601.435/635 Approximation Algorithms **Lecturer:** Michael Dinitz
Topic: Strengthening Relaxations: Knapsack-Cover Inequalities **Date:** 4/25/19
Scribe: Michael Dinitz

24.1 Introduction

Today we're going to talk about a topic that we probably should have done earlier in the semester when we were talking about LPs. My apologies for the somewhat poor planning. But it's also motivated by an observation that we made last class about the SDP relaxation for Correlation Clustering where we had to add $v_i \cdot v_j \geq 0$ constraints in order to make the algorithm and analysis work. Another interpretation was that if we started without those constraints, the SDP was actually a poor (though valid) relaxation for the vector program that we actually wanted to solve. But we could fix this by adding more valid constraints: constraints which are true for any feasible solution of the original problem, but which might not appear in the "obvious" relaxation. It turns out that this is generally a useful thing to think about: if we're given an LP or SDP which is a poor relaxation (it has large integrality gap or we don't know how to round it), we can try to add valid inequalities to make it a better or easier to handle relaxation.

24.2 Min-Knapsack and the Knapsack Cover Inequalities

We'll begin, and probably spend most of our time, on a variation of a problem that we all know and love: Knapsack. However, we're going to consider a minimization version of Knapsack, rather than the slightly more well-known maximization version.

24.2.1 Definition and Basic Relaxation

The Min-Knapsack problem is defined as follows.

- **Input**: A set of $I = [n]$ of items, values $v : I \to \mathbb{R}_{\geq 0}$, sizes $s : I \to \mathbb{R}_{\geq 0}$, demand $D \in \mathbb{R}_{\geq 0}$ (note that we're not given a knapsack capacity)
- **Feasible Solution**: $X \subseteq I$ such that $v(X) = \sum_{i \in X} v(i) \geq D$
- **Objective**: minimize $s(X) = \sum_{i \in X} s(i)$

In other words, we want to find the minimum size knapsack necessary to carry items of total value at least D. Note that without loss of generality, we may assume that $v(i) \leq D$ for all $i \in I$, since if $v(i) > D$ we could just set $v(i) = D$ and the set of feasible solutions and their costs would be unchanged. This will be important later.

It's not hard to see that the PTAS we designed for Knapsack can be modified to work for Min-Knapsack. But, for a variety of reasons that we won't really go into here, it's also interesting to study LP relaxations for these problems. If we start by writing an ILP (as usual), we would probably write the obvious thing:

$$\begin{aligned}
\min \quad & \sum_{i \in I} s(i) x_i \\
\text{s.t.} \quad & \sum_{i \in I} v(i) x_i \geq D \\
& x_i \in \{0,1\} \qquad \forall i \in I
\end{aligned}$$

If we relax this to an LP, we would get

$$\begin{aligned}
\min \quad & \sum_{i \in I} s(i) x_i \\
\text{s.t.} \quad & \sum_{i \in I} v(i) x_i \geq D \\
& 0 \leq x_i \leq 1 \qquad \forall i \in I
\end{aligned}$$

However, this is an extremely weak relaxation. To see this, consider the following case. $I = \{1, 2\}$, with $v(1) = D - 1$ and $s(1) = 1$, and with $v(2) = D$ and $s(2) = 1$. Clearly in this instance $OPT = 1$, since we are forced to take the second item in order to get value D in total. On the other hand, we can set $x_1 = 1$ and $x_2 = 1/D$ to get a feasible LP solution of cost $1/D$. Thus the integrality gap is at least D, which can be extremely large (exponential in the size of the input). And this is despite the fact that Min-Knapsack is an "easy" problem, and the fact that the ILP we wrote was an exact formulation.

So we cannot use this LP to get a reasonable LP rounding or primal-dual algorithm. Can we "add-on" extra constraints to this LP to get a stronger LP? Note that this question is similar to but different from what we did earlier for the Multiway Cut problem. For Multiway Cut, we wrote one LP relaxation and got a 2-approximation, and then wrote an entirely different LP relaxation to get a 3/2-approximation. Our goal here isn't to write something that's completely different, but rather just to add extra constraints to make this a better LP.

24.2.2 Knapsack-Cover Inequalities

To get some intuition, let's think about how our first LP was able to "cheat". At a very high level, in our bad example, the LP was able to "buy" a partial solution, and then because of this the remaining demand was so low that it could be satisfied at an extremely low cost. To cover the remaining demand of 1 after item 1 was bought, the LP only had to buy the second item to a fraction of $1/D$. This is how it "cheated".

How can we add constraints that forbid this? Again, let's think intuitively. After the LP has decided to buy item 1 to a value of 1, what's left is actually another Min-Knapsack problem, with remaining demand $D - (D - 1) = 1$ and only one item (the second of the original items). This item has value of D, but does that really make sense? In general, if we're given an instance of Min-Knapsack, then if any item has value larger than D then we can just reduce its value to D, and nothing changes about the optimal solution. So we want to somehow add constraints which say something like "if after buying some items you're left with an instance with remaining demand D', then pretend like every remaining item has value at most D'"

Now let's do this formally. Let $A \subseteq I$. If we think of buying A, then the remaining demand would be $D - v(A)$, and for every $i \notin A$ we would set its "new" value to be $\min(v(i), D - v(A))$. Let's just write all of those constraints!

$$\begin{aligned}
\min \quad & \sum_{i \in I} s(i) x_i \\
\text{s.t.} \quad & \sum_{i \in I \setminus A} \min(v(i), D - v(A)) x_i \geq D - v(A) \qquad \forall A \subseteq I \\
& 0 \leq x_i \leq 1 \qquad \forall i \in I
\end{aligned}$$

Note that we have the constraint from the original LP explicitly (since when $A = \emptyset$ we get back exactly the constraint from the original LP), but we also have a bunch of new constraints. Since this isn't as obvious as it usually is, let's prove that this is a valid relaxation.

Lemma 24.2.1 *Any integral solution to this LP gives a feasible solution to Min-Knapsack with the same cost.*

Proof: This was true of the original LP relaxation, and any integral solution to our new LP is also an integral solution to our old LP, so it is also true of this LP. ■

Lemma 24.2.2 *Let X be a feasible solution to Min-Knapsack. Let $x_i = 1$ if $i \in X$ and $x_i = 0$ otherwise. Then x is a feasible solution to the new LP with cost $s(X)$.*

Proof: The cost part is easy:

$$\sum_{i \in I} s(i) x_i = \sum_{i \in X} s(i) = s(X)$$

as claimed. Now we need to prove that x is feasible. Consider the constraint for some $A \subseteq I$. Since X is feasible, we know that $\sum_{i \in X} v(i) \geq D$, and thus $\sum_{i \in X \setminus A} v(i) + \sum_{i \in X \cap A} v(i) \geq D$. Rewriting this slightly, we get that $\sum_{i \in X \setminus A} v(i) \geq D - v(A \cap X)$, and thus

$$\sum_{i \in X \setminus A} v(i) \geq D - v(A).$$

But now, as discussed earlier, this means that

$$\sum_{i \in X \setminus A} \min(v(i), D - v(A)) \geq D - v(A).$$

Now by the definition of x, this implies that

$$\sum_{i \in I \setminus A} \min(v(i), D - v(A)) x_i = \sum_{i \in X \setminus A} \min(v(i), D - v(A)) \geq D - v(A).$$

Thus the constraint for A is satisfied. Since we did this for an arbitrary A, this implies that all of the constraints of the LP are satisfied, and thus x is feasible. ■

Before we show how to use this new LP in an algorithm, let's do a sanity check and show that this at least solves the integrality gap instance we developed for the first LP. Consider the set $A = \{1\}$. Then the constraint for this set is just $x_2 \geq 1$, since $I \setminus A = \{2\}$ and $D - v(A) = 1$. So in this new LP, our original motivating instance is solved exactly: the LP is forced to buy the second item integrally, just like in an integral solution!

These new inequalities are known as the *Knapsack-Cover inequalities*, and were introduced by Carr, Fleischer, Leung, and Phillips [CFLP00]. The basic idea, of thinking about what happens if we have a partial solution and reducing values/sizes/demands etc. accordingly, has now been used in a bunch of more complicated problems, where the associated inequalities are still also called knapsack-cover inequalities. For example, a few years ago I used them in an approximation algorithm for a network design problem known as the f-Fault Tolerant 2-Spanner problem [DK11].

24.2.3 Primal-Dual Algorithm

Originally, this new LP was used to give a 2-approximation via LP rounding [CFLP00]. In order to stay consistent with the book, and since it gives a better running time anyway, we're going to use the new LP to give a 2-approximation via a primal-dual algorithm due to Carnes and Shmoys [CS15].

The dual of the LP has a variable for each $A \subseteq I$ and a constraint for each $i \in I$:

$$\begin{aligned}
\max \quad & \sum_{A \subseteq I} (D - v(A)) y_A & \\
\text{s.t.} \quad & \sum_{A \subseteq I : i \notin A} \min(v(i), D - v(A)) y_A \leq s(i) & \forall i \in I \\
& 0 \leq y_A \leq 1 & \forall A \subseteq I
\end{aligned}$$

Now consider the following primal-dual algorithm. Intuitively, we're always going to increase the dual variable corresponding to the set that we've bought so far.

```
• Initialize $y = \vec{0}$ and $X = \emptyset$
• While $v(X) < D$:
    – increase $y_X$ until the dual constraint for some $i \in I \setminus X$ becomes tight; i.e., until there
      is some $i \in I \setminus X$ such that $\sum_{A \subseteq I : i \notin A} \min(v(i), D - v(A)) y_A = s(i)$
    – Set $X \leftarrow X \cup \{i\}$
• Return $X$
```

Lemma 24.2.3 *$\vec{y}$ is a feasible dual solution throughout the algorithm.*

Proof: Induction on the iterations of the algorithm. Clearly it is true initially when $y = \vec{0}$. Now consider some iteration of the algorithm, and let X be the set at the beginning of the iteration. Since all tight dual constraints correspond to items that are in X, when we increase y_X we do not violate any dual constraints. So at the end of the iteration it is still true that $\vec{y}$ is a feasible dual solution. ■

Theorem 24.2.4 *This primal-dual algorithm is a 2-approximation.*

Proof: We start as with a standard primal-dual analysis, where we use the fact that the items we buy correspond to tight dual constraints. Let X be the set returned by the algorithm.

$$\begin{aligned} s(X) &= \sum_{i \in X} s(i) = \sum_{i \in X} \sum_{A \subseteq I : i \notin A} \min(v(i), D - v(A)) y_A = \sum_{A \subseteq I} \sum_{i \in X \setminus A} \min(v(i), D - v(A)) y_A \\ &= \sum_{A \subseteq I} y_A \sum_{i \in X \setminus A} \min(v(i), D - v(A)). \end{aligned}$$

Let $\ell \in X$ be the last item added by the algorithm. Then clearly $v(X \setminus \{\ell\}) < D$ but $v(X) \geq D$.

Now consider some $A \subseteq I$ such that $y_A \neq 0$, i.e., some $A \subseteq X$ which at one point in the algorithm was X. We want to bound $\sum_{i \in X \setminus A} \min(v(i), D - v(A))$ (we don't care about this value for other sets A since if $y_A = 0$ it won't contribute to the overall cost anyway). For any $i \in X \setminus A$ other than ℓ, if we added i to A the total value would still be less than D, or else the algorithm would have stopped after it (eventually) added i, so wouldn't get around to adding ℓ. Thus for all $i \in X \setminus A$ with $i \neq \ell$, we know that $\min(v(i), D - v(A)) = v(i)$. So we can continue the above analysis with this fact to get

$$\begin{aligned} s(X) &= \sum_{A \subseteq I} y_A \sum_{i \in X \setminus A} \min(v(i), D - v(A)) \\ &= \sum_{A \subseteq I} y_A \left(\min(v(\ell), D - v(A)) + \sum_{i \in X \setminus A : i \neq \ell} v(i) \right) \\ &= \sum_{A \subseteq I} y_A \left(\min(v(\ell), D - v(A)) + v(X \setminus \{\ell\}) - v(A) \right) \\ &< \sum_{A \subseteq I} y_A ((D - v(A)) + D - v(A)) \\ &= 2 \sum_{A \subseteq I} (D - v(A)) y_A. \end{aligned}$$

Thus X has cost which is at most twice the value of the feasible dual solution y, so by weak duality we know that X is a 2-approximation. ■

References

[CFLP00] Robert D. Carr, Lisa K. Fleischer, Vitus J. Leung, and Cynthia A. Phillips. Strengthening integrality gaps for capacitated network design and covering problems. In *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA

'00, pages 106–115, Philadelphia, PA, USA, 2000. Society for Industrial and Applied Mathematics.

[CS15] Tim Carnes and David B. Shmoys. Primal-dual schema for capacitated covering problems. *Math. Program.*, 153(2):289–308, 2015.

[DK11] Michael Dinitz and Robert Krauthgamer. Fault-tolerant spanners: Better and simpler. In *Proceedings of the 30th Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing*, PODC '11, pages 169–178, New York, NY, USA, 2011. ACM.