ORIE 6300 Mathematical Programming I October 23, 2014

Lecture 17

Lecturer: David P. Williamson *Scribe: Jeff Tian*

1 The Knapsack Problem

In the Knapsack Problem, we have items $1 = 1, \ldots, m$ with size s_i and value y_i, a knapsack of size W, and want to maximize the value of the goods that fit in the knapsack by taking a_i of each type of good. We can efficiently solve the knapsack problem with a dynamic programming algorithm.

We assume s_i and W are integers and each $s_i > 0$.[1] We then define $F_i(v)$ to be the optimum value of the knapsack if the knapsack size is v and we only can take items from $\{1, \ldots, i\}$. Ultimately, we want $F_m(w)$. First, we compute directly $F_1(v)$ for $v = 0, \ldots, W$ directly, as

$$F_1(v) = \begin{cases} \left\lfloor \frac{v}{s_1} \right\rfloor y_1 & y_1 > 0, \\ 0 & \text{otherwise.} \end{cases}$$

For each $i = 1, \ldots, m-1$, for each $v = 0, \ldots, W$, we can compute the remaining terms by the recurrence

$$F_{i+1}(v) = \underbrace{\max_{a_{i+1} = 0, \ldots, \left\lfloor \frac{v}{s_{i+1}} \right\rfloor}}_{\#\text{ times item i+1 used}} \left(\underbrace{y_{i+1} a_{i+1}}_{\text{value from item i+1}} + \underbrace{F_i(\overbrace{v - a_{i+1} s_{i+1}}^{\substack{\text{Space left over} \\ \text{after taking } a_{i+1} \\ \text{items of type } i+1}})}_{\substack{\text{Optimal value of items } 1, \ldots, i \\ \text{fitting in the remaining space}}} \right)$$

As we have mW entries and we must maximize over at most w terms in computing each entry (if for example, every $s_i = 1$), the algorithm will run in $\mathcal{O}(mW^2)$. By modifying the algorithm to store items that were selected, we can obtain the optimal a as well, or alternatively, we can reconstruct a directly from the dynamic programming table F.

[1] More generally, we need only assume s_i and W are rationals where $s_i > 0$, as we can multiply our constraints by a common denominator to make everything integer valued. We cannot in general extend this algorithm for irrational s_i or w.

2 Dantzig-Wolfe Decompositions

Consider a linear program of the form:

$$\begin{array}{lccccccccl}
\min & c_1^T x_1 & + & c_2^T x_2 & + & \ldots & + & c_m^T x_m & & \\
\text{s.t.} & A_{01}x_1 & + & A_{02}x_2 & + & \ldots & + & A_{2m}x_m & = b_0 & \\
 & A_{11}x_1 & & & & & & & = b_1 & \\
 & & & A_{22}x_2 & & & & & = b_2 & \\
 & & & & & \ldots & & & & \\
 & & & & & & & A_{mm}x_m & = b_m & \\
 & & & & & & & x_j & \geq 0 & \forall j = 1, 2, ..., m,
\end{array}$$

where $A_{ij} \in \mathbb{R}^{m_i \times n_j}$, $c_j, x_j \in \mathbb{R}^{n_j}$, and $b_i \in \mathbb{R}^{m_i}$. This linear program can be equivalently written more consisely as:

$$\begin{array}{lll}
\min & \displaystyle\sum_{i=1}^{m} c_i^T x_i & \\
\text{subject to} & \displaystyle\sum_{i=1}^{m} A_{0i}x_i = b_0 & \\
 & A_{ii}x_i = b_i & \forall i = 1, \ldots, m \\
 & x_i \geq 0 & \forall i = 1, \ldots, m
\end{array}$$

In terms of motivation, such an LP is quite reasonable. For example, consider a bakery chain minimizing its costs. Each bakery in the chain has its own requirements for production, and there are global constraints on shared resources, perhaps flour. We want to minimize the costs of the entire chain.

The first type of constraint $\sum_{i=1}^{m} A_{0i}x_i = b_0$ is called a linking constraint. So we have a set of linking constraints (row 1) followed by m smaller systems. Each of the A_{ij} are matrices of size $m_i \times n_j$; these blocks are not necessarily square. Similarly, the $c_1, ..., c_m$, $x_1, ..., x_m$, and $b_1, ..., b_m$ are vectors. Our goal is to reformulate this problem so that we can take advantage of its special structure to solve it more easily.

In such linear programs, our main assumption is that the following local optimization subproblem is relatively easy to solve for any cost vector $\hat{c}$:

$$\begin{array}{lll}
\min & \hat{c}^T x_i & \\
\text{s.t.} & A_{ii}x_i & = b_i \\
 & x_i & \geq 0
\end{array}$$

We will take advantage of this to help us solve the global optimization problem.

The feasible region of this subproblem is $Q_i = \{x_i \in \Re^{n_i} : A_{ii}x_i = b_i, x_i \geq 0\}$. We will assume for simplicity that Q_i is bounded. Then if we enumerate its vertices $\{v_{i1}, v_{i2}, ..., v_{iN_i}\}$, these v_{ij}'s completely define the feasible region. That is, for any $x_i \in Q_i$ we can write x_i as a convex combination of the v_{ij}. That is, $x_i = \sum_{j=1}^{N_i} \lambda_{ij} v_{ij}$ for some λ_{ij} such that $\sum_{j=1}^{N_i} \lambda_{ij} = 1$ and $\lambda_{ij} \geq 0$. Thus we can rewrite the original LP in terms of the variables λ_{ij} as follows:

$$\begin{array}{lllll}
\min & \sum_{i=1}^{m} \sum_{j=1}^{N_i} \lambda_{ij}(c_i v_{ij}) & & \\
\text{s.t.} & \sum_{i=1}^{m} \sum_{j=1}^{N_i} \lambda_{ij}(A_{0i} v_{ij}) & = b_0 & \\
& \sum_{j=1}^{N_i} \lambda_{ij} & = 1 & \forall i = 1, 2, ..., m \\
& \lambda_{ij} & \geq 0 & \forall i, j.
\end{array}$$

This is sometimes called the *master problem.*

This new formulation reduces the number of constraints to $m_0 + m$, since there are now m_0 constraints for the linking constraints and only 1 constraint for each of the m subproblems. However, the number of variables is now huge! Therefore, to solve this problem we will want to use the revised simplex method, so that we do not need to keep track of all of these variables at once. As in the cutting stock problem, to do this, we need to be able to check whether the reduced cost corresponding to each variable is negative.

We first create the dual variables $y \in \mathbb{R}^{m_0}$ corresponding to the m_0 linking constraints, and we create $z \in \mathbb{R}^m$ corresponding to the m subproblem constraints (which set the sums of the λ_{ij} to 1). To check the reduced cost corresponding to variable λ_{ij}, we consider

$$\begin{aligned}
\bar{c}_{ij} &= c_i^T v_{ij} - \left((A_{0i} v_{ij})^T y + e_i^T z\right) \\
&= c_i^T v_{ij} - y^T (A_{0i} v_{ij}) - z_i \\
&= (c_i - A_{0i}^T y)^T v_{ij} - z_i.
\end{aligned}$$

which we get from taking the inner product of the dual variables with column for λ_{ij}:

$$[y_0 \quad z] \left[\begin{array}{c} A_{0i} v_{ij} \\ \hline e_i \end{array}\right]$$

Does there exist v_{ij} such that $\bar{c}_{ij} < 0$ or equivalently such that $(c_i - y_0 A_{0i})^T v_{ij} < z_i$? How can we answer this? Consider $\bar{c}_i = c_i - A_{0i}^T y$ in the following subproblem:

$$\begin{array}{lll}
\min & \bar{c}_i^T x_i & \\
\text{s.t.} & A_{ii} x_i & = b_i \\
& x_i & \geq 0.
\end{array}$$

Since we assume that Q_i is nonempty and bounded, this problem must have an optimal solution which is a vertex; call it $v = v_{ik}$ for some k. If the optimal value $\bar{c}_i v < z_i$ then $\bar{c}_{ik} < 0$, which implies that the index of variable λ_{ik} should enter the basis. If $\bar{c}_i v \geq z_i$, then this implies that $\bar{c}_{ij} \geq 0$ for all vertices v_{ij} for $j = 1, \ldots, N_i$.

Therefore, by solving the auxiliary optimization problem for each of $Q_1, ..., Q_m$ we either find a negative reduced cost column or else we prove that the current solution to the master problem is optimal.

Notice that what we have really done here is simply the revised simplex method. Using this method is good because it vastly reduced the amount of space required to solve the problem. Also, it is good computationally because the slave problems can be solved in parallel; we simply choose the column corresponding to whichever problem answers back first with negative reduced cost to enter the basis.