Title

Computer Science 604

Advanced Algorithms

Lecture 7: More Linear Programming

David Juedes

School of EECS

juedes@cs.ohiou.edu

Using the GLPK

See `glpk-4.9/doc` for documentation on the GNU Linear Programming Toolkit.

Introduction (From the GLPK manual)

GLPK assumes the following formulation of *linear programming (LP)* problem:

minimize (or maximize)

$$Z = c_1 x_{m+1} + c_2 x_{m+2} + \ldots + c_n x_{m+n} + c_0 \tag{1.1}$$

subject to linear constraints

$$\begin{aligned} x_1 &= a_{11} x_{m+1} + a_{12} x_{m+2} + \ldots + a_{1n} x_{m+n} \\ x_2 &= a_{21} x_{m+1} + a_{22} x_{m+2} + \ldots + a_{2n} x_{m+n} \\ &\ldots\ldots\ldots\ldots\ldots\ldots \\ x_m &= a_{m1} x_{m+1} + a_{m2} x_{m+2} + \ldots + a_{mn} x_{m+n} \end{aligned} \tag{1.2}$$

Cont'd

and bounds of variables

$$\begin{array}{c} l_1 \leq x_1 \leq u_1 \\ l_2 \leq x_2 \leq u_2 \\ \dots\dots\dots \\ l_{m+n} \leq x_{m+n} \leq u_{m+n} \end{array} \qquad (1.3)$$

where: $x_1, x_2, \dots, x_m$ — auxiliary variables; $x_{m+1}, x_{m+2}, \dots, x_{m+n}$ — structural variables; Z — objective function; $c_1, c_2, \dots, c_n$ — objective coefficients; c_0 — constant term ("shift") of the objective function; $a_{11}, a_{12}, \dots, a_{mn}$ — constraint coefficients; $l_1, l_2, \dots, l_{m+n}$ — lower bounds of variables; $u_1, u_2, \dots, u_{m+n}$ — upper bounds of variables.

GLPK, cont'd

Auxiliary variables are also called *rows*, because they correspond to rows of the constraint matrix (i.e. a matrix built of the constraint coefficients). Analogously, structural variables are also called *columns*, because they correspond to columns of the constraint matrix.

The LP formulation for Vertex Cover

Assume that we have a graph $G = (V, E)$ with m edges $e_1, \ldots, e_m$, and vertices $v_1, \ldots, v_n$. Then, the standard linear programming version of the Minimum Vertex Cover problem is written as follows.

Find integers $x_i \in \{0, 1\}$ that minimizes $\sum_{i=1}^{n} x_i$ subject to the constraints that $x_u + x_v \geq 1$ for each edge $\{u, v\} \in E$. The relaxed version replaces $x_i \in \{0, 1\}$ with $x_i \in [0, 1]$, i.e., $0 \leq x_i \leq 1$.

Coding the VLP in GLPK

Coding this relaxed version of the Minimum Vertex Cover problem as a linear program in VLP in GLPK as follows. Use $m+n$ variables x_i, where for each edge $e_i = \{v_j, v_k\}$, $x_i = x_{m+j} + x_{m+k}$. For each $1 \leq i \leq m$, $1 \leq x_i \leq 2$, and for each $1 \leq j \leq n$, $0 \leq x_{m+j} \leq 1$. These are the *bounds* of the variables.

Half Integral Solutions

Getting back to relaxed integer linear programming version of the Minimum Vertex Cover problem (LPVC).

We were proving the following result last time:

Claim: There exists an solution to LPVC such that $\vec{x}^* \in \{0, 1/2, 1\}^n$.

Proof

Assume that $\vec{x}$ is an optimal solution to the LPVC.

Now, look at the values

- $D_x = \{j \mid 0 < x_j < 1/2\}$, and
- $U_x = \{j \mid 1/2 < x_j < 1\}$.

If $|U_x \cup D_x| = 0$, we are done. Otherwise, set
$\epsilon = \min(\{x_j | j \in D_x\} \cup \{1/2 - x_j | j \in D_x\} \cup \{x_j - 1/2 | j \in U_x\} \cup \{1 - x_j | x_j \in U_x\})$.

Cont'd

By our choice of ϵ, if $j \in D_x$, then $0 \leq x_j - \epsilon < x_j + \epsilon \leq 1/2$. Similarly, if $j \in U_x$, then $1/2 \leq x_j - \epsilon < x_j + \epsilon \leq 1$.

Now, we define two other possible solutions, $\vec{y}$ and $\vec{z}$ to VLP, where

$$y_j = \begin{cases} x_j - \epsilon & \text{if } j \in D_x \\ x_j + \epsilon & \text{if } j \in U_x \\ x_j & \text{otherwise,} \end{cases}$$

and

$$z_j = \begin{cases} x_j + \epsilon & \text{if } j \in D_x \\ x_j - \epsilon & \text{if } j \in U_x \\ x_j & \text{otherwise.} \end{cases}$$

Cont'd

Note that by our choice of ϵ, $y_j, z_j \in [0,1]$. Moreover, it must be the case that the vectors $\vec{y}$ and $\vec{z}$ are solutions to the LPVC.

Why?

Proof, cont'd

For each edge $e_i = \{v_k, v_j\}$, we simply need to show that $y_k + y_l \geq 1$ and $z_k + z_l \geq 1$. Assume, wlog, that $x_k < x_j$ and assume that neither is equal to 1. (If one is equal to 1, then both $y_k + y_l \geq 1$ and $z_k + z_l \geq 1$. Similarly, if both are equal to 1/2, we are done.) Then, there are three possible values of x_k, x_l.

1. $0 < x_k < 1/2$, $1/2 < x_l < 1$.

2. $1/2 = x_k$, $1/2 < x_l < 1$.

3. $1/2 < x_k < 1$, $1/2 < x_l < 1$.

Cont'd

In case (1), $y_k = x_k - \epsilon$, $y_l = x_l + \epsilon$. Hence, $y_k + y_l = x_k + x_l \geq 1$. Similarly, $z_k = x_k + \epsilon$, $z_l = x_l - \epsilon$. Hence, $z_k + z_l = x_k + x_l \geq 1$.

In case (2), $y_k = x_k$, $y_l = x_l + \epsilon$. Hence, $y_k + y_l = x_k + x_l + \epsilon \geq 1$. For z, $z_k = x_k = 1/2$, and $z_l = x_l - \epsilon \geq 1/2$. Hence, $z_k + z_l \geq 1$.

In case (3), $y_k = x_k + \epsilon$, and $y_l = x_l + \epsilon$. Hence, $y_k + y_l = x_k + x_l + 2\epsilon \geq 1$. For z, $z_k = x_k - \epsilon \geq 1/2$, $z_l = x_l - \epsilon \geq 1/2$. Hence, $z_k + z_l \geq 1$.

Optimal Values

Notice that

$$\sum_{i=1}^{n} y_j = \sum_{i=1}^{n} x_j + \epsilon|U_x| - \epsilon|D_x|$$

and

$$\sum_{i=1}^{n} z_j = \sum_{i=1}^{n} x_j + \epsilon|D_x| - \epsilon|U_x|.$$

Hence, if $|D_x| < |U_x|$ (or vice-versa), then x is not an optimal solution. (This contradicts our assumption.)

Hence, it must be the case that $|D_x| = |U_x|$. Hence, both $\vec{y}$ and $\vec{z}$ are optimal solutions.

Cont'd

Notice that one of $\vec{y}$ or $\vec{z}$ eliminates a value in the range $0 < x_i < 1/2$ or $1/2 < x_i < 1$. Since this process does not modify those values $x_i \in \{0, 1/2, 1\}$

Continuing this process, we will can remove all non-half integral values.

This completes the proof. (Note: the proof really proceeds by induction on $|U_x \cup D_x|$.)

Remark

Claim 1 is due to Nemhauser and Trotter ('74), but was probably known earlier than that. Their original work concerned "Vertex Packings." This should not be confused with Vertex Cover. Vertex Packing is usually referred to as the Independent Set problem.

Why is this result important???

This result is important because it will allow us to “Shrink” the graph.

This will be important in the area of “parameterized” algorithms.