IO543: Scheduling

Lecture 4

Brian Denton
Associate Professor, IOE

IOE Bld 2893
btdenton@umich.edu

Announcements

- There will be no class or office hours on Thursday
- Assignment 2 due on February 3

Last Time

- In Class Assignments graded out of 3
- These will count for 2/3 of the 15% of your "participation grade"
- The other 1/3 is based on your participation in class and your presentation at the end of the semester

Today

Goals:

- Introduction to integer programming
- Learn the **branch-and-bound** method and use it to solve:
  - Knapsack problem

Integer Programming

- Many scheduling problems can be formulated as **Integer Programs (IPs)**

- Integer programs involve discrete decision variables:

  - Integer decisions
  - Binary decisions (e.g. yes/no decisions)

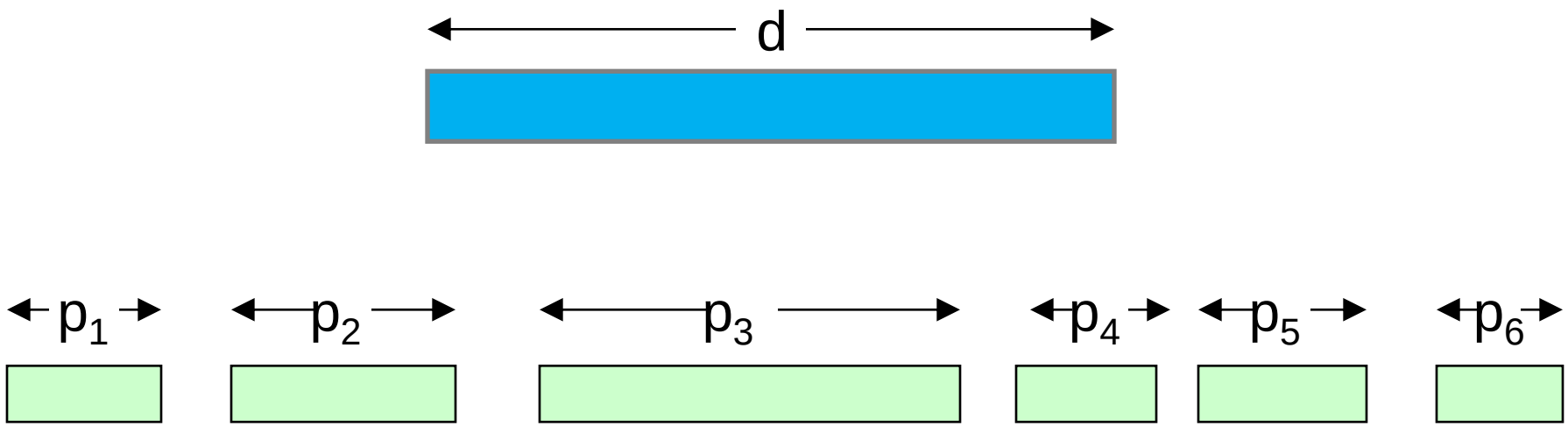
Example: Minimizing Tardy Jobs

- An optimal schedule for $1 \mid \mid \sum U_j$ identifies two sets of jobs: (a) those completed first and on-time; (b) those that are late
  - Minimizing tardy jobs is equivalent to maximizing the percentage of on-time jobs

- The simplest version of this scheduling problem (when all due dates are equal) corresponds to the **knapsack problem**

Review of the Knapsack Problem

Given a set of one-dimensional items what is the largest number that fit in the knapsack?



Review of the Knapsack Problem

Weighted Knapsack Problem

$$\max z = w_1x_1 + w_2x_2 + \cdots + w_nx_n$$
$$\text{s.t.} \quad p_1x_1 + p_2x_2 + \cdots + p_nx_n \leq d$$
$$x_i = 0 \text{ or } 1 \quad (i = 1, 2, \ldots, n)$$

Decision variable: x_i represents the decision to complete the job on time (1) or not (0).

Model parameters: job weights, w_i, job processing times, p_i, due date, d.

Example: Knapsack Problem

Example with 4 jobs and due date of 14 days:

Jobs	1	2	3	4
w_j	16	22	12	8
p_j	5	7	4	3

weighted knapsack problem:

$$\max z = 16x_1 + 22x_2 + 12x_3 + 8x_4$$

$$\text{s.t.} \quad 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14$$

$$x_j = 0 \text{ or } 1 \quad (j = 1,2,3,4)$$

Refresher

Formulate an IP for the following instance of $1 \mid \mid \sum U_j$:

Jobs	1	2	3	4
w_j	16	22	12	8
p_j	5	7	4	3

Due Date = 14 days

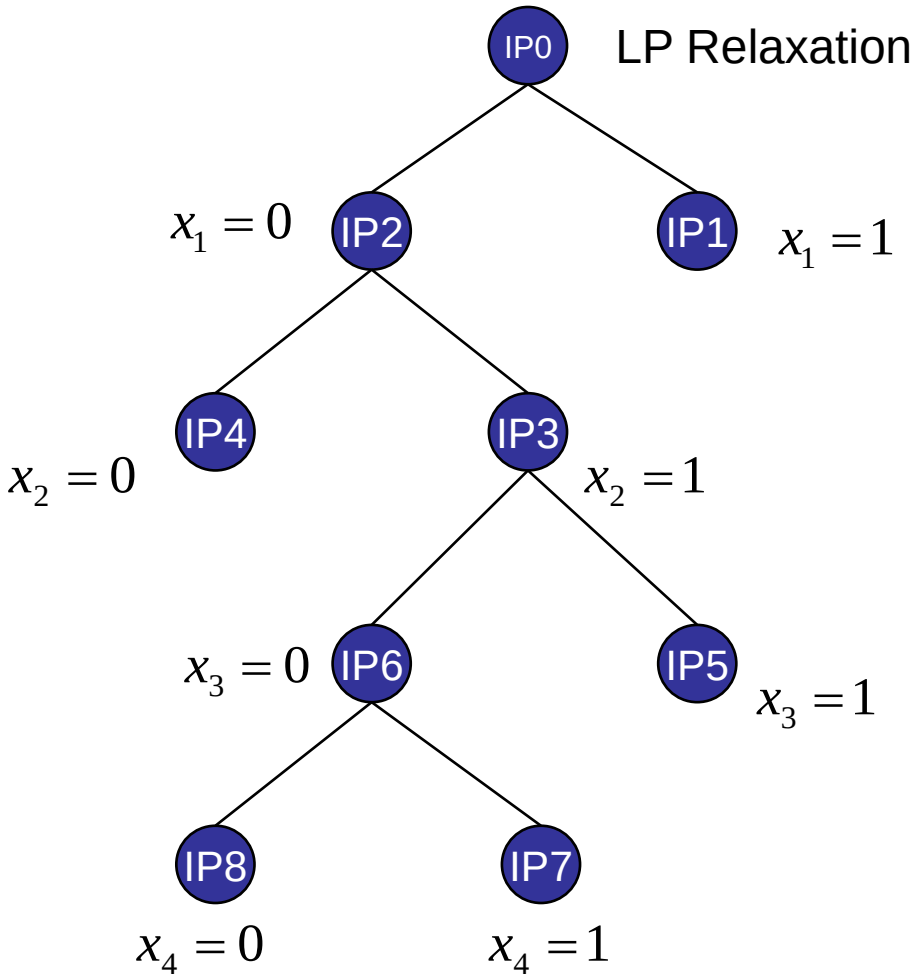
Additional requirements: If J1 is selected then J2 must be selected. If J3 is selected then J4 cannot be selected and vice versa.

The Branch-and-Bound Method

Integer programs can be solved by **branch-and-bound**

- Branch-and-bound **implicitly** enumerates all solutions using a **binary tree**
- Subproblems are generated by **branching** on selected fractional-valued variables until all variables are integer
- By solving **LP subproblems** many branches may be eliminated from consideration

Binary Tree


IP0
LP Relaxation
$x_1 = 0$
IP2
IP1
$x_1 = 1$
IP4
IP3
$x_2 = 0$
$x_2 = 1$
$x_3 = 0$
IP6
IP5
$x_3 = 1$
IP8
IP7
$x_4 = 0$
$x_4 = 1$

Fathoming Nodes

Given a **max problem**, the following three situations result in a subproblem being **fathomed**:

1. The subproblem is infeasible
2. The optimal *z*-value for the subproblem does not exceed the current lower bound (LB)
3. The subproblem yields an optimal solution in which all variables have integer values (this becomes a **candidate solution**, and its *z*-value is the new LB)

Selecting Subproblems

There are many rules for deciding which subproblem to solve next

Examples:

- **Depth first search** selects the most recent subproblem
- **Best Bound** selects the subproblem that appears most promising

Example

Use **branch-and-bound** to find the optimal solution to the following instance of $1 \mid \mid \sum U_j$ is:

Jobs	1	2	3	4
w_j	16	22	12	8
p_j	5	7	4	3

Due Date = 14 days

Example: Getting Started

$$\max z = 16x_1 + 22x_2 + 12x_3 + 8x_4$$
$$\text{s.t.} \quad 5x_1 + 7x_2 + 4x_3 + 3x_4 \le 14$$
$$\cancel{x_i = 0 \text{ or } 1 \quad (i = 1,2,3,4)}$$
$$0 \le x_i \le 1$$

- Drop the binary constraints and solve the **LP relaxation**:

$$x_1 = x_2 = 1,\ x_3 = 0.5,\ z = 44$$

- Choose a fractional variable to branch on and create two subproblems:

$$x_3 = 0$$
$$x_3 = 1$$

- Solve the two subproblem relaxations

Example: Continued

Subproblem 1
$x_1 = x_2 = 1$, $x_3 = 0.5$, $x_4 = 0$, **z = 44**

$x_3 = 0$

Subproblem 2
$x_1 = x_2 = 1$, $x_3 = 0$, $x_4 = 2/3$, **z = 43$_{1/3}$**

$x_3 = 1$

Subproblem 3
$x_1 = x_3 = 1$, $x_2 = 5/7$, $x_4 = 0$, **z = 43$_{5/7}$**

Example: Continued

Subproblem 1
$x_1 = x_2 = 1$, $x_3 = 0.5$, $x_4 = 0$, **z = 44**

$x_3 = 0$

Subproblem 2
$x_1 = x_2 = 1$, $x_3 = 0$, $x_4 = 2/3$, **z = 43$_{1/3}$**

$x_3 = 1$

Subproblem 3
$x_1 = x_3 = 1$, $x_2 = 5/7$, $x_4 = 0$, **z = 43$_{5/7}$**

$x_4 = 0$

Subproblem 8
$x_1 = x_2 = 1$, $x_3 = x_4 = 0$, **z = 38**
(Candidate solution)

$x_4 = 1$

Subproblem 9
$x_1 = x_4 = 1$, $x_2 = 6/7$, $x_4 = 0$, **z = 42$_{6/7}$**

$x_2 = 0$

Subproblem 4
$x_1 = x_3 = 1$, $x_2 = 0$, $x_4 = 1$, **z = 36**
(Candidate solution)

$x_2 = 1$

Subproblem 5
$x_2 = x_3 = 1$, $x_1 = 3/5$, $x_4 = 0$, **z = 43$_{2/5}$**

$x_1 = 0$

Subproblem 6
$x_2 = x_3 = x_4 = 1$, $x_1 = 0$, **z = 42**
(Candidate solution)

$x_1 = 1$

Subproblem 7
Infeasible

Example: Solution

Therefore the optimal schedule is:

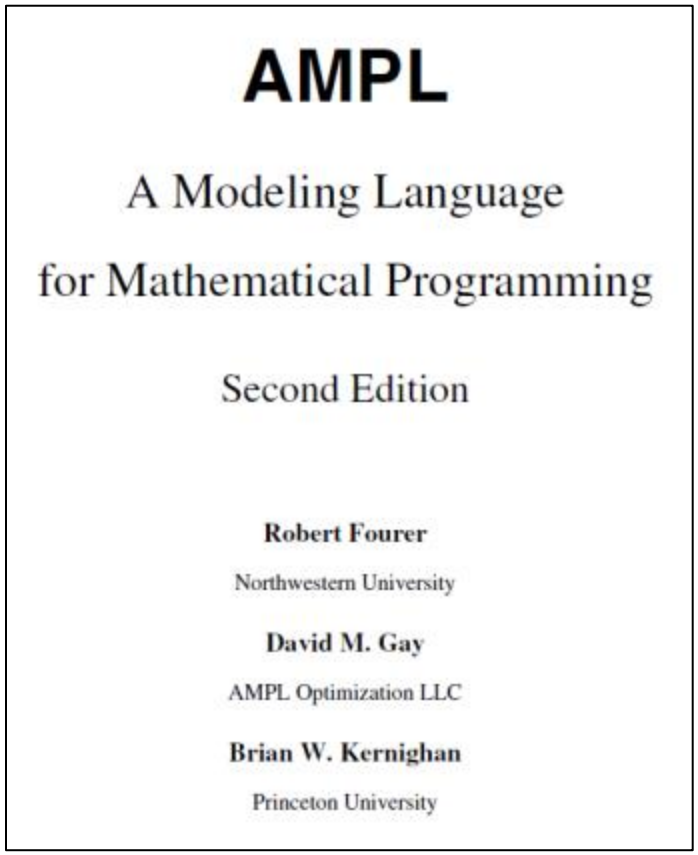
<u>Optimal Solution:</u>
On-time jobs: 2,3,4
Late jobs: 1
$Z^* = 42$

AMPL is a powerful solver that can be used to solve problems like this automatically

AMPL

AMPL Resources on IO543 CTools site:

- AMPL software for Mac and Windows systems
- AMPL book
- IOE510 Tutorial on AMPL


AMPL

A Modeling Language

for Mathematical Programming

Second Edition

Robert Fourer

Northwestern University

David M. Gay

AMPL Optimization LLC

Brian W. Kernighan

Princeton University

Next Time

Section 3.3 covers $1 \mid \mid \sum U_j$

For next time read section 3.2 and 3.3 of Pinedo.