Advanced Heuristic and Approximation Algorithms Grenoble-INP (ORCO)

Lecture 4

Lecturer: Alantha Newman *October 22, 2018*

1 Set Cover

In the *set cover* problem we are given a universe of elements and a family of sets on these elements. The goal is to find a minimum cardinality (or cost) collection of sets such that each element occurs in at least one of the chosen sets.

Input: A universe of n elements, $U = \{e_1, e_2, \dots, e_n\}$, and a family of m sets, $\mathcal{T} = \{T_1, T_2, \dots, T_m\}$, such that $T_j \subseteq U$ for all $j \in \{1, 2, \dots, m\}$. There is a cost function $c : \mathcal{T} \rightarrow \mathbb{R}^+$ indicated by $c_j = c(T_j)$.
Output: A minimum cost subset of $\mathcal{S} \subseteq \mathcal{T}$ such that for each $e_i \in U$, there is some set $T_j \in \mathcal{S}$ such that $e_i \in T_j$. The cost of $\mathcal{S}$ is indicated by $c(\mathcal{S}) = \sum_{T_j \in \mathcal{S}} c(T_j)$.

We can model this problem with the following integer program. Let $y_j \in \{0, 1\}$ denote whether or not set T_j is included in the set cover.

$$\begin{aligned} &\min \sum_{T_j \in \mathcal{T}} c_j y_j \\ \text{subject to: } &\sum_{j : e \in T_j} y_j \geq 1, \quad \text{for all } e \in U, \\ &y_j \in \{0, 1\}. \end{aligned}$$

Now let us consider the linear programming relaxation in which y_j now denotes the fractional amount of set T_j included in the optimal fractional solution. We can assume that $y_j \leq 1$ since otherwise we can show the solution is not a minimum cost solution.

$$\begin{aligned} &\min \sum_{T_j \in \mathcal{T}} c_j y_j \\ \text{subject to: } &\sum_{j : e \in T_j} y_j \geq 1, \quad \text{for all } e \in U, \\ &y_j \geq 0. \end{aligned} \tag{$P_{set\text{-}cover}$}$$

If each element is in at most f sets, then $\mathcal{S} = \{T_j \mid y_j \geq \frac{1}{f}\}$ is a feasible set cover and choosing this set cover is f-approximation algorithm. However, since f could be quite large, this might not be a good approach in general.

A common approach to finding feasible solutions for such problems is to interpret the y_j variables as probabilities, i.e. the probability of adding the set T_j to the solution set. This is the main idea behind *randomized rounding* of linear programs.

1.1 Randomized Rounding for Set Cover

We now present an algorithm based on randomized rounding of a solution to the linear program ($P_{set\text{-}cover}$).

```
RANDOMIZED-ROUNDING-SET-COVER 1

Find an optimal solution $\mathbf{y}^*$ for ($P_{set\text{-}cover}$).
For $j = 1$ to $m$:
    Include each set $T_j$ in the solution set $\mathcal{S}$ with probability $y_j^*$.
```

The cost of this solution is exactly $\sum_{T_j \in \mathcal{T}} c_j y_j^*$. But this solution might not be feasible. In other words, there may be some elements in U that are not covered by any set in $\mathcal{S}$. However, we can prove that each element has a constant probability of being covered by some set.

Lemma 1. Randomized-Rounding-Set-Cover 1 *covers each element with probability at least* $(1-\frac{1}{e})$.

Proof. Consider an element e_i that belongs to k sets, each of which is selected by the algorithm with probability $p_1, p_2, \ldots, p_k$. By the constraints from $(P_{set\text{-}cover})$, we have:

$$\sum_{j=1}^{k} p_j \quad \geq \quad 1.$$

Using the fact that $1 + x < e^x$ for all $x \neq 0$, we have:

$$\begin{aligned} \Pr[e_i \text{ is not covered}] \quad &= \quad \prod_{j=1}^{k} (1 - p_j) \\ &< \quad \prod_{j=1}^{k} e^{-p_j} \quad = \quad e^{-\sum p_j} \quad \leq \quad \frac{1}{e}. \end{aligned}$$

So, $\Pr[e_i \text{ is covered}] > 1 - \frac{1}{e}$. □

Now we modify our algorithm to amplify the probability of success, i.e. the probability of covering all elements. (Note that if set T_j is actually chosen after fewer than $2 \log n$ rounds, we can break the For-loop and move to the next value of j.)

```
Randomized-Rounding-Set-Cover 2

Find an optimal solution $\mathbf{y}^*$ for $(P_{set\text{-}cover})$.
For $j = 1$ to $m$:
    Repeat $2 \log n$ times: Include each set $T_j$ in the solution set $\mathcal{S}$ with probability $y_j^*$.
```

Lemma 2. Randomized-Rounding-Set-Cover 2 *covers each element with probability at least* $1-\frac{1}{n^2}$.

Proof.

$$\begin{aligned} \Pr[e_i \text{ is not covered}] \quad &= \quad \prod_{j: e_i \in T_j} (1 - y_j^*)^{2 \log n} \\ &< \quad \prod_{j: e_i \in T_j} e^{-y_j^* 2 \log n} \\ &= \quad e^{-2 \log n \sum_{j: e_i \in T_j} y_j^*} \\ &\leq \quad \left(\frac{1}{e}\right)^{2 \log n} \quad = \quad \frac{1}{n^2}. \end{aligned}$$

So, $\Pr[e_i \text{ is covered}] > 1 - \frac{1}{n^2}$. □

Lemma 3. Randomized-Rounding-Set-Cover 2 *covers all elements with probability at least* $1 - \frac{1}{n}$.

Proof. By union bound, we can show:

$$\Pr[\text{exists } e_i \text{ that is not covered}] \quad \leq \quad \frac{1}{n^2} \cdot n \;=\; \frac{1}{n}. \qquad \text{(Bad event I)}$$

So we have:

$$\Pr[\text{all elements are covered}] \quad \geq \quad 1 - \frac{1}{n}.$$

□

We are not done, because there is some probability that even if all elements are covered, the cost exceeds the expected cost. Let $OPT_f = \sum_{T_j \in \mathcal{T}} c_j y_j^*$. By Markov's inequality, we have:

$$\Pr[c(\mathcal{S}) > 4(OPT_f \cdot 2 \log n)] \quad \leq \quad \frac{1}{4}. \qquad \text{(Bad event II)}$$

By union bound the probability that either bad events I or II can occur is at most $\frac{1}{4} + \frac{1}{n} < \frac{1}{3}$ for sufficiently large n. Thus, with constant probability, we find a solution that is (i) a valid set cover and (ii) costs at most $8 \log n \cdot OPT$. We can conclude that Randomized-Rounding-Set-Cover 2 is a $O(\log n)$-approximation algorithm. There are instances for the set cover problem exhibiting an integrality gap of $\Omega(\log n)$.

2 Dominating Set on Digraphs

Let $G = (V, A)$ be a directed graph on n vertices. We say that $(i, j) \in A$ is an *arc*, directed from i to j. Suppose each vertex $i \in V$ has a nonnegative cost, c_i. The *dominating set problem* is to find a minimum cost (or minimum cardinality if each vertex has unit cost) subset of vertices, $S \subseteq V$, such that for all $j \in V$, either j belongs to S or there exists an arc $(i, j) \in A$ such that i belongs to S.

For a vertex $i \in V$, let $N^-(i) \subset V$ denote the set of in-neighbors of vertex i. (We say that j is an *in-neighbor* of i if the arc (j, i) belongs to A.) In other words, $N^-(i)$ is the *in-neighborhood* of vertex i. Let $N^-[i]$ denote the *closed in-neighborhood* of vertex i, i.e. $N^-[i] = N^-(i) \cup \{i\}$. The *out-neighborhood* and *closed out-neighborhood* can be defined analogously and are denoted by $N^+(i)$ and $N^+[i]$, respectively. We can write the following linear program for the dominating set problem.

$$\begin{aligned} & \min \sum_{i \in V} c_i x_i & & \\ \text{subject to:} \quad & \sum_{j \in N^-[i]} x_j \geq 1, & & \text{for all } i \in V, \\ & x_i \geq 0, & & \text{for all } i \in V. \end{aligned} \qquad (P_{dom\text{-}set})$$

The dominating set problem can be viewed as a set cover problem, because by adding an element i to the solution set S, we are covering all elements belonging to the set $N^+[i]$. (Thus, our goal is to find a minimum cost subset $S \subseteq V$ such that $N^+[S] = V$.) Therefore, we can apply the same approach used for the set cover problem to obtain a $\log n$-approximation algorithm for the dominating set problem.

```
Randomized-Rounding-Dominating-Set

Find an optimal solution $\mathbf{x}^*$ for $(P_{dom\text{-}set})$.
For $j = 1$ to $n$:
    Repeat $2 \log n$ times: Include each vertex $i$ in the solution set $\mathcal{S}$ with probability $x_i^*$.
```

2.1 Distance-2-Domination

For a digraph $G = (V, A)$, a subset $S \subseteq V$ is a *distance-2-dominating set* if for every vertex $v \in V$, v is either dominated by some vertex in S, or there exists some arc (u, v) in A such that vertex u is dominated by some vertex in S. In other words, each vertex in V is distance at most two from some vertex in S. Let $N^{++}[i]$ denote the subset of vertices that are at a distance at most two from vertex i. In other words, a vertex j belongs to $N^{++}[i]$ if $j \in N^{+}[i]$ or if there exists an arc $(h, j) \in A$ such that $h \in N^{+}[i]$.

Recall that a tournament on n vertices, $T = (V, A)$, is a directed graph in which for every pair of vertices, i and j, either arc (i, j) belongs to A or arc (j, i) belongs to A. (Note that there is exactly one arc between i and j.) The following theorem can be proven via the greedy algorithm.

Theorem 4. *A tournament contains a dominating set of cardinality at most* $\log n$ *vertices.*

If $N^{++}[v] = V$, then vertex v is called a "king". The following theorem is due to Landau.

Theorem 5. [Lan53] *Every vertex of maximum outdegree in a tournament is a king.*

Proof. Suppose to the contrary that u is a vertex with maximum out-degree in a tournament T and u is not a king. Then there exists another vertex v in T such that is not reachable from u within two steps. But this means that u and all out-neighbours of u are reachable from v in one step and so $N^{+}(v) > N^{+}(u)$, a contradiction. □

For general graphs, we can use the linear program ($P_{dom\text{-}set}$) to find a distance-2-dominating set of cardinality at most $\gamma(G) \cdot \log \gamma(G)$, where $\gamma(G)$ denotes the optimal value of ($P_{dom\text{-}set}$) on G. Since a minimum dominating set can be larger than a distance-2-dominating set, this algorithm is not a $\log OPT$-approximation. However, if $\gamma(G)$ is smaller than n, then the cardinality of this solution is smaller than the cardinality of a dominating set returned by Randomized-Rounding-Set-Cover 2.

Theorem 6. *A directed graph $G = (V, A)$ has a distance-2-dominating set of cardinality at most $\gamma(G) \cdot (\log \gamma(G) + 1)$, where $\gamma(G)$ is the optimal solution to the linear program ($P_{dom\text{-}set}$) on G.*

Before we prove Theorem 6, we prove the following lemma.

Lemma 7. *Let $\gamma(G)$ denote the optimal value of ($P_{dom\text{-}set}$) on G. Then for any weight function $w : V \rightarrow \mathbb{R}^{+}$, there is a subset of vertices $S \subseteq V$ such that:*

1. $|S| \leq \gamma(G) \cdot (\log \gamma(G) + 1)$, *and*
2. $w(N^{+}[S]) > \left(1 - \frac{1}{\gamma(G)}\right) w(V)$.

Proof. Let $\mathbf{x}^*$ be a solution for ($P_{dom\text{-}set}$) with value $\gamma(G)$. Include each vertex $i \in V$ with probability x_i^*. Consider vertex $i \in V$, which can be covered by any element in $N^{-}[i]$, each of which is selected by the algorithm with probability $p_1, p_2, \ldots, p_k$. By the constraints from ($P_{dom\text{-}set}$), we have:

$$\sum_{j=1}^{k} p_j \quad \geq \quad 1.$$

Next, we have:

$$\begin{aligned} \Pr[i \text{ is not covered}] \quad &= \quad \prod_{j=1}^{k} (1 - p_j) \\ &< \quad \prod_{j=1}^{k} e^{-p_j} \quad = \quad e^{-\sum p_j} \quad \leq \quad \frac{1}{e}. \end{aligned}$$

So, $\Pr[i \text{ is covered}] > 1 - \frac{1}{e}$. Now suppose we repeat this procedure $\lceil \log \gamma(G) \rceil$ times. Then,

$$\begin{aligned} \Pr[i \text{ is not covered}] &< \left(\frac{1}{e}\right)^{\lceil \log \gamma(G) \rceil} && (1) \\ &\le \frac{1}{\gamma(G)}. && (2) \end{aligned}$$

So, $\Pr[i \text{ is covered}] > 1 - \frac{1}{\gamma(G)}$. Thus, there exists a solution S of size at most $\gamma(G) \cdot \lceil \log \gamma(G) \rceil \le \gamma(G) \cdot (\log \gamma(G) + 1)$ for which expected value of $w(N^+[S]) > (1 - \frac{1}{\gamma(G)}) \cdot w(V)$. □

Now we can prove Theorem 6.

Proof of Theorem 6. Let $\mathbf{x}^*$ be a solution for $(P_{dom\text{-}set})$ with value $\gamma(G)$. We define the weight function to be $w_i := x_i^*$ for each $i \in V$. Note that this implies $w(V) = \gamma(G)$. Then we can apply Lemma 7 and obtain a set $S \subset V$ such that:

1. $|S| \le \gamma(G) \cdot (\log \gamma(G) + 1)$, and
2. $w(N^+[S]) > \gamma(G) - 1$.

Now, we claim that $N^{++}[S] = V$. If not, then there exist a vertex $u \in V$ such that $u \notin N^{++}[S]$. From the constraints of $(P_{dom\text{-}set})$, we have $\sum_{i \in N^-[u]} x_i^* \ge 1$. However,

$$x^*(N^+[S]) = w(N^+[S]) > \gamma(G) - 1.$$

Thus,

$$N^+[S] \cap N^-[u] \neq \emptyset.$$

In other words, strictly less than one unit of fractional value (from the total value $x^*(V)$) is not contained in the set $N^+[S]$. So for each vertex, at least one vertex in its closed in-neighborhood belongs to $N^+[S]$. □

3 Stable Sets and Coloring

Let $G = (V, E)$ be an undirected graph on n vertices. A *stable set* of G is a subset of vertices, $S \subseteq V$, that contains no edges. (A stable set is also called an *independent set*.) The size of the maximum stable set of G is denoted by $\alpha(G)$. The *chromatic number* of G is the minimum number of stable sets that cover all the vertices (i.e., each vertex is included in one of the stable sets). The chromatic number of G is denoted by $\chi(G)$.

Finding the minimum number of stable sets that cover the vertices is equivalent to the graph coloring problem (i.e., coloring the vertices with the minimum number of colors) since each color is a stable set. Note that the graph coloring problem is a special case of set cover, where the universe is the vertices of the graph and the set system is the family of all stable sets. Let $\mathcal{S}$ denote the family of all stable sets in G. Let x_S denote the indicator variable for whether or not a stable set S is chosen to be in the solution set. The following integer program models the graph coloring problem.

$$\min \sum_{S \in \mathcal{S}} x_S$$
$$\text{subject to: } \sum_{S : v \in S} x_S \ge 1, \quad \text{for all } v \in V,$$
$$x_S \in \{0, 1\}.$$

Its linear programming relaxation models the *fractional coloring* problem, which parallels the relaxation $(P_{set\text{-}cover})$.

$$\begin{aligned} \min \sum_{S \in \mathcal{S}} & x_S \\ \text{subject to: } \sum_{S : v \in S} & x_S \geq 1, \quad \text{for all } v \in V, \\ & x_S \geq 0. \end{aligned} \qquad (P_{frac\text{-}color})$$

In an optimal extreme point solution for $(P_{frac\text{-}color})$, the number of nonzero variables is at most the number of constraints (not including the nonnegativity constraints). Therefore, the number of stable sets $S \in \mathcal{S}$ for which $x_S > 0$ is at most n. Suppose that we could actually efficiently find an optimal extreme point solution for $(P_{frac\text{-}color})$. Then since graph coloring is a set cover problem, we can apply the same approach used for the set cover problem to obtain a $\log n$-approximation algorithm for the graph coloring problem.

```
RANDOMIZED-ROUNDING-COLORING
Find an optimal extreme point solution $\mathbf{x}^*$ for $(P_{frac\text{-}color})$.
For all $S \in \mathcal{S}$ such that $x_S > 0$:
    Repeat $2 \log n$ times: Include each stable set $S$ in the solution set with probability $x^*_S$.
```

However, since it is known that $\chi(G)$ is hard to approximate to within better than $n^{1-\epsilon}$ for any constant $\epsilon > 0$, this shows that in general we cannot efficiently find an optimal extreme point solution for $(P_{frac\text{-}color})$. Indeed, $(P_{frac\text{-}color})$ is an exponential-sized linear program since there might be exponentially many stable sets and therefore exponentially many x_S variables. Nevertheless, the above randomized rounding algorithm implies that the linear programming relaxation $(P_{frac\text{-}color})$ has integrality gap $O(\log n)$.

References

[Arc01] Aaron Archer. Two $O(\log^* k)$-approximation algorithms for the asymmetric k-center problem. *In Proceedings of Integer Programming and Combinatorial Optimization (IPCO)*, pages 1–14, 2001.

[BLT17] N. Bousquet, W. Lochet, and S. Thomassé. A proof of the Erdős-Sands-Sauer-Woodrow conjecture. *arXiv:1703.08123*, 2017.

[Lan53] H. G. Landau. On dominance relations and the structure of animal societies: III The condition for a score structure. *Bulletin of Mathematical Biophysics*, 15(2):143–148, 1953.

[Vaz13] Vijay V. Vazirani. *Approximation Algorithms.* Springer, 2013.

[WS11] David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms.* Cambridge University Press, 2011.

Section 1 of these lecture notes is based on Chapter 1 of [WS11] and Chapter 14 of [Vaz13]. Theorem 6 is based on ideas found in [Arc01] and [BLT17].