Artificial Intelligence | **Spring 2005**
Professor Greenwald | **Lecture #8**

Mathematical Programming[1]

Linear Programming
Integer Programming
Stochastic Programming

A *mathematical program* is an optimization problem of the form:

$$\min_{\mathbf{x} \in \mathbf{R^n}} f(\mathbf{x})$$

subject to

$$\mathbf{A}\mathbf{x} \leq \mathbf{b}$$

where

- $\mathbf{A} = (a_{ij})_{1 \leq i \leq m, 1 \leq j \leq n}$ is a matrix of real-valued constants
- $\mathbf{b} = (b_j)_{1 \leq j \leq m}$ is a vector of real-valued constants
- $\mathbf{x} = (x_i)_{1 \leq i \leq n}$ is a vector of variables
- f is a function

If f is linear, then the program is called a *linear* program. If f is linear, and for all i, $x_i \in \mathbb{Z}$, then the program is called an *integer* linear program. If f is linear, and for some i, $x_i \in \mathbb{R}$, while for other i, $x_i \in \mathbb{Z}$, then the program is called a *mixed* integer linear program. If f is linear, and for all i, $x_i \in \{0, 1\}$, then the program is called a *binary* integer linear program.

1 Linear Programming

The Diet Problem Consider a set of m vitamins contained in a set of n foods. The minimum daily requirement of vitamin i is b_i, for all $1 \leq i \leq m$. The cost (in dollars or calories) per unit of food j is c_j, for all $1 \leq j \leq n$. The number of units of vitamin i contained in each unit of food j is a_{ij}. How much of each food should be consumed so as to minimize cost without failing to satisfy the minimum daily requirements?

Solution If x_j is the number of units of food j that one consumes daily, then the objective is to minimize $c^T x$. But at the same time, one must satisfy the minimum daily requirement. The term $a_{ij}x_j$ denotes the number of units of vitamin i that are consumed in x_j units of food j. Summing over all foods j yields the constraint $A_i x \geq b_i$, for all vitamins i. The linear program that solves the diet problem can be stated as follows:

$$\begin{aligned} & \min_{x_j} \sum_{j=1}^{n} c_j x_j \\ s.t. \quad & \sum_{j=1}^{n} a_{ij} x_j \geq b_i, \quad \forall i \\ & x_j \geq 0, \quad \forall j \end{aligned}$$

The Production Problem A company manufactures n products using m resources. The company has in its inventory b_i units of resource i, for all $1 \leq i \leq m$, and it earns c_j profits by producing one unit of product j, for all $1 \leq j \leq n$. The number of units of resource i required to produce one unit of product j is a_{ij}. How much of each product should be manufactured so as to maximize total profits, given the stated resource constraints?

Solution If x_j is the number of units of product j that is produced, then the objective is to maximize $c^T x$. But the company's resources are limited. The term $a_{ij}x_j$ denotes the number of units of resource i that is necessary to manufacture x_j units of product j Summing over all products j yields the constraint $A_i x \leq b_i$, for all resources i. The linear program that solves the production problem can be stated as follows:

$$\begin{aligned} & \max_{x_j} \sum_{j=1}^{n} c_j x_j \\ s.t. \quad & \sum_{j=1}^{n} a_{ij} x_j \leq b_i, \quad \forall i \\ & x_j \geq 0, \quad \forall j \end{aligned}$$

2 Integer Linear Programming

The capital budgeting problem is the binary version of the production problem: that is, $x_j \in \{0, 1\}$ indicates whether or not product j is produced (or the jth investment opportunity is selected). The knapsack problem is the special case of capital budgeting where $m = 1$.

Knapsack Problem Given n objects s.t. object i has value v_i but weighs w_i. Find an optimal set of objects to pack in a knapsack of capacity k.

LP Solution Let $x_i \in \mathbb{R}$ denote the quantity of object i that is packed in the knapsack.

$$\max_{x_i} \sum_i v_i x_i$$

subject to

$$\sum_i w_i x_i \leq k$$

and

$$x_i \geq 0, \quad \forall i$$

ILP Solution Define integer variable x_i that indicates whether or not object i is packed in the knapsack. Add the constaint:

$$x_i \in \mathbb{Z}, \quad \forall i$$

Binary ILP Solution Define binary variable x_i that indicates whether or not object i is packed in the knapsack. Replace the constraint $x_i \in \mathbb{Z}$ with the constraint $x_i \in \{0, 1\}$.

Assignment Problem Given n jobs and n machines. The cost of machine i serving job j is c_{ij}. Each machine can execute exactly one job. All jobs must be completed. Assign jobs to machines so as to minimize cost.

Solution Define binary variables $x_{ij} \in \{0, 1\}$ that indicates whether or not machine i serves job j.

$$\max_{x_{ij}} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$$

subject to

$$\sum_{i=1}^{n} x_{ij} = 1, \quad \forall j$$

$$\sum_{j=1}^{n} x_{ij} = 1, \quad \forall i$$

$$x_{ij} \in \{0, 1\}$$

The first set of constraints states that every job is served by some machine. The second set of constraints states that every machine serves some job.

2.1 Sample Constraints

Set Covering Constraints

$$\sum_j x_j \geq 1$$

Set Packing Constraints

$$\sum_j x_j \leq 1$$

Set Partitioning Constraints

$$\sum_j x_j = 1$$

2.2 ILP is NP-Hard

Reduction from an instance of 3CNF to an instance of ILP.

Roughly: Boolean variables map to binary variables; Clauses map to constraints.

Solution 1 Given an instance of 3CNF, construct variables x_i and $\overline{x}_i$ for each variable w_i that appears in this instance of 3CNF.

All variables are boolean: $x_i, \overline{x_i} \in \{0, 1\}$

Exactly one of the two variables x and $\overline{x}$ is true: $1 \leq x_i + \overline{x}_i \leq 1$

For each clause $\{z_1, z_2, z_3\}$, construct constraint $y_1 + y_2 + y_3 \geq 1$, where $y_i = x_j$ if $z_i = w_j$, and $y_i = \overline{x}_j$ if $z_i = \neg w_j$.

A solution to this ILP is a solution to 3CNF: In any solution to this ILP, all variables are set to either 0 or 1. If $x_i = 1$ (and $\overline{x}_i = 0$), then set $v(w_i) = t$; otherwise, if $x_i = 0$ (and $\overline{x}_i = 1$), then set $v(w_i) = f$. All clauses are satisfied. The assignment is legal: no variable and its complement are both true.

Solution 2 Given an instance of 3CNF, construct variable x_i for each variable w_i that appears in this instance of 3CNF.

All variables are boolean: $x_i \in \{0, 1\}$.

For each clause $\{z_1, z_2, z_3\}$, construct constraint $y_1 + y_2 + y_3 \geq 1$, where $y_i = x_j$ if $z_i = w_j$, and $y_i = 1 - x_j$ if $z_i = \neg w_j$.

Example 1: $\{w_1, w_2, w_3\} \mapsto x_1 + x_2 + x_3 \geq 1$.

Example 2: $\{\neg w_1, w_2, w_3\} \mapsto (1 - x_1) + x_2 + x_3 \geq 1$,
equivalently $-x_1 + x_2 + x_3 \geq 0$.

Example 3: $\{\neg w_1, \neg w_2, \neg w_3\} \mapsto (1 - x_1) + (1 - x_2) + x_3 \geq 1$,
equivalently $-x_1 - x_2 + x_3 \geq -1$.

Example 4: $\{\neg w_1, \neg w_2, \neg w_3\} \mapsto (1 - x_1) + (1 - x_2) + (1 - x_3) \geq 1$,
equivalently $-x_1 - x_2 - x_3 \geq -2$.