Stochastic Programming
Lecture 15

Brian Denton

NC State University

March 31, 2009

Today's Class

1. Research Projects
2. Introduction to Integer Programming
3. Formulating IPs
4. Branch and Bound Method

Outline

1. Research Projects
2. Introduction to Integer Programming
3. Formulating IPs
4. Branch and Bound Method

Independent research projects are **due on December 3**. Presentations are Nov 30 and Dec 2.

If you are unable to present on one of those days **let me know by Friday.**

- Provide your written report in hard and softcopy
- Electronic copy of your in-class presentation (24 hours in advance of class)
- Electronic copy of (well commented) code you developed

Reminder: Projects are worth 50% of the final grade.
Project grade breakdown:

- Proposal: 20%
- Presentation: 10%
- Final project: 70%

Your projects should be in the form of a journal manuscript and include the following sections:

- **Title page** with author contact information and Abstract
- **Introduction** that motivates your project
- **Literature review**
- **Model Formulation** describing the model you propose to study including: source of uncertainty, first and second stage problem description, deterministic equivalent problem, description of special structure, characterization of feasibility (e.g. complete recourse)
- **Methodology** section with a detailed description of the methods you used and why they are suited to the problem you are studying
- **Discussion** of results and findings
- **Conclusions** summarizing most important findings
- **References, Appendices**

Good References on Statistical Approximations

The following are helpful references in understanding Monte Carlo bounds:

- Section 7.3, Kall and Mayer, “Stochastic Linear Programming: Models, Theory, and Computation”, Springer, New York
- Infanger, “Planning Under Uncertainty”, International Thompson Publishing, 1994

Outline

Introduction to Integer Programming

Sources

- Wolsey,L.A., Integer Programming
- Winston,W. L., Operations Research: Applications and Algorithms
- Hillier,F. and Lieberman,G. , Introduction to Operations Research

Introduction to Integer Programming

- An Integer Program(IP) in which all variables are required to be integers is called a **pure integer programming problem**.
- An IP in which only some of the variables are required to be integers is called a **mixed integer programming problem**.
- An IP in which only some of the variables are required to be integers is called a **mixed integer programming problem**.
- An IP in which all variables must equal to 0 or 1 is called a **binary (0 – 1) integer programming problem**.

LP Relaxation

The LP obtained by omitting all integer constraints on variables is called **LP relaxation** of the IP.

- The feasible region for any IP must be contained in the feasible region for the corresponding LP relaxation.
  For a minimization problem:
  $z^*_{LPRelaxation} \leq z^*_{IP}$
- Recall that simplex algorithm proceeds by going from one feasible solution to a better one to solve LPs.
  It examines only a fraction of all basic feasible solutions
  Unlike LPs, extreme points are not necessarily optimal in Integer Programming

Outline

Formulating IPs

Example: Knapsack Problem (Winston,2004 page 479)

Table: Weights and Benefits for Items in Josie's Knapsack

Item	Weight(Pounds)	Benefit
1	5	16
2	7	22
3	4	12
4	3	8

$$x_j = \begin{cases} 1 & \text{if Josie takes item j on the hike} \\ 0 & \text{otherwise.} \end{cases}$$

$$\begin{aligned} &\max \ 16x_1 + 22x_2 + 12x_3 + 8x_4 \\ &\text{s.t.} \\ &\quad 5x_1 + 7x_2 + 4x_3 + 3x_4 \le 14 \\ &\quad x_j = \{0 \ or \ 1\} \ \ (j = 1, 2, 3, 4) \end{aligned}$$

Examples

- **Capital Budgeting Problem**
  Various possible investments to maximize the Net Present Value
  Investment decisions: binary integer variables
  Formulate as a knapsack problem
- **Fixed-Charge Problem**
  Various items to produce to maximize weekly profit
  Whether to produce a certain product: binary integer variable
  Amount produced for each product: continuous variable
- **Set Covering Problem**
  Each member of a set(set 1) must be covered by an acceptable number of some set(set 2)
  Example: Facility Location Problem
  Determine minimum number of fire stations(and their locations) to build such fire stations are minimum of 15 minute of drive from each city

Examples (continued)

- **Bin Packing Problem**
  Objective: to find the smallest number of bins such that the items of varying size can fit into the bins
  Let $j = 1, \ldots, m$ index bins of capacity c (which may or may not be needed) and $i = 1, \ldots, n$ index items of size d_i

$$x_j = \begin{cases} 1 & \text{if bin j is used} \\ 0 & \text{if bin j is not used.} \end{cases}$$

$$y_i j = \begin{cases} 1 & \text{if i assigned to bin j} \\ 0 & \text{otherwise.} \end{cases}$$

$$\begin{aligned} &\max \; x_1 + x_2 + \ldots + x_m \\ &\text{s.t.} \\ &\quad y_{ij} \leq x_j \\ &\quad \textstyle\sum_{j=1}^{m} y_{ij} = 1 \;\; \forall(i,j) \\ &\quad \textstyle\sum_{i=1}^{n} d_i y_{ij} \leq c x_{ij} \;\; \forall(j) \\ &\quad y_{ij}, \; x_j \in \{0, 1\} \end{aligned}$$

Outline

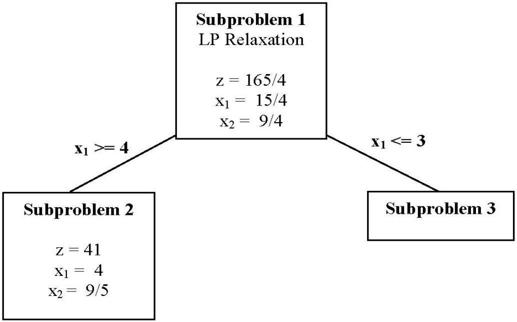
Branch and Bound Method

- Optimal solution is not necessarily an extreme point
- We need a cleverly structured enumeration procedure to find the optimal solution: Branch and Bound Method
- The IP is solved using a tree by solving LP subproblems and eliminating many possible solutions from consideration.
- Subproblems are generated by branching on an appropriately chosen fractional-valued variable until all integer requirements are satisfied.

Branch and Bound Method

- Branch and Bound Method begins with solving the LP relaxation of the IP
    - If the solution to the LP relaxation satisfies integer constraints, then the optimal solution to the LP relaxation is also the optimal solution to the IP
    - Otherwise, we call LP relaxation **subproblem** 1 and start branching on this subproblem
- The display of all subproblems is called a **tree**
- Each subproblem is referred to as a **node** of the tree
- Each line connecting two nodes is called an **arc** which carry the constraint leading from one subproblem to another

Branch and Bound Method



Branch and Bound Method

When further branching on a subproblem cannot yield any useful information, we **fathom** that subproblem.
Following three situations can occur (for a max problem)

- The subproblem is infeasible, thus it cannot yield the optimal solution to the IP.
- The subproblem yields an optimal solution in which all variables have integer values. If this optimal solution has a better z-value than any previously obtained solution that is feasible in the IP, than it becomes a **candidate solution**, and its z-value becomes the current lower bound (LB) on the optimal z-value for the IP.
- The optimal z-value for the subproblem does not exceed (in a max problem) the current LB, so it may be eliminated from consideration.

Branch and Bound Method

Two general approaches are used to determine which subproblem should be solved next

- **LIFO (Backtracking)**chooses to solve the most recently created subproblem
  - Goes down one side of the Branch and Bound tree and quickly finds a candidate solution
  - Then we backtrack our way up to the top of the other side
- **Jumptracking** solves all subproblems created by branching on a node.
  - Then, branches again on the node with the best z-value
  - Often jumps from one side of the tree to another

Jumptracking usually creates more nodes and requires more computer storage, however the idea is to find the best z-value more quickly by moving towards the nodes with good z-values

B&B Method for Solving Mixed Integer Programming Problems

- Some variables are required to be integers, others are allowed to be either integer or nonintegers
- To solve a MIP by the branch-and-bound method, modify the method by branching only on variables that are required to be integers
- For a solution to a subproblem to be a candidate solution, it needs only to have integer values to those variables that are required to be integers

Example

Consider the Knapsack problem we formulated

$$\begin{aligned}
&\max \; 16x_1 + 22x_2 + 12x_3 + 8x_4 \\
&\text{s.t.} \\
&\quad 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14 \\
&\quad x_j = \{0 \; or \; 1\} \;\; (j = 1, 2, 3, 4)
\end{aligned}$$

- solution to the LP relaxation:
  $x_1 = x_2 = 1, x_3 = 0.5$ and $z = 44$
- We branch on the variable which has the fractional value
  $x_3 \leq 0$
  $x_3 \leq 1$

B&B Tree for Knapsack Problem