Computational Intractability 2010/5/20

Lecture 6

Professor: David Avis *Scribe: Lin Chang-Hong*

We know how to solve linear programs and prove solutions are correct. How can we use this to solve integer programs? The first approach is to try solving it as an LP. In this lecture we consider a multi-period scheduling problem called the Uncapacitated Lot Size (ULS) problem.

1 ULS problem with no fixed costs

In this problem we have n periods and a demand at each period. The problem is to schedule the production at each of n periods to meet the demand of that period, and possibly leave some extra inventory for the periods to come. There is a given production cost per unit, and a holding cost for the inventory. A flow-chart for the problem with $n = 3$ periods is given in Figure 1.

Symbols:

1. n : periods
2. d_i: demand at period i
3. p_i: cost of producing 1 unit in period i
4. x_i: amount produced in period i
5. s_i : amount of inventory at the end of period i.

We can now formulate the problem in Figure 1 as an LP:

$$\begin{aligned} &min : 3x_1 + 4x_2 + 3x_3 + s_1 + s_2 + s_3 \qquad (1)\\ &x_1 = 6 + s_1\\ &x_2 = 7 - s_1 + s_2\\ &x_3 = 4 - s_2 + s_3 \end{aligned}$$

where, $x_i, s_i \geq 0$ and integer.

When does an LP-relaxation of an ILP have an integer solution? This is a difficult question in general, as it often depends on the objective function. In Figure 2, for some objective functions we get an integer solution, but for other objective functions we will not. One thing we can say is that an LP has an integer solution for every objective function if and only if all vertices of the feasible region are integer.

- **Facts:**

1. An optimum LP solution can always be found at a vertex (Simplex method gives it)
2. If any vertex v is fractional, there is always some objective function optimizing here uniquely.

Proof: Consider any vertex v and write down its dictionary (see Figure 3). Assuming we have a maximization problem, we can replace the current objective row with

$$w = 0 - \sum_{j \in N} x_j \qquad N = \text{cobasis for } v.$$

We can rewrite w by substituting for any slack variable in N using the initial dictionary which defines the slacks. We now have w in terms of original decision variables, and this function uniquely optimizes at vertex v. This completes the proof.

In order to check if all vertices are integer, we can use vertex enumeration to generate all vertices of a polyhedron $Ax \leq b$. There are several programs, one is lrs (others are cdd, porta,...) The input is called an H-representation (halfspace or inequality representation) and the output is called a V-representation (vertices and rays).

The input format for lrs is $[b - A]$, and equations are specified by the 'linearity' command. The input for our LP is:

- **H-representation:**

linearity	3	1	2	3			
vars	b	x_1	x_2	x_3	s_1	s_2	s_3
	6	-1	0	0	1	0	0
	7	0	-1	0	-1	1	0
	4	0	0	-1	0	-1	1
	0	1	0	0	0	0	0
	0	0	1	0	0	0	0
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	0	0	0	0	0	0	1

The linearity command says there are three equations, in rows 1,2, and 3. The remaining inequalities are non-negativity constraints

The output after running lrs will be a V-representation(Vertices and Rays) of the polyhedron (see Figure 5).

The format is:

$1\ x_1\ x_2\ \cdots\ x_{n-1}$ vertex(x_1, x_2, $\cdots$,x_{n-1})

$0\ y_1\ y_2\ \cdots\ y_{n-1}$ ray(y_1, y_2, $\cdots$, y_{n-1})

For our example, the output consists of 4 vertices and 4 rays:

- **V-representation:**

	x_1	x_2	x_3	s_1	s_2	s_3
1	6	7	4	0	0	0
0	0	0	1	0	0	1
1	13	0	4	7	0	0
0	0	0	1	0	0	1
1	6	11	0	0	4	0
0	0	1	0	0	1	1
1	17	0	0	11	4	0
0	1	0	0	1	1	1

We observe that all vertices are integers. Therefore, regardless of the objective function we will always get an integer solutions to the LP. This can be proven formally for this simple version of the ULS problem.

2 ULS problem with fixed costs

Whenever $x_i > 0$, we add fixed cost f_i. If you produce anything at all in period i, you must pay fixed cost f_i.

Let

$$y_i = \begin{cases} 1 & \text{produce in period i} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

We need to link x_i with y_i. Let M be any upper bound on x_i. The linking equation is $x_i \leq My_i$. This means that if x_i is positive, then $y_i = 1$.

Since we are minimizing $x_1 \leq 17$(total demand), $x_2 \leq 11$, $x_3 \leq 4$, the new inequalities are:

$$\begin{aligned} x_1 &\leq 17y_1 \\ x_2 &\leq 11y_2 \\ x_3 &\leq 4y_3 \end{aligned} \tag{3}$$

Suppose $f_1 = 12, f_2 = 10, f_3 = 10$. The objective function now becomes

$$min : 3x_1 + 4x_2 + 3x_3 + s_1 + s_2 + s_3 + 12y_1 + 10y_2 + 12y_3 \tag{4}$$

Note that $y_1 = 1, x_1 = 0$ is feasible but it is not going to be optimum since the fixed costs are positive (ie. $y_1 = 0$ is also feasible at lower cost.) If we now solve the ULS problem as an LP we find that we get an integer optimum solution: Indeed, this problem has many fractional vertices, as you will discover if you run lrs on the corresponding H-representation.

3 LP-relaxations for an Integer Program

In general integer programs have many formulations as linear programs. The property we would like is that the only feasible integer solutions to the LP-formulation are feasible integer solutions to our original problem. These formulations are called LP-relaxations (see Figure 6).

Definition 1. *Let X={feasible integer solutions to the problem} and* $P = \{\boldsymbol{x} : \boldsymbol{Ax} \leq \boldsymbol{b}\}$ *be a polyhedron. P is an LP-relaxation for X if* $X = P \cap Z^n$*, where* Z^n *are the integer vectors or length n.*

If we have an integer optimum solution to an LP-relaxation, then we are sure that it is the optimum integer solution of the original problem. We call the LP-relaxation and ideal formulation whenever all of its vertices are integer. In this case the LP solution is always the integer optimum solution.

4 Another formulation for the ULS problem with fixed costs

We introduce new decision variables. For each pair of periods $i \leq j$ we let w_{ij} denote the production at period i that satisfies demand in period j. In this model we do not need variables for the inventory between periods. The cost of inventory will be included in the coefficient c_{ij} of w_{ij} in the objective. Figure 4 shows the revised flowchart. Our new LP formulation is as follows:

$$min : 3w_{11} + 4w_{12} + 5w_{13} + 4w_{22} + 5w_{23} + 3w_{33} + 12y_1 + 10y_2 + 12y_3$$

$$\begin{array}{lll} w_{11} = d_1, & w_{12} + w_{22} = d_2, & w_{13} + w_{23} + w_{33} = d_3 \\ w_{11} \leq 6y_1, & w_{12} \leq 7y_1, & w_13 \leq 4y_1 \\ w_{22} \leq 7y_2, & w_{23} \leq 4y_2, & w_33 \leq 4y_3 \end{array}$$

This turns out to be a perfect formulation for the ULS problem with fixed costs. The feasible region has only integer vertices, so we can solve it with the simplex method.

Figure 1

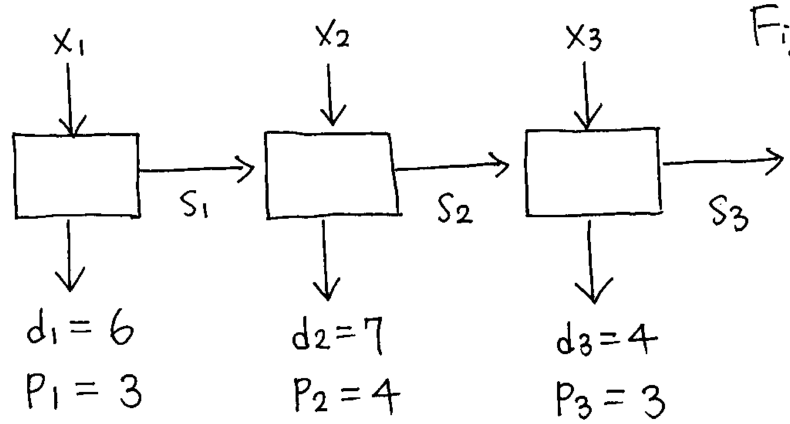


Figure 2

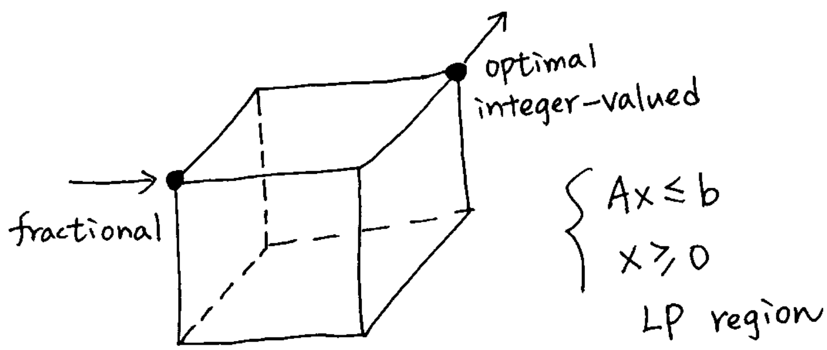


Figure 3

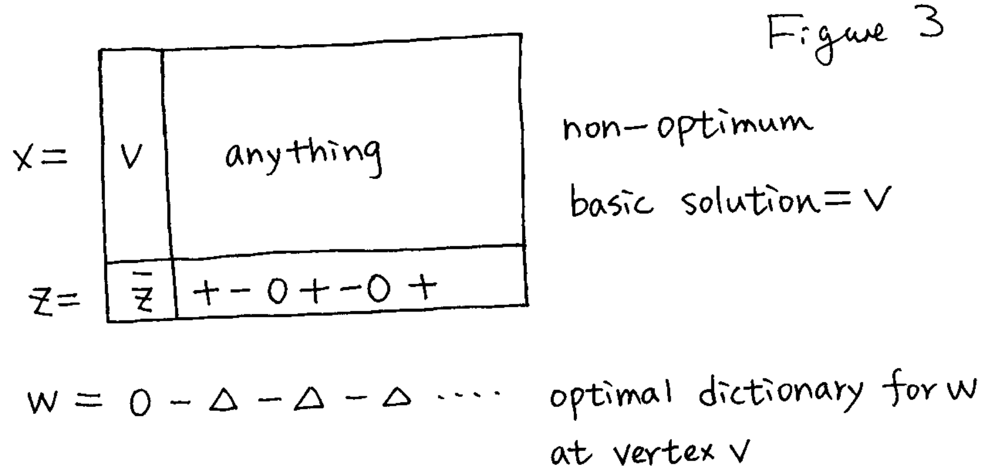
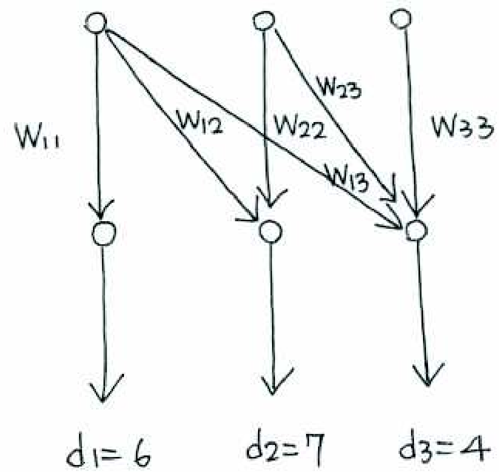


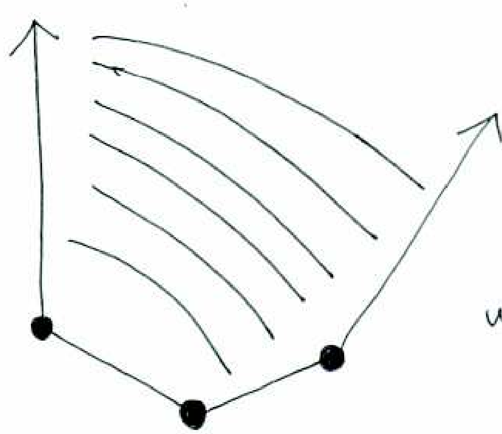
Figure 4



Wij : amount produced in period i and used in period j

include storage cost for j > i

Figure 5



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

unbounded : 3 vertices
2 rays.

Figure 6