Advanced Heuristic and Approximation Algorithms Grenoble-INP (ORCO)

Lecture 3

Lecturer: Alantha Newman *October 16, 2017*

1 Linear Programming Background

Our next goal is to present algorithms for the *maximum cut problem* (max-cut). Given an undirected, unweighted graph $G = (V, E)$, the max-cut problem is to find a proper, nonempty subset of vertices $S \subset V$ such that $|E(S, V \setminus S)|$ is maximized. Recall that an edge $ij \in E$ belongs to $E(S, V \setminus S)$ if one endpoint belongs to S and the other endpoint belongs to the complement of S, namely $\bar{S} = V \setminus S$.

Since max-cut is NP-hard, we cannot hope to obtain an algorithm that ouputs the optimal solution with high probability, as we saw for min-cut. A simple $\frac{1}{2}$-approximation algorithm for max-cut is to assign each vertex to S with probability $\frac{1}{2}$. There are no known *simple* combinatorial algorithms for max-cut with an approximation ratio greater than $\frac{1}{2}$. Therefore, we turn to the *mathematical programming* toolbox for help.

1.1 Linear Program for s-t-Min-Cut

First, we consider linear programming. We will show how to solve the s-t-min-cut using a linear program. We introduce the following variables, which correspond to edges and vertices:

- $\ell : E \rightarrow \mathbb{R}$, where $\ell(i, j)$ is the *length* of an edge ij,
- $p : V \rightarrow \mathbb{R}$, where $p(i)$ is the *potential* of a vertex i.

$$
\begin{aligned}
& \min \sum_{ij \in E} \ell(ij) \\
\text{subject to:} \quad & \ell(ij) \geq p(i) - p(j), \quad ij \in E, \\
& p(s) = 1, \\
& p(t) = 0 \\
& \ell(ij) \geq 0, \quad ij \in E. \qquad (P_{min-cut})
\end{aligned}
$$

Lemma 1. *The optimal value of $(P_{min-cut})$ on the graph G equals the value of the minimum s-t-cut for the graph G.*

Proof. Suppose G has a minimum s-t-cut with value k. This means that G is k-edge connected. (Recall that a graph is k-edge-connected if any set of $k - 1$ edges does not disconnect the graph.)

$\Rightarrow$ Assign all vertices in S value 1 and all those in $\bar{S}$ value 0. This shows that the solution to $(P_{min-cut})$ is at most k.

$\Leftarrow$ By Menger's Theorem, there exist k edge-disjoint paths between s and t. Each path has value at least 1. Thus, the solution to $(P_{min-cut})$ is at least k. $\square$

Exercise 1. *Given an optimal (possibly fractional) solution to $(P_{min-cut})$ on a graph G, find an optimal (integral) minimum cut for G.*

1.2 Formal Description of a Linear Program

Let n be a positive integer, let $\mathbf{a} \in \mathbb{R}^n$ be a vector and let $b \in \mathbb{R}$ be a scalar.

Definition 2. *The set $\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{a}^\intercal \mathbf{x} = b\}$ is called a* hyperplane.

Definition 3. *The set* $\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{a}^\intercal \mathbf{x} \geq b\}$ *is called a* halfspace.

Note that a halfspace consists of all points that are one side of a hyperplane.

Definition 4. *A* polyhedron *is the intersection of a finite number of halfspaces.*

A *linear program* is the problem of optimizing a linear objective function over the points in a polyhedron. For instance, let $\mathbf{c} \in \mathbb{R}^n$ be a vector. Then the following is a (generic) linear program.

$$\begin{aligned} \min \quad & \mathbf{c}^\intercal \mathbf{x} \\ \text{subject to:} \quad & \mathbf{A}\mathbf{x} \geq \mathbf{b}. \end{aligned} \tag{P}$$

We often require that the variables are non-negative and write a linear program as follows.

$$\begin{aligned} \min \quad & \mathbf{c}^\intercal \mathbf{x} \\ \text{subject to:} \quad & \mathbf{A}\mathbf{x} \geq \mathbf{b} \\ & \mathbf{x} \geq 0. \end{aligned} \tag{P}$$

An optimal solution to (P) is a point $\mathbf{y} \in P$ such that the objective value, $\mathbf{c}^\intercal \mathbf{y}$, is minimum.

Note that writing an integer program (and consequently a linear programming relaxation) for a discrete optimization problem (e.g. min-cut, max-flow, etc.) is not always obvious or easy. We saw a linear program for s-t-min-cut. Writing a linear program for global min-cut is harder.

1.3 Integrality Gap

One way of measuring the quality of a linear programming relaxation is known as the *integrality gap*. Given a set $\mathcal{I}$ of instances for a minimization problem, let $OPT(I)$ denote the optimal (integral) solution of a fixed integer program on instance I, and let $OPT_f(I)$ denote the optimal fractional solution for the linear programming relaxation of this integer program. Then the integrality gap of the linear programming relaxation is defined as:

$$\sup_{I \in \mathcal{I}} \frac{OPT(I)}{OPT_f(I)}.$$

For a maximization problem, it is defined as:

$$\inf_{I \in \mathcal{I}} \frac{OPT(I)}{OPT_f(I)}.$$

The integrality gap establishes a limit on the usefulness of a linear programming relaxation. Specifically, if our approximation algorithm is solely based on a linear program, then we cannot obtain an approximation ratio better than the integrality gap of this linear program. Note that for the s-t-min-cut problem, the integrality gap of $(P_{min-cut})$ is 1.

We now look at examples of integer programs for some NP-hard discrete optimization problems and their respective linear programming relaxations.

2 Vertex Cover

Given a graph $G = (V, E)$, the *vertex cover problem* is to find a subset of vertices $S \subset V$ such that for each edge $ij \in E$, at least one endpoint belongs to S. In other words, either i or j belongs to S or, alternatively, both i and j belong to S. The vertex cover is NP-hard. We now present a 2-approximation algorithm. Let OPT_{VC} denote that size of a minimum vertex cover for the graph G.

2.1 2-Approximation Algorithm

Since vertex cover is a minimization problem, we first need to find a (preferably large) lower bound. A subset of edges $M \subseteq E$ is a *matching* in G if all edges are vertex disjoint.

Lemma 5. *Let $M \subseteq E$ be a matching in G. Then $|M|$ is a lower bound on the size of a minimum vertex cover in G.*

Proof. Any matching is a set of vertex-disjoint edges. Thus, at least one endpoint of each edge belongs to a vertex cover. Otherwise, there is some edge that is not covered. □

A subset of edge $M \subseteq E$ is a *maximal matching* for G if for every edge ij in E, either i or j (or both) are endpoints of some edge in M. In other words, there are no edges that can be added to M to increase the size (cardinality) of M.

VERTEX-COVER-MATCHING(G)

1. $S \leftarrow \emptyset$.
2. Find a maximal matching M in G.
3. For each edge $ij \in M$:
   (a) Add i to S.
   (b) Add j to S.
4. Output the set of vertices S.

Theorem 6. VERTEX-COVER-MATCHING(G) *is a 2-approximation algorithm.*

Proof. First, we show that the set of vertices output by the algorithm, S, is a valid vertex cover. Next, we show that the size of S is at most twice the size of a minimum vertex cover.

Suppose S is not a valid vertex cover. Then there exists an edge $ij \in E$ such that neither i nor j belong to S. This implies that neither i nor j belong to M, which contradicts the assumption that M is a *maximal* matching.

By Lemma 5, we know that $|M|$ is a lower bound on the size of minimum vertex cover. Recall that OPT_{VC} denotes that size of a minimum vertex cover for the graph G. Then,

$$|M| \leq OPT_{VC}.$$

It follows that

$$|S| = 2|M| \leq 2 \cdot OPT_{VC}.$$

Thus, we have proved the theorem. □

2.2 Linear Programming Relaxation

The vertex cover problem can be formulated as the following integer program.

$$\begin{aligned} \min \sum_{i \in V} x_i & \\ \text{subject to:}\quad x_i + x_j \geq 1, &\quad \text{for all } ij \in E, \\ x_i \in \{0,1\}. & \end{aligned} \tag{IP$_{VC}$}$$

Consider the linear programming relaxation of the integer program (IP$_{VC}$).

$$\begin{aligned} \min \sum_{i \in V} x_i & \\ \text{subject to:}\quad x_i + x_j \geq 1, &\quad \text{for all } ij \in E, \\ 0 \leq x_i \leq 1. & \end{aligned} \tag{LP$_{VC}$}$$

VERTEX-COVER-LP(G)

1. $S \leftarrow \emptyset$.
2. Find an optimal solution for (LP$_{VC}$).
3. For each vertex $i \in V$:
   (a) If $x_i \geq \frac{1}{2}$, add i to S.
4. Output the set of vertices S.

Theorem 7. VERTEX-COVER-LP(G) *is a 2-approximation algorithm.*

Proof. Let $\mathbf{x}^*$ denote an optimal solution for (LP$_{VC}$) and let $OPT_f(G)$ denote the objective value of $\mathbf{x}^*$. Let $S = \{i \in V \mid x_i^* \geq \frac{1}{2}\}$. Note that S is feasible since for each edge $ij \in E$, the constraints from (LP$_{VC}$) imply that either x_i^* or x_j^* (or both) have value at least $\frac{1}{2}$.

How large can $|S|$ be? Each vertex $i \in V$ for which $x_i^* < \frac{1}{2}$ contributes zero to $|S|$, and each vertex for which $x_i^* \geq \frac{1}{2}$ contributes one vertex to $|S|$. Therefore, $|S| \leq 2 \cdot \sum_{i \in V} x_i^*$. So we have,

$$|S| \;\leq\; 2 \cdot \sum_{i \in V} x_i^* \;=\; 2 \cdot OPT_f(G) \;\leq\; OPT_{VC}.$$

the size of S is no more than twice that of a minimum vertex cover. □

For general graphs, we cannot hope to approximate the vertex cover problem to within a factor better than 2 using just this linear program.

Theorem 8. *The integrality gap of* (LP$_{VC}$) *is at least* $2 - \frac{2}{n}$.

Proof. Consider the complete graph K_n on n vertices. If we set $x_i = \frac{1}{2}$ for all $i \in V$, then the value of $OPT_f(K_n)$ is $\frac{n}{2}$. However, the size of a minimum vertex cover in K_n is $n-1$. Thus, the integrality gap is at least $\frac{n-1}{n/2} = 2 - \frac{2}{n}$, i.e. arbitrarily close to 2. □

3 Max-Cut

Now we consider integer and linear programs for the maximum cut problem. The following integer program sets the variable $x_i = 0$ if vertex i is assigned to the left side of the cut and $x_i = 1$ if vertex i is assigned to the right side. It sets the variable $z_{ij} = 1$ if vertex i is on the left size of the cut and vertex j is on the right side.

$$\begin{aligned} \min \sum_{ij \in E} (z_{ij} + z_{ji}) & \\ \text{subject to: } z_{ij} \leq x_i, \quad & \text{for all } ij \in E, \\ z_{ij} \leq 1 - x_j, \quad & \text{for all } ij \in E, \\ x_i, z_{ij} \in \{0, 1\}. & \end{aligned} \tag{IP$^1_{MC}$}$$

Alternatively, we can use the variable $y_{ij} = 1$ to denote that edge ij crosses the cut, i.e $y_{ij} = z_{ij} + z_{ji}$. Then we have,

$$\begin{aligned} \min \sum_{ij \in E} y_{ij} & \\ \text{subject to: } y_{ij} \leq x_i + y_j, \quad & \text{for all } ij \in E, \\ y_{ij} \leq 2 - x_j - x_j, \quad & \text{for all } ij \in E, \\ x_i, y_{ij} \in \{0, 1\}. & \end{aligned} \tag{IP$^2_{MC}$}$$

Now consider the linear programming relaxaiton of (IP$^2_{MC}$).

$$\begin{aligned} \min \sum_{ij \in E} y_{ij} & \\ \text{subject to: } y_{ij} \leq x_i + y_j, \quad & \text{for all } ij \in E, \\ y_{ij} \leq 2 - x_j - x_j, \quad & \text{for all } ij \in E, \\ 0 \leq x_i, y_{ij} \leq 1. & \end{aligned} \tag{LP$^2_{MC}$}$$

What is the integrality gap of (LP$^2_{MC}$)? By setting $x_i = \frac{1}{2}$ for each vertex $i \in V$, we see that $y_{ij} = 1$ for each $ij \in E$ is a feasible solution. Thus, the value of (LP$^2_{MC}$) is $|E|$ for any graph! So this linear program does not give us any new information. Note that the complete graph K_n has a maximum cut that, asymptotically, has size equal to half the edges. We can conclude that the integrality gap of (LP$^2_{MC}$) is 2.

3.1 Cycle Constraint Relaxation

The following linear program is based on the fact that an odd cycle C can contribute at most $|C| - 1$ edges to a (maximum) cut.

$$\begin{aligned} & \max \sum_{ij \in E} x_{ij} \\ & \sum_{ij \in C} x_{ij} \leq |C| - 1, \quad \text{for all odd cycles } C \in E, \\ & 0 \leq x_{ij} \leq 1, \quad \text{for all edges } ij \text{ in } E. \end{aligned} \tag{LP$_{odd-cycle}$}$$

The integrality gap for (LP$_{odd-cycle}$) is also arbitrarily close to 2. In particular, Poljak proved the following theorem using the probabilistic method [Pol91]. Note that the *girth* of a graph G is the length of the shortest cycle in G.

Theorem 9. *Let g be a positive integer and $\epsilon > 0$ be a (small) fixed constant. Then there is a sufficiently large integer N_0 such that for any $n \geq N_0$, there exists a graph on n vertices with girth g and maximum cut at most $(\frac{1}{2} + \epsilon)|E|$.*

Consider a graph $G = (V, E)$ with girth g and maximum cut at most $(\frac{1}{2} + \epsilon)|E|$. Setting $x_{ij} = 1 - \frac{1}{g}$ for all edges $ij \in E$ implies that the objective value of $(\text{LP}_{odd-cycle})$ is at least $(1 - \frac{1}{g})|E|$. Therefore, the integrality gap of $(\text{LP}_{odd-cycle})$ is at least:

$$\frac{(1 - \frac{1}{g})}{(\frac{1}{2} + \epsilon)}.$$

Since g can be chosen to be arbitrarily large and ϵ can be chosen to be arbitrarily close to zero, this quantity can be arbitrarily close to 2.

3.2 Polynomial-Size Cycle Constraint Relaxation

There is another linear program that gives the same upper bound on the value of a maximum cut as that given by $(\text{LP}_{odd-cycle})$. Here, we have a variable x_{ij} for every pair of vertices i, j in V, even if there is no edge ij in E. This linear program is based on the fact that for a triangle i, j, k in V, either two edges or no edges can cross a cut. For example, neither $x_{ij} = x_{ik} = x_{jk} = 1$ nor $x_{ij} = 1, x_{ik} = x_{jk} = 0$ are feasible solutions for the following linear program.

$$\begin{aligned} \max \sum_{ij \in E} x_{ij} & \\ x_{ij} + x_{jk} + x_{ik} \leq 2, \quad & \text{for all } i, j, k \in V, \\ x_{ij} - x_{jk} - x_{ik} \geq 0, \quad & \text{for all } i, j, k \in V, \\ -x_{ij} - x_{jk} + x_{ik} \geq 0, \quad & \text{for all } i, j, k \in V, \\ -x_{ij} + x_{jk} - x_{ik} \geq 0, \quad & \text{for all } i, j, k \in V, \\ 0 \leq x_{ij} \leq 1, \quad & \text{for all edges } ij \text{ in } E. \end{aligned} \qquad (\text{LP}_{triangle})$$

Since $(\text{LP}_{triangle})$ and $(\text{LP}_{odd-cycle})$ have the same optimal values, the integrality gap of $(\text{LP}_{triangle})$ is also arbitrarily close to 2.

References

[Pol91] Svatopluk Poljak. *Polyhedral and eigenvalue approximations of the max-cut problem.* Universität Bonn. Institut für Ökonometrie und Operations Research, 1991.