Algorithms and Uncertainty, Winter 2017/18 Lecture 12 (4 pages)

Stochastic Vertex Cover and Stochastic Set Cover

Instructor: Thomas Kesselheim

In the analysis of online algorithms, we assumed that we have to make commitments right away. In practice often restrictions are not as strict. Just suppose you have to fly to New York City two months from now. You could either buy the ticket now for a cheap price or later on. Now the ticket is cheap but there is a chance that you actually cannot go on the trip. So, it might also make sense to wait and buy the ticket for a higher price when it is certain that you have to go.

This is a typical example of a multi-stage optimization problem. These are problems in which the optimization instance gets more and more concrete over time and decisions can be made on the way. There are both models with stochastic as well as adversarial inputs. Today, we will consider simple examples of such stochastic problems.

1 Stochastic Vertex Cover

Recall the standard offline weighted Vertex Cover problem: We are given a graph $G = (V, E)$ and vertex weights $(w_v)_{v \in V}$. We have to choose a subset $F \subseteq V$ of the vertices such that for each edge at least one endpoint is contained in F. That is, for all $\{u, v\} \in E$, we have $u \in F$ or $v \in F$. The objective is to minimize the sum of weights of selected vertices $\sum_{v \in F} w_v$.

In the stochastic version, the edge set E is uncertain. It is drawn from a known probability distribution. The probability that the edge set is E is given as p_E. Our algorithm knows the entire vector $(p_E)_E$ from the start. We assume that $p_E = 0$ for all except polynomially many sets E.

We can pick vertices at two points in time: Before the edge set E is revealed and afterwards. In the first stage, vertices are cheaper: For vertex v, we have to pay w_v^{I}. In the second stage, for vertex v, we have to pay $w_v^{\mathrm{II}} \geq w_v^{\mathrm{I}}$.

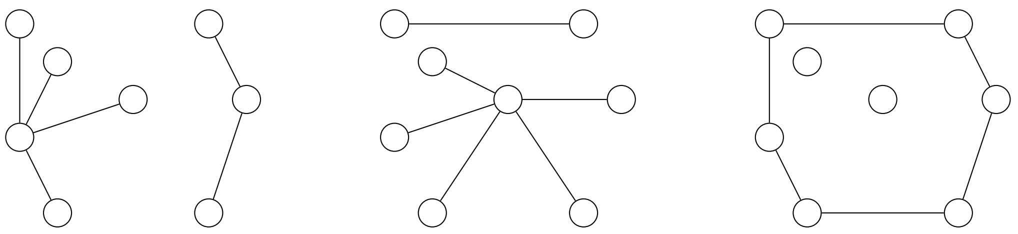
Important special cases are as follows. We might have $w_v^{\mathrm{I}} = w_v^{\mathrm{II}}$ for all v. In this case, choosing sets in the first stage does not make any sense and we might as well wait until the second stage. If $w_v^{\mathrm{II}} = \infty$, then we want to cover all edges that can possibly show up already in the first stage.

We know the distribution $(p_E)_E$ and well as both cost vectors $(w_v^{\mathrm{I}})_{v \in V}$ and $(w_v^{\mathrm{II}})_{v \in V}$ in advance. The goal is to minimize the expected cost

$$\sum_{v \text{ selected in first stage}} w_v^{\mathrm{I}} + \mathbf{E}\left[\sum_{v \text{ selected in second stage}} w_v^{\mathrm{II}}\right] .$$

Observe that this problem can be modeled as a Markov decision process with time horizon $T = 2$. So, we could in principle use the algorithm based on dynamic programming to compute an optimal policy. However, the number of states will be huge. Computing it is at least as hard as solving the Vertex Cover problem optimally because one special case is that $p_E = 1$ for one set E. Therefore, we will be interested in approximating the optimal policy in polynomial time.

Example 12.1. *An example instance could look as follows. There is a fixed set of vertices, there are three scenarios, corresponding to different edges. The problem is already interesting if in the first stage every vertex costs $w_v^I = 1$ and in the second stage every vertex costs $w_v^{II} = \lambda$.*



$p_E = 0.2$ $p_E = 0.5$ $p_E = 0.3$

2 LP-Based Algorithm

Our approach to approximating the optimal policy will be to first formulate a linear program that any policy has to fulfill but not every solution corresponds to a feasible policy. For the stochastic vertex-cover problem, we can write the following LP.

$$\begin{aligned} \min \sum_{v \in V} w_v^{\mathrm{I}} x_v + \sum_{E} p_E \sum_{v \in V} w_v^{\mathrm{II}} y_{E,v} & \\ \text{subject to } x_u + y_{E,u} + x_v + y_{E,v} \geq 1 & \qquad \text{for all } E,\ \{u,v\} \in E \\ x_u, y_{E,u} \geq 0 & \qquad \text{for all } E,\ u \in V \end{aligned}$$

Observe that we get a feasible solution by we setting $x_v = 1$ if the optimal policy chooses vertex v in the first stage and $y_{E,v} = 1$ if the optimal policy chooses vertex v in the second stage when the edge set is E. The objective function value is exactly the expected cost of the optimal policy.

Our approximation algorithm computes an optimal solution (x^*, y^*) to this LP. This can be done in polynomial time if $p_E > 0$ for only polynomially many sets E. This solution does not necessarily correspond to a feasible policy because values can be fractional. We derive a feasible policy as follows.

- In the first stage, pick all vertices for which $x_v^* \geq \frac{1}{4}$.
- In the second stage, when knowing E, pick all vertices for which $y_{E,v}^* \geq \frac{1}{4}$.

Theorem 12.2. *The algorithm computes a feasible policy whose expected cost is at most 4-times the cost of the optimal policy.*

Proposition 12.3. *The algorithm always computes a feasible policy.*

Proof. Consider any scenario E and $e = \{u,v\} \in E$. As (x^*, y^*) is a feasible LP solution, we have

$$x_u^* + y_{E,u}^* + x_v^* + y_{E,v}^* \geq 1 \ .$$

This means that one of x_u^*, $y_{E,u}^*$, x_v^*, and $y_{E,v}^*$ is at least $\frac{1}{4}$. This means that edge e is covered in scenario E. □

Proposition 12.4. *The expected cost of the computed policy is at most 4-times the expected cost of the optimal policy.*

Proof. Let F_0 be the set of vertices picked by the computed policy in the first stage, F_E be the set of vertices picked in the second stage if the edge set if E.

We now have

$$\sum_{v \in F_0} w_v^{\mathrm{I}} \leq 4 \sum_{v \in V} w_v^{\mathrm{I}} x_v^* \qquad \text{and} \qquad \sum_{v \in F_E} w_v^{\mathrm{II}} \leq 4 \sum_{v \in V} w_v^{\mathrm{II}} x_{E,v}^* \enspace .$$

Therefore

$$\sum_{v \in F_0} w_v^{\mathrm{I}} + \mathbf{E}\left[\sum_{v \in F_E} w_v^{\mathrm{II}}\right] = \sum_{v \in F_0} w_v^{\mathrm{I}} + \sum_E p_E \sum_{v \in F_E} w_v^{\mathrm{II}} \leq 4 \left(\sum_{v \in V} w_v^{\mathrm{I}} x_v^* + \sum_E p_E \sum_{v \in V} w_v^{\mathrm{II}} x_{E,v}^*\right) \enspace .$$

As observed above, the cost of the optimal LP-solution is upper bounded by the expected cost of the optimal policy. □

3 Stochastic Set Cover

The technique that we used for stochastic Vertex Cover generalizes to a stochastic version of Set Cover.

Recall that in the offline Set Cover problem, there is a universe of m elements U and a family of subsets $\mathcal{S} \subseteq 2^U$. Each set $S \in \mathcal{S}$ has a weight w_S. We have to select a cover $\mathcal{C} \subseteq \mathcal{S}$ such that for all $j \in U$ there is some $S \in \mathcal{C}$ with $j \in \mathcal{S}$. We want to minimize the cost $\sum_{S \in \mathcal{C}} w_S$.

In the stochastic version, it is uncertain only a subset $A \subseteq U$ has to be covered. That is only $j \in A$ we there has to be $S \in \mathcal{C}$ with $j \in \mathcal{S}$. It is uncertain which set it is. The set A is drawn from a known probability distribution. We denote by p_A, $A \subseteq U$, the probability that A has to be covered.

Eventually, we will have to cover all of A. We have two opportunities to select sets: Before A is revealed and afterwards. Before A is revealed (stage I), adding $S \in \mathcal{S}$ costs w_S^{I}; after A is revealed (stage II), it costs $w_S^{\mathrm{II}} \geq w_S^{\mathrm{I}}$.

To formulate an LP, given an arbitrary policy, let $x_S = 1$ if set S is selected in the first stage, 0 otherwise. Let $y_{A,S} = 1$ if set S is selected in the second stage if set A has to be covered, 0 otherwise.

$$\begin{aligned}
\text{minimize} & \sum_{S \in \mathcal{S}} w_S^{\mathrm{I}} x_S + \sum_{A \subseteq U} p_A \sum_{S \in \mathcal{S}} w_S^{\mathrm{II}} y_{A,S} & \\
\text{subject to} & \sum_{S\colon e \in S} x_S + \sum_{S\colon e \in S} y_{A,S} \geq 1 & \text{for all } A \subseteq U,\ e \in A \\
& x_S, y_{A,S} \geq 0 & \text{for all } S \in \mathcal{S},\ A \subseteq U
\end{aligned}$$

Again, if $p_A > 0$ only for a small number of sets, we can solve this linear program in polynomial time.

Theorem 12.5. *Any fractional solution to the LP can be turned into a feasible policy of at most $O(\log m)$-times the cost in polynomial time.*

Proof. Given an optimal solution (x^*, y^*), we proceed as follows. If for an element $e \in U$ we have $\sum_{S\colon e \in S} x_S^* \geq \frac{1}{2}$, then we make sure to cover it in the first stage. Let U_0 be the set of such elements and consider the deterministic LP of the Set Cover instance in which there is only one stage and U_0 has to be covered:

$$\begin{aligned} &\text{minimize} \sum_{S \in \mathcal{S}} w^{\mathrm{I}}_S x_S && \\ &\text{subject to} \sum_{S \colon e \in S} x_S \geq 1 && \text{for all } e \in U_0 \\ &x_S \geq 0 && \text{for all } S \in \mathcal{S} \end{aligned}$$

Note that $2x^*$ is a feasible solution, so the optimal value is bounded by $2 \sum_{S \in \mathcal{S}} w^{\mathrm{I}}_S x^*_S$.

To choose which sets S to pick in the first stage, we find an integral Set Cover solution to this LP. For example, we have seen much earlier in this course how to round fractional solutions in a randomized way. But in this case, one could also apply a simple Greedy algorithm. As a result, we get a first-stage solution that covers all of U_0 and has cost at most

$$O(\log m) \cdot 2 \sum_{S \in \mathcal{S}} w^{\mathrm{I}}_S x^*_S \enspace .$$

In the second stage, we only have to cover $A \setminus U_0$. Observe that for all $e \in A \setminus U_0$

$$\sum_{S \colon e \in S} y^*_{A,S} \geq \frac{1}{2} \enspace .$$

So, we can follow just the same idea as above and get a cover of cost at most

$$O(\log m) \cdot 2 \sum_{S \in \mathcal{S}} w^{\mathrm{II}}_S y^*_{A,S} \enspace .$$

In combination, our cover will cost in expectation

$$O(\log m) \cdot \left(\sum_{S \in \mathcal{S}} w^{\mathrm{I}}_S x^*_S + \sum_A p_A \sum_{S \in \mathcal{S}} w^{\mathrm{II}}_S y^*_{A,S} \right) \enspace .$$

□

4 Extensions and Limitations

All that we have done today also works if the second-stage costs depend on the scenario. You could even model that sometimes vertices or sets get cheaper on the second stage.

One major difficulty is that the LP enumerates all scenarios explicitly. In particular, if each edge is present with probability $\frac{1}{2}$ independently, we would have $2^{\frac{n(n-1)}{2}}$ different scenarios and the LP gets huge. There are still approaches to solve it but this would exceed the scope of this course. Next time, we will consider a different algorithmic approach that is also able to deal with such a huge number of scenarios.

References

- On the costs and benefits of procrastination: Approximation algorithms for stochastic combinatorial optimization problems, N. Immorlica, D. Karger, M. Minkoff. V. Mirrokni, SODA 2004 (Vertex Cover)
- Stochastic optimization is (almost) as easy as deterministic optimization, D. Shmoys, C. Swamy, FOCS 2004 (Set Cover and generalizations)