Optimization Methods in Finance **(EPFL, Fall 2010)**

Lecture 11: Integer programming and Branch and Bound

1.12.2010

Lecturer: Prof. Friedrich Eisenbrand *Scribe: Boris Oumow*

1 Integer Programming

Definition 1. An integer program (IP) is a problem of the form:

$$\begin{aligned} & \max c^T x \\ \text{s.t.} \quad & Ax \leq b \\ & x \in \mathbb{Z}^n \end{aligned}$$

If we drop the last constraint $x \in \mathbb{Z}^n$, the linear program obtained is called the *LP-relaxation* of (IP).

Example 2 (Combinatorial auctions)**.** A combinatorial auction is a type of smart market in which participants can place bids on combinations of discrete items, or "packages", rather than just individual items or continuous quantities. In many auctions, the value that a bidder has for a set of items may not be the sum of the values that he has for individual items. It may be more or it may be less. In other words, let $M = \{1, \dots, m\}$ be the set of items that the auctioneer has to sell. A *bid* is a pair $B_j = (S_j, p_j)$ where $S_j \subseteq M$ is a nonempty subset of the items and p_j is the price for this set. We suppose that the auctioneer has received n items B_j, $j = 1, \dots, n$.
Question: How should the auctioneer determine the winners and the loosers of bidding in order to maximize his revenue?

In lecture 10, we have seen a numerical example of this problem. Let's see now its generalization.
The (IP) for this problem is:

$$\begin{aligned} & \max \sum_{j=1}^n p_j x_j \\ \text{s.t.} \quad & Ax \leq 1 \\ & 0 \leq x \leq 1 \\ & x \in \mathbb{Z}^n \end{aligned}$$

where $A \in \{0,1\}^{m \times n}$ is the matrix defined by

$$A_{ij} = \begin{cases} 1 & \text{if } i \in S_j \\ 0 & \text{otherwise} \end{cases}$$

	L.A	Pittsburgh	Boston	Houston
West	60	120	180	180
Midwest	48	24	60	60
East	216	180	72	180
South	126	90	108	54

Table 1: Lost interest in thousand of $ per year.

and where the variables are such that $x_i \in \{0,1\}$ means whether bid i is served or not (observe the last two constraints are equivalent to $x \in \{0,1\}^n$).
A combinatorial auction problem as above is also called a *weighted set-packing problem*.

Example 3 (Lockbox problem)**.** Consider a national US firm that receives checks from all over the United States. Due to the vagaries of the U.S. Postal Service, as well as the banking system, there is a variable delay from when the check is postmarked (and hence the customer has met her obligation) and when the check clears (and when the firm can use the money). For instance, a check mailed in Pittsburgh sent to a Pittsburgh address might clear in just two days. A similar check sent to Los Angeles might take eight days to clear. It is in the firm's interest to have the check clear as quickly as possible since then the firm can use the money. In order to speed up this clearing, firms open offices (called lockboxes) in different cities to handle the checks and minimize loss of interest (obtained by considering the clearing time and the daily value from each region and the investment rate) .

For example, suppose we receive payments from four regions (West, Midwest, East, and South). We are considering a set of alternatives for opening lockboxes in L.A., Pittsburgh, Boston, and/or Houston. Operating a lockbox costs 90,000 per year. The average loss for each possible assignment are given in table 1.

Question: Where to put lock boxes and how to assign regions to lock boxes such that operational cost + loss are minimal?

Let p_{ij} be the loss if region $i \in \{1,\ldots,4\}$ sends the ckecks to the office at place $j \in \{1,\ldots,4\}$ (e.g. $p_{11} = 60, p_{12} = 120,\ldots$). The (IP) for this problem is:

$$\begin{array}{lrcll} \min\ 90\sum_{j=1}^4 y_j + \sum_{i=1}^4 \sum_{j=1}^4 p_{ij}x_{ij} & & & & \\ \text{s.t.} & \sum_{i=1}^4 x_{ij} & \leq & 4y_j & \forall j \\ & \sum_{j=1}^4 x_{ij} & = & 1 & \forall i \\ & x_{ij}, y_j & \in & \{0,1\} & \forall i,j \end{array}$$

where the variables are such that $y_j \in \{0,1\}$ means whether lock box j is open or not and $x_{ij} \in \{0,1\}$ means whether region i sends its checks to lock box j or not.

The first constraint means that a region can only be assigned to an open lockbox and the second one that each region must be assigned to one lockbox.

Example 4 (Index Fund)**.** Active portfolio management tries to achieve superior performance by using technical and fundamental analysis as well as forecasting techniques. On the other hand, passive portfolio management avoids any forecasting techniques and rather relies on diversification to achieve a desired performance. One passive management strategy is *indexing*. Let's describe this strategy. We consider a large set of stocks.

Question: How to choose a small subset of stocks that replicates as good as possible the larger portfolio? Such a portfolio is called an index fund. Given a target population of n stocks, how to select q stocks, to represent the target population as closely as possible?

We present a model that clusters the assets into groups of similar assets and selects one representative asset from each group to be included in the index fund portfolio. To do this, we consider that we have the following similarity measure p for all pairs (i, j) of stocks:

$$p_{ij} = \text{similarity between stock i and stock j}, \quad 0 \leq p_{ij} \leq 1.$$

For example, $p_{ii} = 1$ and p_{ij} is larger for more similar stocks. An example of this is the correlation between the returns of stocks i and j.

The (IP) for this problem is then:

$$\begin{array}{rrcll} \max \sum_{i,j} p_{ij}x_{ij} & & & & \\ \text{s.t.} & \sum_j y_j & = & q & \\ & \sum_j x_{ij} & = & 1 & \forall i \\ & x_{ij}, y_j & \in & \{0,1\} & \forall i, j \end{array}$$

2 Branch and Bound

We have gone through a number of examples of integer programs. A natural question is "How can we obtain solutions to these models?" There are two common approaches. Historically, the first method developed was based on cutting planes (adding constraints to force integrality). In the last twenty years or so, however, the most effective technique has been based on dividing the problem into a number of smaller problems in a method called *branch and bound*. But we can also combine the cutting plane method to perform the efficiency of the branch and bound.

Example 5. We first explain branch and bound by solving the following pure integer linear program:

$$\begin{array}{rrcl} \max x_1 + x_2 & & & \\ \text{s.t.} & x_2 - x_1 & \leq & 2 \\ & 8x_1 + 2x_2 & \leq & 19 \\ & x_1, x_2 & \geq & 0 \\ & x_1, x_2 & \in & \mathbb{Z} \end{array}$$

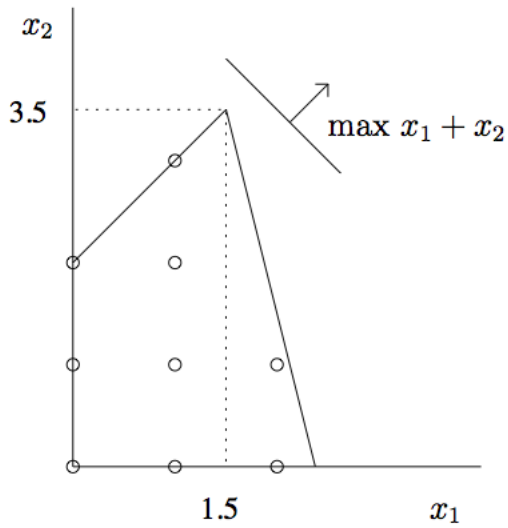

Figure 1: Graphical representation of the two-variable program of the example.

This situation is represented in the Figure 1. The first step is to solve the linear programming relaxation obtained by ignoring the last constraint. The solution is (1.5, 3.5) with objective value 5. This is not a feasible solution to the integer program since the values of the variables are fractional. How can we exclude this solution while preserving the feasible integral solutions? One way is to branch, creating two linear programs, say one with $x_1 \leq 1$ and the other with $x_1 \geq 2$. Clearly, any solution to the integer program must be feasible to one or the other of these two problems. We will solve both of these linear programs:

$$\begin{array}{rcl} \max x_1 + x_2 & & \\ \text{s.t.} \quad x_2 - x_1 & \leq & 2 \\ 8x_1 + 2x_2 & \leq & 19 \\ x_1 & \leq & 1 \\ x_1, x_2 & \geq & 0 \end{array}$$

and

$$\begin{array}{rcl} \max x_1 + x_2 & & \\ \text{s.t.} \quad x_2 - x_1 & \leq & 2 \\ 8x_1 + 2x_2 & \leq & 19 \\ x_1 & \geq & 2 \\ x_1, x_2 & \geq & 0 \end{array}$$

The solution of the first problem is $(1, 3)$ with objective value 4. This is a feasible integral solution. So we now have an upper bound of 5 as well as a lower bound of 4 on the value of an optimum solution to the integer program. The solution of the second problem is $(2, 1.5)$ with objective value 3.5. Because this value is worse than the lower bound of 4 that we already have, we do not need any further branching. We conclude that the feasible integral solution of value 4 found earlier is optimal.

Now we go to the general case. The task is to solve an integer program:

$$(I) \qquad \begin{array}{rr} & \max c^T x \\ \text{s.t.} & Ax \le b \\ & x \in \mathbb{Z}^n \end{array}$$

The main idea is that the optimal integer solution satisfies

$$(IIa) \begin{cases} Ax \le b \\ x_j \le \beta \end{cases} \quad \text{or} \quad (IIb) \begin{cases} Ax \le b \\ x_j \ge \beta + 1 \end{cases}$$

Suppose now that we know a feasible solution $\bar{x} \in \mathbb{Z}^n$ of (I), and suppose that:

$$\max\{c^T x \mid x \text{ sat. (IIa)}\} \le c^T \bar{x}$$

Then we do not need to search for an optimal solution satisfying (IIa) since all integral solution of (IIa) have value $\le c^T \bar{x}$.

<u>The algorithm</u>:

Input : (I) and assume that we have bounds: $-M \le x_i \le M$ for all $i = 1, \dots, n$.

Initialize : $L := \big\{\{Ax \le b\}\big\}$, $\bar{x} := \emptyset$, and $\ell := -\infty$. (ℓ is the best lower bounds so far)

While $L \neq \emptyset$:

- Pick sub problem $A'x \le b'$ from L and delete it.
- Solve the linear program:

$$(\beta) \qquad \begin{array}{rr} & \max c^T x \\ \text{s.t.} & A'x \le b' \\ & x \in \mathbb{R}^n \end{array}$$

  Let x^* be the optimal solution found.
- If $x^* \in \mathbb{Z}^n$ and $c^T x^* > \ell$, then set $\bar{x} = x^*$ and $\ell = c^T x^*$.
- If $x^* \notin \mathbb{Z}^n$ and $c^T x^* > \ell$ then let $j \in \{1, \dots, n\}$ such that $x_j^* \notin \mathbb{Z}$. $L := L \cup \big\{\{A'x \le b', x_j \le \lfloor x_j \rfloor\}, \{A'x \le b', x_i \ge \lceil x_j \rceil\}\big\}$.

Efficiency: The efficiency of the Branch and Bound algorithm depends on the "tightness" of the LP-relaxation. In order to improve the efficiency of this algorithm, we can use the so called cutting planes technique. Branch and Bound + cutting planes give effective methods for (IP).

The fundamental idea behind cutting planes is to add constraints to a linear program until the optimal basic feasible solution takes on integer values. Of course, we have to be careful which constraints we add: we would not want to change the problem by adding the constraints. We will add a special type of constraint called a cutting plane.

Definition 6. A *cutting plane* is an inequality $c^T x \leq \delta$ that is satisfied by all points in $\{x \in \mathbb{Z}^n : Ax \leq b\}$ but not by all points in $\{x \in \mathbb{R}^n : Ax \leq b\}$.

In other words, cutting plane relative to a current fractional solution satisfies the following criteria:

- every feasible integer solution is feasible for the cut
- the current fractional solution is not feasible for the cut.

Example 7. We consider the following integer program:

$$\max \sum_{i=1}^{5} x_i$$

$$\text{s.t.} \quad \begin{pmatrix} 1 & 1 & & & \\ & 1 & 1 & & \\ & & 1 & 1 & \\ & & & 1 & 1 \\ 1 & & & & 1 \end{pmatrix} x \leq \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

$$\begin{array}{rcl} 0 \leq x_i & \leq & 1 \quad \forall i = 1, \ldots, 5 \\ x & \in & \mathbb{Z}^5 \end{array}$$

If we look at the conflict graph of this problem, we obtain a single cycle with nodes $(x_1, x_2, x_3, x_4, x_5)$. That means that $\forall i = 1, \ldots, 5$, $\begin{cases} x_i = 1 \\ x_{i+1} = 1 \end{cases}$ are in conflict (if we define $x_6 = x_1$). Hence, we can see that we can have only two (non-consecutive) nodes equal to 1. Therefore, the optimal solution of the integer program is 2.

Now note that $(\frac{1}{2}, \ldots, \frac{1}{2})$ is a feasible point of the LP-relaxation problem and has value $\frac{5}{2} > 2$ which is the optimal value of the LP. Thus, the inequality $\sum_{i=1}^{5} x_i \leq 2$ is valid for all integer solutions but not for $(\frac{1}{2}, \ldots, \frac{1}{2})$ so it is a cutting plane.

References

[1] Gerard CORNUEJOLS & Reha TÜTÜNCÜ. Optimization Methods in Finance, *Cambridge University Press*, Cambridge (USA), 2007.