Can we prove this fact some other way? The following exercise aims to explain this.

Exercise 2. Let $\mathbf{y}^*$ be an optimal solution to (3). Let r be any number in $[0, 1)$. Show that the cut $[B(r), \overline{B(r)}]$ has capacity $\text{cost}(\mathbf{y}^*)$ without using probabilistic arguments. From this exercise, it is clear that we can find a minimum cut in polynomial time. Just take the cut corresponding to $r = 0$, for example. (**Hint:** complementary slackness.)

Exercise 3. This exercise shows a stronger result than that of the previous one. Let $\mathbf{y}$ be any vertex of the polyhedron corresponding to (3). Show that

1. $y_e = d(s, v) - d(s, u)$ for any edge $e = (u, v)$ of the graph,
2. and that $\mathbf{y}$ is a convex combination of characteristic vectors of members of $\mathcal{C}$. Conclude that $\mathbf{y}$ must be a characteristic vector of a cut in $\mathcal{C}$.

Exercise 4. There is another common way to formulate the min-cut problem. Let x_v indicates if $v \in S$ of the s,t-cut (S, T), and y_{uv} indicates if edge (u, v) belongs to the cut. We need a constraint to ensure that $x_u = 1$, $x_v = 0$ implies $y_{uv} = 1$. As usual, this constraint can be written as $y_{uv} \leq x_u - x_v$. The ILP is then

$$\begin{array}{lll} \min & \sum_{e \in E} c_{uv} y_{uv} & \\ \text{subject to} & y_{uv} \geq x_u - x_v, & \forall uv \in E, \\ & x_s = 1, x_t = 0, & \\ & x_v, y_{uv} \in \{0, 1\}, & \forall uv \in E, \forall v \in V. \end{array} \tag{4}$$

Relaxation gives the following LP:

$$\begin{array}{lll} \min & \sum_{e \in E} c_{uv} y_{uv} & \\ \text{subject to} & y_{uv} \geq x_u - x_v, & \forall uv \in E, \\ & x_s = 1, x_t = 0, & \\ & \quad x_v, y_{uv} \geq 0, & \forall uv \in E, \forall v \in V. \end{array} \tag{5}$$

Show that (5) has an optimal integral solution using the randomized rounding method. (**Hint:** pick $r \in (0, 1]$ at random. Set $S = \{v \mid x_v \geq r\}$.)

Exercise 5. Explain how to use the min-cut procedure for directed graphs (which we have developed) to find a minimum s, t-cut in an undirected graph.

Exercise 6 (Multiway cut). The MULTIWAY CUT problem is a natural generalization of the min-cut problem. Given an undirected graph G with positive edge capacities. There are $k \geq 2$ terminals $t_1, \dots, t_k$ and we would like to find a minimum capacity subset of edges whose removal disconnects the terminals from each other. Formulate an ILP for this problem in a similar fashion to (3).

(a) Write down the LP relaxation of the ILP.

(b) Show that the LP has the half-integrality property, i.e. each vertex of the corresponding polyhedron is half-integral.

(c) Use the randomized rounding method to show that, given any feasible solution $\mathbf{y}$ to the LP, there is an integral solution with capacity at most $2 \operatorname{cost}(\mathbf{y})$.

(**Hint**: pick $r \in [0, 1/2]$ at random. Consider the balls $B_{t_i}(r)$ of radius r around each terminal t_i. Choose the cut $C = \bigcup_i [B_{t_i}, \overline{B_{t_i}}]$. Show that the expected capacity of C is at most $2 \operatorname{cost}(\mathbf{y})$.)

(d) Derandomize the above procedure and give a modification to yield a deterministic $(2 - 2/k)$-approximation algorithm for the MULTIWAY CUT problem.

1.2 Multiway cut (TBD)

2 Satisfiability Problems

A *conjunctive normal form* (CNF) formula is a boolean formula on n variables $\mathcal{X} = \{x_1, \dots, x_n\}$ consisting of m *clauses* $C_1, \dots, C_m$. Each clause is a subset of *literals*, which are variables and negations of variables. A clause can be viewed as the sum (or the OR) of the literals. A clause is satisfied by a truth assignment $a : \mathcal{X} \to \{$TRUE, FALSE$\}$ if one of the literals in the clause is TRUE.

For integers $k \geq 2$, a *k-CNF formula* is a CNF formula in which each clause is of size at most k, an *Ek-CNF formula* is a CNF formula in which each clause is of size exactly k.

Given a CNF formula φ, the MAX-SAT problem is to find a truth assignment satisfying the maximum number of clauses in φ. If φ is of the form X-CNF, for X $\in \{k$, E$k\}$, then we get the corresponding MAX-XSAT problems.

Exercise 7. Show that the problem of deciding if a 2-CNF formula is satisfiable is in P, but MAX-2SAT is NP-Hard (i.e. its decision version is **NP**-complete).

Exercise 8. State the decision version of MAX-E3SAT and show that it is **NP**-complete.

2.1 Max-E3SAT

Theorem 2.1. *There is an $8/7$-approximation algorithm for* MAX-E3SAT.

Proof. Let φ be an E3-CNF formula with m clauses $C_1, \dots, C_m$. Let S_φ be the random variable counting the number of satisfied clauses of φ by randomly setting x_i independently to be TRUE with probability $1/2$. Since the probability that a clause C_j is satisfied is 7/8, by linearity of expectation $\mathrm{E}[S_\varphi] = 7m/8$. This number clearly is within a factor $7/8$ of the optimal value. Hence, this simple randomized algorithm achieves (expected) approximation ratio $8/7$. We can derandomize this algorithm by a method known as *conditional expectation*. The basic idea is as follows.

Consider a fixed $k \in [n]$. Let $a_1, \dots, a_k \in \{\text{TRUE}, \text{FALSE}\}$ be k boolean values. Let φ' be a formula obtained by setting $x_i = a_i,\ i \leq j$, and discarding all c clauses that are already satisfied. Then, it is easy to see that

$$\mathrm{E}[S_\varphi \mid x_i = a_i, 1 \leq i \leq k] = \mathrm{E}[S_{\varphi'}] + c.$$

Hence, given $a_1, \dots, a_k$ we can easily compute $\mathrm{E}[S_\varphi \mid x_i = a_i, 1 \leq i \leq k]$ in polynomial time.

Now, for $k \geq 1$, notice that

$$\begin{aligned} & \mathrm{E}[S_\varphi \mid x_i = a_i, 1 \leq i \leq k-1] \\ = \quad & \frac{1}{2}\mathrm{E}[S_\varphi \mid x_i = a_i, 1 \leq i \leq k-1,\ x_k = \text{TRUE}] + \frac{1}{2}\mathrm{E}[S_\varphi \mid x_i = a_i, 1 \leq i \leq k-1,\ x_k = \text{FALSE}] \end{aligned}$$

The larger of the two expectations on the right hand side is at least $\mathrm{E}[S_\varphi \mid x_i = a_i, 1 \leq i \leq k-1]$. Hence, we can set x_i to be TRUE or FALSE one by one, following the path that leads to the larger expectation, to eventually get a truth assignment which satisfies as many clauses as $\mathrm{E}[S_\varphi] = 7m/8$. □

2.2 Max-SAT

2.2.1 The straightforward randomized algorithm

Consider the WEIGHTED MAX-SAT problem in which a formula ϕ consists of m clauses $C_1, \dots, C_m$ weighted $w_1, \dots, w_m \in \mathbb{Z}^+$. Let $x_1, \dots, x_n$ be the variables and l_j denote the length of clause C_j. Suppose we follow the previous section and set each variable to be TRUE with probability $1/2$, and derandomized this algorithm. Then, what is the approximation ratio?

Let I_j denote the random variable indicating the event $\{C_j \text{ is satisfied}\}$, i.e.

$$I_j := \begin{cases} 1 & \text{if } C_j \text{ is satisfied} \\ 0 & \text{otherwise.} \end{cases}$$

Let S_ϕ be the cost (weight) of a random assignment and $\text{OPT}(\phi)$ be the cost of an optimal assignment, then $S_\phi = \sum_j w_j I_j$. We have

$$\mathrm{E}[S_\phi] = \sum_{j=1}^m w_j \,\text{Prob}[I_j = 1] = \sum_{j=1}^m w_j (1 - (1/2)^{l_j}) \geq \frac{1}{2}\sum_{j=1}^m w_j \geq \frac{1}{2}\text{OPT}(\phi).$$

In other words, with derandomization using the method of conditional expectation, we can get a deterministic approximation algorithm for MAX-SAT with approximation ratio 2.

Exercise 9. Consider the following algorithm for MAX-SAT: let τ be any truth assignment and τ' be its complement, i.e. $\tau'(x_i)$ is the negation of $\tau(x_i)$. Compute the cost of both τ and τ', then output the better assignment. Show that this is a 2-approximation algorithm.

Exercise 10. Let $\mathbb{F}_2 = \{0, 1\}$. Arithmetics over $\mathbb{F}_2$ is done modulo 2. Consider a system of m linear equations on n variables over $\mathbb{F}_2$. The LINEAR EQUATIONS OVER $\mathbb{F}_2$ problem is the problem of finding an assignment to variables that satisfies as many equations as possible. Give a randomized algorithm for this problem with approximation ratio 2, then derandomize it using the method of conditional expectation.

2.2.2 A randomized algorithm with a biased coin

The approximation ratio 2 as done above is not nearly as good as $8/7$ we had for MAX-3SAT. Perhaps this is due to the fact that MAX-SAT is not as symmetrical as MAX-3SAT. Thus, our "rounding probability" should not be $1/2$. This observation suggest us to set each variable to TRUE with some probability q to be determined. Due to symmetry (of a variable and its negation), we only need to consider $q \geq 1/2$ (thus $q \geq 1 - q$).

Let n_j and p_j be the number of negated variables and non-negated variables in clause C_j, then

$$\mathrm{E}[S_\phi] = \sum_{j=1}^{m} w_j (1 - q^{n_j}(1-q)^{p_j}).$$

To get a good approximation ratio, we want all the $q^{n_j}(1-q)^{p_j}$ to be as small as possible. This product is large for small clauses, especially the clauses with only one single literal. Let us consider them first.

- If singleton clauses contain no negations of variables, then it is easy to see that $q^{n_j}(1-q)^{p_j} \leq \max\{1-q, q^2\}$, for all j. To minimize the $\max$, we pick q such that $1 - q = q^2$, i.e. $q \approx 0.618$. In this case, we have
$$\mathrm{E}[S_\phi] \geq \frac{1}{q}\mathrm{OPT}(\phi).$$
(Note that this is slightly better than the ratio 2.)

- If there is no i such that both $\{x_i\}$ and $\{\bar{x}_i\}$ are clauses, then by swapping labels of some x_i and $\bar{x}_i$, we can obtain the same bound.

- The situation comes down to the case when there are x_i such that both $\{x_i\}$ and $\{\bar{x}_i\}$ are clauses. Firstly, note that two clauses of the form $\{x_i\}$ (or of the form $\{\bar{x}_i\}$) can be combined into one (whose weight is the total weight). Consequently, we can assume that x_i (and $\bar{x}_i$) does not appear in two singleton clauses. Secondly, if $\{x_i\}$ and $\{\bar{x}_i\}$ are both clauses, we can assume that the weight of the x_i-clause is at least the weight of the $\bar{x}_i$-clause, otherwise we swap x_i and $\bar{x}_i$. Thirdly, assume the rest of the singleton clauses contain only non-negated variables. Define
$$N = \{j \mid C_j = \{\bar{x}_i\}, \text{ for some } i\}.$$
Then,
$$\mathrm{OPT}(\phi) \leq \sum_{j=1}^{m} w_j - \sum_{j \in N} w_j.$$
And,
$$\mathrm{E}[S_\phi] = \sum_{j \notin N} w_j(1 - q^{n_j}(1-q)^{p_j}) + \sum_{j \in N} w_j(1-q) \geq q \sum_{j=1}^{m} w_j - q \sum_{j \in N} w_j \geq q \cdot \mathrm{OPT}(\phi).$$

2.2.3 A randomized algorithm with different biased coins based on linear programming

The above randomized algorithms do not deal well with small-size clauses. In this section, we make use of a linear programming formulation of the problem to determine the rounding probability of each variable.

An integer program for MAX-SAT can be obtained by considering the following 01-variables: (a) $y_i = 1$ iff $x_i = \text{TRUE}$; and (b) $z_j = 1$ iff C_j is satisfied. We then have the following integer program

$$\begin{array}{rll} \max & w_1 z_1 + \cdots + w_m z_n & \\ \text{subject to} & \displaystyle\sum_{i:x_i \in C_j} y_i + \sum_{i:\bar{x}_i \in C_j} (1 - y_i) \geq z_j, & \forall j \in [m], \\ & y_i, z_j \in \{0, 1\}, & \forall i \in [n], j \in [m] \end{array}$$

and its relaxed LP version

$$\begin{array}{rll} \max & w_1 z_1 + \cdots + w_n z_n & \\ \text{subject to} & \displaystyle\sum_{i:x_i \in C_j} y_i + \sum_{i:\bar{x}_i \in C_j} (1 - y_i) \geq z_j, & \forall j \in [m], \\ & 0 \leq y_i \leq 1 & \forall i \in [n], \\ & 0 \leq z_j \leq 1 & \forall j \in [m]. \end{array}$$

Obtain an optimal solution (y^*, z^*) for the linear program, and round $x_i = \text{TRUE}$ with probability y_i^*. Basically, the values y_i^* tells us how much x_i leans toward TRUE of FALSE. Then,

$$\begin{aligned} \mathrm{E}[S_\phi] &= \sum_{j=1}^m w_j \left(1 - \prod_{i:x_i \in C_j} (1 - y_i^*) \prod_{i:\bar{x}_i \in C_j} y_i^* \right) \\ &\geq \sum_{j=1}^m w_j \left(1 - \left[\frac{\displaystyle\sum_{i:x_i \in C_j} (1 - y_i^*) + \sum_{i:\bar{x}_i \in C_j} y_i^*}{l_j} \right]^{l_j} \right) \\ &= \sum_{j=1}^m w_j \left(1 - \left[\frac{l_j - \left(\displaystyle\sum_{i:x_i \in C_j} y_i^* + \sum_{i:\bar{x}_i \in C_j} (1 - y_i^*) \right)}{l_j} \right]^{l_j} \right) \\ &\geq \sum_{j=1}^m w_j \left(1 - \left[1 - \frac{z_j^*}{l_j} \right]^{l_j} \right) \\ &\geq \sum_{j=1}^m w_j \left(1 - \left[1 - \frac{1}{l_j} \right]^{l_j} \right) z_j^* \\ &\geq \min_j \left(1 - \left[1 - \frac{1}{l_j} \right]^{l_j} \right) \sum_{j=1}^m w_j z_j^* \\ &\geq \left(1 - \frac{1}{e} \right) \text{OPT}(\phi). \end{aligned}$$

(We have used the fact that the function $f(x) = (1 - (1 - x/l_j)^{l_j}$ is concave when $x \in [0, 1]$, thus it lies above the segment through the end points.) We have just proved

Theorem 2.2. *The LP-based randomized rounding algorithm above has approximation ratio* $e/(e-1)$.

Note that $e/(e-1) \approx 1.58$, while $1/q \approx 1/0.618 \approx 1.62$. Thus, this new algorithm is slightly better than the one with a biased coin.

Exercise 11. Describe how to use the method of conditional expectation to derandomize the algorithm above.

Exercise 12. Let $g(y)$ be any function such that $1 - 4^{-y} \leq g(y) \leq 4^{y-1}, \forall y \in [0,1]$. Suppose we set each $x_i = \text{TRUE}$ with probability $g(y_i^*)$, where (y^*, z^*) is an optimal solution to the linear program. Show that this strategy gives a $4/3$-approximation algorithm for MAX-SAT.

2.2.4 The "best-of-two" algorithm

Note that the rounding algorithm in the previous section works fairly well if clauses are of small sizes. For instance, if $l_j \leq 2$ for all j, then the approximation ratio would have been $1/(1-(1-1/2)^2) = 4/3$. On the other hand, the straightforward randomized algorithm works better when clauses are large. It just makes sense to now combine the two: run both algorithms and report the better assignment. Let S_ϕ^1 and S_ϕ^2 (which are random variables) denote the corresponding costs. Then, it is easy to see the following

$$\begin{aligned}
\mathrm{E}[\max\{S_\phi^1, S_\phi^2\}] &\geq \mathrm{E}[(S_\phi^1 + S_\phi^2)/2] \\
&\geq \sum_{j=1}^m w_j \left(\frac{1}{2}\left(1 - \frac{1}{2^{l_j}}\right) + \frac{1}{2}\left(1 - \left[1 - \frac{1}{l_j}\right]^{l_j}\right) z_j^* \right) \\
&\geq \frac{3}{4} \sum_{j=1}^m w_j z_j^* \\
&\geq \frac{3}{4} \text{OPT}(\phi).
\end{aligned}$$

Thus, the BEST-OF-TWO algorithm has performance ratio $4/3$.

3 Covering Problems

In the WSC problem, we are given a collection $\mathcal{S} = \{S_1, \dots, S_n\}$ of subsets of $[m] = \{1, \dots, m\}$, where S_j is of weight $w_j \in \mathbb{Z}^+$. The objective is to find a sub-collection $\mathcal{C} = \{S_i \mid i \in J\}$ with least total weight such that $\bigcup_{i \in J} S_i = [m]$. The corresponding integer program is

$$\begin{aligned}
\min \quad & w_1 x_1 + \cdots + w_n x_n \\
\text{subject to} \quad & \sum_{j : S_j \ni i} x_j \geq 1, \quad \forall i \in [m], \\
& x_j \in \{0,1\}, \quad \forall j \in [n].
\end{aligned}$$

And, relaxation gives the following linear program:

$$\begin{aligned}
\min \quad & w_1 x_1 + \cdots + w_n x_n \\
\text{subject to} \quad & \sum_{j : S_j \ni i} x_j \geq 1, \quad \forall i \in [m], \\
& 0 \leq x_j \leq 1 \quad \forall j \in [n].
\end{aligned}$$

Suppose we have an optimal solution $\mathbf{x}^*$ of the LP. To obtain $\mathbf{x}^A$, a sensible rounding strategy is to round x_j^* to 1 with probability x_j^*, namely

$$\text{Prob}[x_j^A = 1] = x_j^*.$$

It follows that

$$\mathrm{E}[\mathrm{cost}(\mathbf{x}^A)] = \sum_{j=1}^{n} w_j x_j^* = \textsc{opt}(LP).$$

What we really want is to find the probability that $\mathbf{x}^A$ is feasible and $\mathrm{cost}(\mathbf{x}^A) \leq \rho \cdot \textsc{opt}$. If this probability at least some positive constant, then ρ is an approximation ratio of this algorithm. (If the probability is small, we can run the algorithm independently a few times.) We can estimate the desired probability as follows.

$$\begin{aligned}
& \mathrm{Prob}[\ \mathbf{x}^A \text{ is feasible and } \mathrm{cost}(\mathbf{x}^A) \leq \rho \cdot \textsc{opt}] \\
= \ & 1 - \mathrm{Prob}[\ \mathbf{x}^A \text{ is not feasible } \textbf{or } \mathrm{cost}(\mathbf{x}^A) > \rho \cdot \textsc{opt}] \\
\geq \ & 1 - \mathrm{Prob}[\mathbf{x}^A \text{ is not feasible}] - \mathrm{Prob}[\mathrm{cost}(\mathbf{x}^A) > \rho \cdot \textsc{opt}].
\end{aligned}$$

Let us first estimate the probability that $\mathbf{x}^A$ is not feasible. Consider any element $i \in [m]$, and suppose the inequality constraint corresponding to i is

$$x_{j_1} + \cdots + x_{j_k} \geq 1.$$

We will refer to this as the ith constraint. Then, the probability that this constraint is not satisfied by $\mathbf{x}^A$ is

$$(1 - x_{j_1}^*) \ldots (1 - x_{j_k}^*) \leq \left(\frac{k - (x_{j_1}^* + \cdots + x_{j_k}^*)}{k} \right)^k \leq \left(1 - \frac{1}{k}\right)^k \leq \frac{1}{e}.$$

Thus, $\mathrm{Prob}[\mathbf{x}^A \text{ is not feasible}] \leq m/e$. This is a very bad bound since m is large. We can get a better bound by setting x_j^A to be 0 with lower probability. Let t be a number to be determined, and set $x_j^A = 0$ probability $(1 - x_j^*)^t$. (This is equivalent to running the previous strategy independently t rounds, and set $x_j^A = 0$ only when $x_j^A = 0$ in all rounds.) In this case,

$$\mathrm{Prob}[\mathbf{x}^A \text{ does not satisfy constraint } i] \leq (1/e)^t.$$

Thus, the probability that $\mathbf{x}^A$ is not a feasible solution is at most $m(1/e)^t$. When t is (logarithmically) large, $m(1/e)^t < 1$.

Secondly, we estimate the probability that $\mathrm{cost}(\mathbf{x}^A) > \rho \cdot \textsc{opt}$. In one round, we have shown that $\mathrm{E}[\mathrm{cost}(\mathbf{x}^A)] = \textsc{opt}(LP) \leq \textsc{opt}$. Hence, with t rounds we have $\mathrm{E}[\mathrm{cost}(\mathbf{x}^A)] \leq t \cdot \textsc{opt}$. Markov inequality gives

$$\mathrm{Prob}[\mathrm{cost}(\mathbf{x}^A) > \rho \cdot \textsc{opt}] < \frac{\mathrm{E}[\mathrm{cost}(\mathbf{x}^A)]}{\rho \cdot \textsc{opt}} \leq \frac{t \cdot \textsc{opt}}{\rho \cdot \textsc{opt}} = \frac{t}{\rho}.$$

Remark 3.1. Let X be a random variable in $\mathbb{R}^+$, and a be a positive number, Markov inequality says that $\mathrm{Prob}[X \geq a] \leq \frac{\mathrm{E}[X]}{a}$.

Consequently,

$$\mathrm{Prob}[\mathbf{x}^A \text{ is feasible and } \ \mathrm{cost}(\mathbf{x}^A) \leq \rho \cdot \textsc{opt}(IP)] \geq 1 - m(1/e)^t - \frac{t}{\rho}.$$

We can pick $t = \theta(\lg m)$ and $\rho = 4t$ so that $1 - m(1/e)^t - \frac{t}{\rho} \geq \frac{1}{2}$. In other words, this algorithm gives a solution with approximation ratio $\Theta(\lg m)$ with probability at least $1/2$. We can then run the algorithm a few times until the solution is feasible. The expected number of runs is 2, and the expected approximation ratio is $\Theta(\lg m)$.

Exercise 13. Suppose we run the above randomized rounding algorithm with only one round (instead of t rounds). Prove that, with positive probability the resulting $\mathbf{x}^A$ satisfies at least half of the constraints at cost at most $O(\textsc{opt}(IP))$.

Exercise 14. Give a randomized rounding algorithm for the GENERAL COVER problem with approximation ratio $O(\lg m)$.

Appendix

3.1 Probability theory

Lemma 3.2 (Linearity of Expectation). *If $X_1, \ldots, X_n$ are n random variables, then for any n constants $a_1, \ldots, a_n$,*

$$E\big[a_1 X_1 + \cdots + a_n X_n\big] = a_1 \mathrm{E}[X_1] + \cdots + a_n \mathrm{E}[X_n].$$

Many deviation bounds are useful in designing randomized algorithms.

Theorem 3.3 (Markov's Inequality). *If X is a random variable taking only non-negative values, then for any $a > 0$*

$$\mathrm{Prob}[X \geq a] \leq \frac{\mathrm{E}[X]}{a}. \tag{6}$$

A slightly more intuitive form of (6) is

$$\mathrm{Prob}[X \geq a\mu] \leq \frac{1}{a}. \tag{7}$$

Markov's inequality is possibly the only possible estimate when there's no further information about the random variable. If we do now its variance, for instance, we can show stronger bound.

Theorem 3.4 (Chebyshev's Inequality). *If X is a random variable with mean μ and variance σ^2, then for any $a > 0$,*

$$\mathrm{Prob}\big[|X - \mu| \geq a\big] \leq \frac{\sigma^2}{a^2}. \tag{8}$$

Again, there is a more intuitive way of writing (8):

$$\mathrm{Prob}\big[|X - \mu| \geq a\sigma\big] \leq \frac{1}{a^2}. \tag{9}$$

A twice-differentiable function f is *convex* if $f''(x) \geq 0$ for all x, and *concave* when $f''(x) \geq 0$ for all x. A linear function is both convex and concave. Thus, the following theorem implies linearity of expectation.

Theorem 3.5 (Jensen's inequality). *Let $f(x)$ be a convex function, then*

$$\mathrm{E}[f(X)] \geq f(E[X]). \tag{10}$$

If f is concave, the inequality is reversed. The same result holds for multiple random variables.

The most useful deviation bounds are variations of Chernoff bounds.

Theorem 3.6 (Chernoff Bound (Lower Tail)). *Let $X_1, \ldots, X_n$ be a set of mutually independent Bernulli random variables, where $\mathrm{Prob}[X_i = 1] = p_i$, and $\mathrm{Prob}[X_i = 0] = 1 - p_i$, for $0 < p_i < 1$. Let $S_n = X_1 + \cdots + X_n$, and $\mu = \mathrm{E}[S_n] = p_1 + \cdots + p_n$. Then, for any $0 < \epsilon < 1$,*

$$\mathrm{Prob}[S < (1-\epsilon)\mu] < \left(\frac{e^{-\epsilon}}{(1-\epsilon)^{1-\epsilon}}\right)^{\mu} < e^{-\mu\epsilon^2/2} \tag{11}$$

Theorem 3.7 (Chernoff Bound (Upper Tail)). *Let $X_1, \ldots, X_n$ be a set of mutually independent Bernulli random variables, where $\mathrm{Prob}[X_i = 1] = p_i$, and $\mathrm{Prob}[X_i = 0] = 1 - p_i$, for $0 < p_i < 1$. Let $S_n = X_1 + \cdots + X_n$, and $\mu = \mathrm{E}[S_n] = p_1 + \cdots + p_n$. Then,*

1. *for any* $\epsilon > 0$,
$$\text{Prob}[S > (1+\epsilon)\mu] < \left(\frac{e^{+\epsilon}}{(1+\epsilon)^{1+\epsilon}}\right)^{\mu}. \tag{12}$$

2. *for* $\epsilon > 2e - 1$,
$$\text{Prob}[S > (1+\epsilon)\mu] < 2^{-\epsilon\mu}, \tag{13}$$

3. *for* $0 < \epsilon < 2e - 1$,
$$\text{Prob}[S > (1+\epsilon)\mu] < e^{-\mu\epsilon^2/4}, \tag{14}$$

4. *and lastly, for* $0 < \epsilon < 1$,
$$\text{Prob}[S > (1+\epsilon)\mu] < e^{-\mu\epsilon^2/3}. \tag{15}$$

3.2 The Probabilistic Method

The basic idea of the probabilistic method is the following: to show that some object with a certain property exists, under suitable settings we can just show that it exists with positive probability. The classic reference [1] is now a "must-read" for Computer Science students.

To make this idea a little more precise, consider a finite probability space Ω. To show that there is a member ω of Ω having some property P, we only have to show that Prob$[\omega$ has property $P] > 0$.

To illustrate this idea, consider a tennis tournament T where there are n players, each player plays every other player, and the matches' results are already recorded. Thus, there are totally $2^{\binom{n}{2}}$ possible tournaments. Tournament T is said to have property P_k if for every set of k players there is another player who beats them all. We will prove that, if $\binom{n}{k}(1-2^{-k})^{n-k} < 1$, then there is a tournament on n players having property S_k.

Let Ω be the set of all $2^{\binom{n}{2}}$ tournaments, where we choose a random tournament by letting player i beat player j with probability $1/2$. Consider a randomly chosen tournament T from Ω. We want to estimate the probability that T has property S_k. For any subset K of k players, let A_K be the event that no other player beats all members of K. Tournament T has property S_k iff A_K does not hold, for every K. The probability that a particular player (not in K) does not beat all members of K is $(1-2^{-k})$. Hence, Prob$[A_K] = (1-2^{-k})^{n-k}$. Consequently,

$$\text{Prob}\left[\bigcup_K A_K\right] \leq \sum_K \text{Prob}[A_K] = \binom{n}{k}(1-2^{-k})^{n-k} < 1.$$

In other words, the probability that none of the events A_K occurs is positive, concluding our proof.

Another technique that is very commonly used in the probabilistic method is the following idea. Let X be any real random variable on Ω, i.e. $X : \Omega \to \mathbb{R}$. Let $\mu = \text{E}[X]$. Then, there must be an ω with $X(\omega) \leq \mu$, and similarly there must be an ω with $X(\omega) \geq \mu$. Again consider a tournament T on n players as defined above. A *Hamiltonian circuit* on this tournament is a permutation π of players, where $\pi(i)$ beats $\pi(i+1)$, for all i, circularly. Let Ω be the probability space of all random tournaments. For each $T \in \Omega$, let $X(T)$ be the number of Hamiltonian circuits of T. For each permutation π of n players, let I_π be the random variable indicating if π defines a Hamiltonian circuit on T, namely

$$I_\pi = \begin{cases} 1 & \text{if } \pi(i) \text{ beats } \pi(i+1), \text{ circularly, for all } i \\ 0 & \text{otherwise.} \end{cases}$$

Then,

$$\text{E}[I_\pi] = \text{Prob}[I_\pi = 1] = \frac{1}{2^n}.$$

Thus, by linearity of expectation,

$$\mathrm{E}[X] = \mathrm{E}\left[\sum_{\pi} I_{\pi}\right] = \sum_{\pi} \mathrm{E}[I_{\pi}] = \frac{n!}{2^n}.$$

We can now conclude that there is at least one tournament T with at least $\frac{n!}{2^n}$ Hamiltonian circuits, and there is at least one tournament T with at most $\frac{n!}{2^n}$ Hamiltonian circuits.

3.3 Inequalities

In algorithm analysis, when upper or lower bounding an expression we often need to "turn" a sum into a product or vice versa. In that case, the following standard inequality is extremely useful.

Theorem 3.8 (Arithmetic-Geometric means inequality). *For any non-negative numbers $a_1, \ldots, a_n$, we have*

$$\frac{a_1 + \cdots + a_n}{n} \geq (a_1 \cdots a_n)^{1/n}. \tag{16}$$

There is also the stronger weighted version. Let $w_1, \ldots, w_n$ be positive real numbers where $w_1 + \cdots + w_n = 1$, then

$$w_1 a_1 + \cdots + w_n a_n \geq a_1^{w_1} \cdots a_n^{w_n}. \tag{17}$$

Equality holds iff all a_i are equal.

Talking about classic inequalities, one cannot ignore Cauchy-Schwarz and Jensen inequalities.

Theorem 3.9 (Cauchy-Schwarz inequality). *Let $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$ be non-negative real numbers. Then,*

$$\left(\sum_{i=1}^{n} a_i b_i\right)^2 \leq \left(\sum_{i=1}^{n} a_i^2\right)\left(\sum_{i=1}^{n} b_i^2\right). \tag{18}$$

Theorem 3.10 (Jensen inequality). *Let $f(x)$ be a convex function on an interval (a, b). Let $x_1, \ldots, x_n$ be points in (a, b), and $w_1, \ldots, w_n$ be non-negative weights such that $w_1 + \cdots + w_n = 1$. Then,*

$$f\left(\sum_{i=1}^{n} w_i x_i\right) \leq \sum_{i=1}^{n} w_i f(x_i). \tag{19}$$

If f is strictly convex and if all weights are positive, then equality holds iff all x_i are equal. When f is concave, the inequality is reversed.

Historical Notes

Recent books on approximation algorithms include [2, 7, 11, 13]. See [1, 12] for randomized algorithms, derandomization and the probabilistic methods. For inequalities, the classic text [5] is a must-have.

The $8/7$-approximation algorithm for MAX-E3SAT follows the line of Yannakakis [14], who gave the first $4/3$-approximation for MAX-SAT. A 2-approximation for MAX-SAT was given in the seminal early work of Johnson [8]. Johnson's algorithm can also be interpreted as a derandomized algorithm, mostly the same as the one we presented. The LP-based randomized algorithm and the best-of-two algorithm for MAX-SAT are due to Goemans and Williamson [4]. The algorithm with a biased coin is due to Lieberherr and Specker [10].

Later, Karloff and Zwick [9] gave an $8/7$-approximation algorithm for MAX-3SAT based on semidefinite programming. This approximation ratio is optimal as shown by Håstad [6]. The conditional expectation method was implicit in Erdős and Selfridge [3].

References

[1] N. ALON AND J. H. SPENCER, *The probabilistic method*, Wiley-Interscience Series in Discrete Mathematics and Optimization, Wiley-Interscience [John Wiley & Sons], New York, second ed., 2000. With an appendix on the life and work of Paul Erdős.

[2] G. AUSIELLO, P. CRESCENZI, G. GAMBOSI, V. KANN, A. MARCHETTI-SPACCAMELA, AND M. PROTASI, *Complexity and approximation*, Springer-Verlag, Berlin, 1999. Combinatorial optimization problems and their approximability properties, With 1 CD-ROM (Windows and UNIX).

[3] P. ERDŐS AND J. L. SELFRIDGE, *On a combinatorial game*, J. Combinatorial Theory Ser. A, 14 (1973), pp. 298–301.

[4] M. X. GOEMANS AND D. P. WILLIAMSON, *New $\frac{3}{4}$-approximation algorithms for the maximum satisfiability problem*, SIAM J. Discrete Math., 7 (1994), pp. 656–666.

[5] G. H. HARDY, J. E. LITTLEWOOD, AND G. PÓLYA, *Inequalities*, Cambridge University Press, Cambridge, 1988. Reprint of the 1952 edition.

[6] J. HÅSTAD, *Some optimal inapproximability results*, in STOC '97 (El Paso, TX), ACM, New York, 1999, pp. 1–10 (electronic).

[7] D. S. HOCHBAUM, ed., *Approximation Algorithms for NP Hard Problems*, PWS Publishing Company, Boston, MA, 1997.

[8] D. S. JOHNSON, *Approximation algorithms for combinatorial problems*, J. Comput. System Sci., 9 (1974), pp. 256–278. Fifth Annual ACM Symposium on the Theory of Computing (Austin, Tex., 1973).

[9] H. KARLOFF AND U. ZWICK, *A 7/8-approximation algorithm for MAX 3SAT?*, in Proceedings of the 38th Annual IEEE Symposium on Foundations of Computer Science, Miami Beach, FL, USA, IEEE Press, 1997.

[10] K. J. LIEBERHERR AND E. SPECKER, *Complexity of partial satisfaction*, J. Assoc. Comput. Mach., 28 (1981), pp. 411–421.

[11] E. W. MAYR AND H. J. PRÖMEL, eds., *Lectures on proof verification and approximation algorithms*, vol. 1367 of Lecture Notes in Computer Science, Springer-Verlag, Berlin, 1998. Papers from the Workshop on Proof Verification and Approximation Algorithms held at Schloß Dagstuhl, April 21–25, 1997.

[12] R. MOTWANI AND P. RAGHAVAN, *Randomized algorithms*, Cambridge University Press, Cambridge, 1995.

[13] V. V. VAZIRANI, *Approximation algorithms*, Springer-Verlag, Berlin, 2001.

[14] M. YANNAKAKIS, *On the approximation of maximum satisfiability*, J. Algorithms, 17 (1994), pp. 475–502. Third Annual ACM-SIAM Symposium on Discrete Algorithms (Orlando, FL, 1992).