Algorithmic Modeling and Complexity **Fall 2003**

Lecture 5: September 16

Lecturer: J. van Leeuwen *Scribe: E. Maes*

5.1 Overview

This lecture is about designing 'algorithmic models' using (versions of) Linear Programming. Until now, networks were used tot model problems. It would be great to have a generic framework in which problems can be 'programmed' and solved by standard means. The formalism of Linear Programming comes close. We will emphasize *Integer Linear Programming* (ILP) and show how some common problems like the dominating set problem and the vertex cover problem can be seen as ILPs.

5.2 Linear Programs

A Linear Program (LP) is about maximizing or minimizing a linear goal function under a set of linear constraints.

minimize $\quad z = cx \quad$ (linear goal function)

subject to

$Ax \geq b \qquad$ (linear constraints)
$x \geq 0$

where x, c and b are vectors, and A is a matrix with

$x = (x_1, \cdots, x_n)$
$c = (c_1, \cdots, c_n)$
$A = m \times n$ matrix
$b = (b_1, \cdots, b_m) \quad$ (constraint vector)

In coordinate form this can also be written as:

$$z = cx \Leftrightarrow z = \sum_{i=1}^{n} c_i x_i$$

$$Ax \geq b \Leftrightarrow \sum_{j=1}^{n} a_{ij} x_j \geq b_i \text{ (for all } 1 \leq i \leq m)$$

Definition 5.1 *Any x that satisfies the constraints $(Ax \geq b, x \geq 0)$ is called a feasible solution.*

Proposition 5.2 *The feasible solutions of a Linear Program form a convex set $\subseteq \mathbb{R}^n$, the so-called LP-polytope. The LP-polytope has finitely many facets and extreme points.*

The 'polytope' of a given LP is not always finitely bounded, and may be 'open' to infinity.

Solving an LP leads to finding a point of the LP-polytope where the goal function achieves its minimum (or maximum): if the minimum is finite, this will happen in (at least one of) the extreme points. In higher dimensions the number of extreme points can become exponential (in m). In the figure below a preview of a (2-dimensional) linear program is shown.

$$\min \quad z = x_1 + x_2$$

subject to

$$\begin{aligned} -2x_1 &\geq -3 \\ -2x_2 &\geq -6 \\ 2x_1 + x_2 &\geq 1 \\ x_1, x_2 &\geq 0 \end{aligned}$$

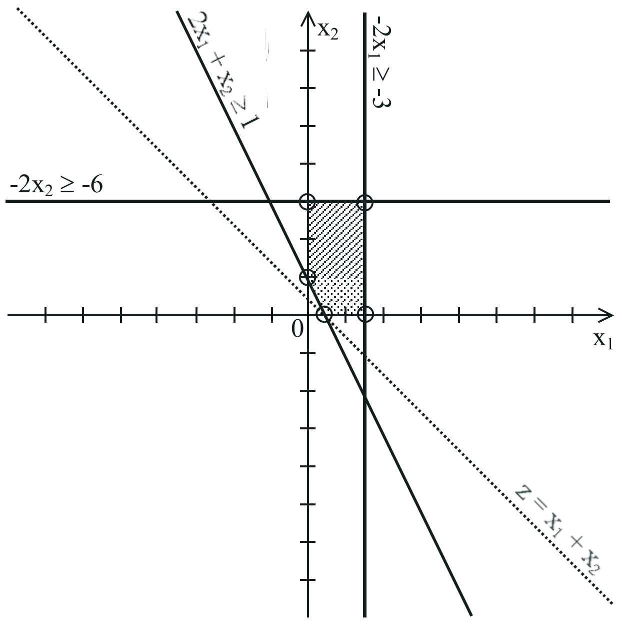

Figure 1: A preview of a Linear Program

The circles indicate the extreme points of the face: $(0, 3), (\frac{3}{2}, 3), (\frac{3}{2}, 0), (\frac{1}{2}, 0), (0, 1)$. The face bounded by these points contains the feasible solutions. The minimum of the goal function is the point $(\frac{1}{2}, 0)$.

We will especially consider Linear Programs with further constraints:

- *Integer Linear Programs* (ILP)
  $x \geq 0$ replaced by: $x \geq 0$ and x_i integer (for all $1 \leq i \leq n$).
- *Binary* or *0-1 Linear Programs* (0-1 LP)
  $x \geq 0$ replaced by: $x_i \in \{0, 1\}$ (for all $1 \leq i \leq n$).

These types of constraints normally arise when the 'x_i' are interpreted as 'quantities' or 'decisions (yes/no)'.

LPs are routinely solved using standard software packages that implement the *simplex method* (viz. CPLEX) or any of the newer *interiorpoint methods*. Solving ILPs or 0-1 LPs is not done in the same way as solving LPs, because the extreme points of the underlying LP-polytope might not be integral.

Replacing the constraint '$x \geq 0, x_i$ integer' by $x \geq 0$ in an ILP or the constraint '$x_i \in \{0, 1\}$' by '$0 \leq x \leq 1$' in a 0-1 LP is called *relaxing* the problem.

Exercise. Let OPT be the optimum of an ILP or 0-1 LP (minimization version), let $z(x^*)$ be the optimum of the relaxed problem, and let $z(y)$ be the value of the goal function after *rounding* the solution x^* of the relaxed problem to a *feasible* integer or 0-1 solution y of the original problem. Then $z(x^*) \leq OPT \leq z(y)$.

5.3 Modeling problems as ILPs

Many problems that were encountered in previous lectures can be modeled as an ILP or 0-1 LP. This is shown for: Set Cover, Dominating Set, Vertex Cover, and their extensions to 'weighted' networks.

5.3.1 The Set Cover problem

The (Minimum Cost) Set Cover problem is defined as follows:

- given a universe $U = \{1, \cdots, n\}$,
- a collection of subsets $S = \{S_1, \cdots, S_m\}$ with $S_i \subseteq U$, and
- a cost function $c : S \rightarrow \mathbb{Q}_+$ where $c(S_i) = c_i$,
- determine a minimum cost subcollection ('cover') $S_{j_1}, \cdots, S_{j_k}$ of S such that

$$\bigcup_{i=1}^{k} S_{j_i} = U.$$

For example: U is a collection of tasks, S_i is the set of tasks which contractor i is able to carry out, and c_i is the cost that contractor i charges for it. The goal in this case is to determine a set of k contractors that can do all tasks for the least total cost $\sum_{i=1}^{k} c_{j_i} = |U|$.

Note that some tasks may be 'covered' by more than one contractor. In the Set *Packing* problem it is required that every element is covered only once.

We now design a *model* for the Set Cover problem. There is a decision to be made for every subset S_i, namely whether to include it in the cover or not. Therefore decision variables (or 'indicator variables') will be used:

x_i: decision variable indicating whether S_i is in the solution or not.

This lead to the following 0-1 LP:

(SCP)

$$\min \quad z = \sum_{i=1}^{n} c_i x_i$$

subject to

$$\sum_{j=1}^{m} a_{ij} x_j \geq 1 \text{ (for all } 1 \leq i \leq n)$$

$$x_i \in \{0, 1\} \text{ (for all } 1 \leq i \leq m).$$

where

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

We will always assume that $\bigcup_{i=1}^{k} S_i = U$, otherwise the problem has no feasible solution.

The Set Cover problem has been well-studied since the mid-1960's. There are various algorithms for solving it exactly, although none of these algorithms is polynomial in n and m.

Exercise. Consider the relaxation of SCP in which the constraint $x_i \in \{0, 1\}$ is replaced by: $x_i \geq 0$ and integer. Show that the relaxed version has the same optimum solution as SCP. (Hint: consider what would happen when $x_i > 1$ for some i in the optimum solution.)

5.3.2 The Dominating Set problem in weighted networks

Let $G =< V, E >$ be a network. We first repeat the definition of a dominating set: a set S of vertices in a network is a dominating set if every vertex not in S is adjacent to a vertex in S.

Now consider the case that *weights* are attached to the nodes. Say weight w_i is attached to node i. The (Minimum Weight) Dominating Set problem asks for a dominating set in G of least total weight.

The Dominating Set problem can be modeled by a 0-1 LP. Use:

x_i: decision variable expressing whether node i is in the dominating set or not .

Then design the following model:

(DSP)

$$\min \quad z = \sum_{i=1}^{n} w_i x_i$$

subject to

$$\sum_{j=1}^{n} a_{ij} x_j \geq 1 \text{ (for all } 1 \leq i \leq n).$$

$$x_i \in \{0, 1\} \text{ (for all } 1 \leq i \leq n).$$

where

$$a_{ij} = \begin{cases} 1 & \text{if } i = j; \\ 1 & \text{if i and j are connected by an edge;} \\ 0 & \text{otherwise.} \end{cases}$$

Theorem 5.3 *The Minimum Weight Dominating Set problem is 'equivalent' to the Minimum Cost Set Cover problem.*

Proof: We show that every instance of one problem can be easily transformed into an equivalent instance of the other problem.

($\Rightarrow$) Consider the Minimum Weight Dominating Set problem D_G for network $G =< V, E >$ and assume w.l.o.g. that $V = \{1, \cdots, n\}$. Create a Minimum Cost Set Cover problem S_G with universe $V = \{1, \cdots, n\}$ and subsets S_i such that $S_i =$ 'i together with all its neighbors in G'. Let $c(S_i) = w_i$, the weight of node i.

The minimum set cover solutions are related in a 1-1 fashion to the minimum weight dominating set solutions.

($\Leftarrow$) Consider an instance S_U of the Minimum Cost Set Cover problem with $U = \{1, \cdots, n\}$, $S = \{S_1, \cdots, S_m\}$ and $c(S_i) = c_i$. Assume w.l.o.g. that $c_i > 0$ for all i and $\cup_{i=1}^{m} S_i = U$.

Create a network G as shown in Figure 2. There is a special top-node u_0 with weight 0. It is connected to nodes $u_1, \ldots, u_m$ where u_i has weight c_i and 'corresponds' to subset S_i. Finally there are nodes $v_1, \ldots, v_n$ corresponding to the elements of U and an edge (u_i, v_j) if and only if S_i contains j. The nodes v_j are given very large weights so they cannot possibly be chosen in a minimum weight dominating set of G: a weight of $2 \sum_{i=1}^{m} c_i$ will do.

Consider a feasible solution $(S_{j_1}, \ldots, S_{j_k})$ to S_U of cost C. Then the nodes $u_0, u_{j_1}, \ldots, u_{j_k}$ form a solution of cost C to the dominating set problem in G. (Node u_0 is needed to dominate the nodes u_j for which S_j is not in the set cover.)

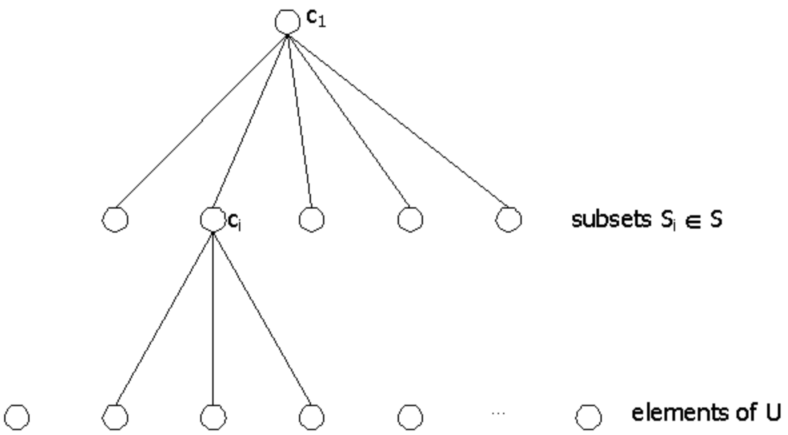

Figure 2: Minimum Cost Set Cover versus Minimum Weight Dominating Set

Conversely, consider a minimum weight solution to the Dominating Set problem in G. This solution will have weight $\leq \sum_{i=1}^{m} c_i$, as the set $u_1, \ldots, u_m$ is a feasible solution of this weight. Thus no v-node can belong to the dominating set. The solution to the Dominating Set problem thus consists entirely of u-nodes. The u_i-nodes in it with $i > 0$ easily correspond to a feasible set cover of S of the same cost. ■

5.3.3 The Vertex Cover problem on weighted networks

Let $G =< V, E >$ be a network, with $|V| = n$. A vertex cover C is a subset of V, such that for each edge $(u, v) \in E$, at least one of u and v is in C.

Again we consider the case that *weights* are attached to the nodes. Say weight w_i is attached to node i. The (Minimum Weight) Vertex Cover problem asks for a vertex cover in G of least total weight.

The Vertex Cover problem can be modeled by a 0-1 LP. Use:

x_i: decision variable expressing whether node i is in the vertex cover or not.

Then design the following model:

$$\min \quad z = \sum_{i=1}^{n} w_i x_i$$

subject to

$$x_i + x_j \geq 1 \text{ for all edges } (i, j) \in E.$$
$$x_i \in \{0, 1\} \ (1 \leq i \leq n).$$

Theorem 5.4 *The Minimum Weight Vertex Cover problem 'reduces' to the Minimum Cost Set Cover problem.*

Proof: We show that instances of the Minimum Weight Vertex Cover problem can easily be transformed into equivalent instances of the Minimum Cost Set Cover problem.

Consider the Minimum Weight Vertex Cover problem on $G = (V, E)$. Make a set cover problem where $U = \{$ the set of edges of $G\}$ and subsets $S_i =$ 'all edges incident to node i'. Let $c(S_i) = w_i$.

One easily sees that the a weight-w solution to the Vertex Cover problem on G corresponds to a cost-w solution to constructed set cover problem and vice versa. ■

5.3.4 The Vertex Cover polytope

Look at the constraints of the Vertex Cover problem. Consider the LP-polytope defined by the *relaxed* constraints:

$$x_i + x_j \geq 1 \text{ for every } (i, j) \in E$$

$$0 \leq x_i \leq 1 \text{ for all } 1 \leq i \leq n.$$

Definition 5.5 *A feasible solution x is called half-integral if its coordinates are* $\in \{0, \frac{1}{2}, 1\}$

Lemma 5.6 (Nemhauser and Trotter, 1973) *All extreme points of the Vertex Cover polytope are half-integral.*

Proof: Take any feasible solution x that is *not* half-integral. We show it is not an extreme point.

Relate the solution $x = (x_1, \ldots, x_n)$ to the nodes of the network. For some nodes i we will have $x_i \in \{0, \frac{1}{2}, 1\}$ but not for all. Define:

$$V_+ = \{i \mid \tfrac{1}{2} < x_i < 1\},$$
$$V_- = \{i \mid 0 < x_i < \tfrac{1}{2}\}.$$

Thus $V_+ \cup V_-$ is not empty. Now consider two proposed new solutions to the relaxed Vertex Cover problem:

$$y_i = \begin{cases} x_i + \epsilon & \text{if } x_i \in V_+; \\ x_i - \epsilon & \text{if } x_i \in V_-; \\ x_i & \text{otherwise} \end{cases}$$

$$z_i = \begin{cases} x_i - \epsilon & \text{if } x_i \in V_+; \\ x_i + \epsilon & \text{if } x_i \in V_-; \\ x_i & \text{otherwise} \end{cases}$$

For ϵ *small* enough but > 0, we can guarantee that $0 \leq y, z \leq 1$. Clearly $y, z \neq x$.

Note that $\frac{1}{2}y + \frac{1}{2}z = x$, thus x cannot be an extreme point provided y, z are elements of the polytope, i.e. are feasible.

Claim 5.7 *For ϵ small enough > 0 we can achieve that y and z are feasible.*

Proof: Verify the covering constraints: $x_i + x_j \geq 1$. For small enough ϵ this remains true for y and z as well, by the following argument. There are two cases.

Case 1. $x_i + x_j > 1$. The values $y_i + y_j$ and $z_i + x_j$ are at most by $2 \cdot \epsilon$ smaller than $x_i + x_j$ so for small enough ϵ they remain .1.

Case 2. $x_i + x_j = 1$. This leads to the following subcases:

Case 2.1. $x_i = 0, x_j = 1$ so identical for y, z.

Case 2.2. $x_i = 1, x_j = 0$ so identical for y, z.

Case 2.3. $x_i = \frac{1}{2}, x_j = \frac{1}{2}$ so identical for y, z.

Case 2.4. $x_i \in V_-, x_j \in V_+$ or $x_i \in V_+, x_j \in V_-$. Then $(x_i - \epsilon) + (x_j + \epsilon) = 1$ thus the constraint is identically satisfied for y and likewise for z in this case as well.

So for small enough ϵ, both constraints are satisfied. ∎

This completes the proof. ∎

The lemma leads to an intriguing approximation algorithm for the *weighted* vertex cover problem.

Theorem 5.8 *The Minimum Weight Vertex Cover problem can be solved by an efficient algorithm with performance ratio 2.*

Proof: The algorithm simply is: *find any extreme point x of the Vertex Cover polytope.* (This can be done e.g. by using the Simplex algorithm to solve the relaxed Minimum Weight Vertex Cover problem, i.e. the problem with $x_i \in \{0, 1\}$ replaced by $0 \leq x_i \leq 1$.)

Let x be a solution: by the Nemhauser-Trotter Lemma, x is half-integral. Now round x to a solution y as follows:

- if $x_i \in \{\frac{1}{2}, 1\}$ then set $y_i = 1$.
- if $x_i \in \{0\}$, then set $y_i = 0$ otherwise.

Now y is feasible. For any edge $(i, j) \in E$ we have $x_i + x_j \geq 1$ and thus at least one of x_i, x_j is $\in \{\frac{1}{2}, 1\}$. Hence at least one of y_i, y_j is 1 and we have $y_i + y_j \geq 1$. Feasibility of y follows.

But as a 0-1 solution, y is a feasible solution to the original Vertex Cover problem.

Let the relaxed problem have optimum $z^* = \sum_{i=1}^{n} w_i x_i$. Let the Minimum Weight Vertex Cover problem have optimum OPT. Then

$$z^* \leq OPT \leq z(y) \leq 2 \cdot z^* \leq 2 \cdot OPT.$$

■

Whereas the Nemhauser-Trotter Lemma shows how the combinatorial understanding of the LP-polytope of a problem can help, a different rounding trick can circumvent it in the present case. Relax the problem by replacing $x_i \in \{0, 1\}$ by: $x \geq 0$. Consider any optimum solution x of the relaxed problem. (One easily argues that $0 \leq x_i \leq 1$ for every $1 \leq i \leq n$ but we do not even need this observation). Now *round* x to a 0-1 solution y as follows:

- if $x_i \geq \frac{1}{2}$ then set $y_i = 1$.
- if $x_i < \frac{1}{2}$ then set $y_i = 0$.

Exercise. Show that y is a feasible solution to the Vertex Cover problem and that $z(y)$ is within a factor of 2 from optimum. (Hint: by the same argument as in Lemma 5.6.)

5.4 Other ILP models

Many other problems can be modeled as ILP problems. This includes e.g. the (many variants of the) Vehicle Routing Problem.

References

[1] G. Nemhauser, L.E. Trotter. Vertex packings: Structural properties and algorithms. *Math. Programming* 8 (1975) 232-248.