HW III Solutions

EE 588: Optimization for the information and data sciences

University of Southern California

Release Date: October 17, 2018

4.15 *Relaxation of Boolean LP.* In a *Boolean linear program*, the variable x is constrained to have components equal to zero or one:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \preceq b \\ & x_i \in \{0,1\}, \quad i = 1, \ldots, n. \end{array} \tag{1}$$

In general, such problems are very difficult to solve, even though the feasible set is finite (containing at most 2^n points).

In a general method called *relaxation*, the constraint that x_i be zero or one is replaced with the linear inequalities $0 \leq x_i \leq 1$:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \preceq b \\ & 0 \leq x_i \leq 1, \quad i = 1, \ldots, n. \end{array} \tag{2}$$

We refer to this problem as the *LP relaxation* of the Boolean LP (1). The LP relaxation is far easier to solve than the original Boolean LP.

1. Show that the optimal value of the LP relaxation (2) is a lower bound on the optimal value of the Boolean LP (1). What can you say about the Boolean LP if the LP relaxation is infeasible?
2. It sometimes happens that the LP relaxation has a solution with $x_i \in \{0, 1\}$. What can you say in this case?

Solution

1. The feasible set of the relaxation includes the feasible set of the Boolean LP. It follows that the Boolean LP is infeasible if the relaxation is infeasible, and that the optimal value of the relaxation is less than or equal to the optimal value of the Boolean LP.
2. The optimal solution of the relaxation is also optimal for the Boolean LP.

■

- *Optimal activity levels.* We consider the selection of n nonnegative activity levels, denoted $x_1, \ldots, x_n$. These activities consume m resources, which are limited. Activity j consumes $A_{ij}x_j$ of resource i, where A_{ij} are given. The total resource consumption is additive, so the total of resource i consumed is $c_i = \sum_{j=1}^n A_{ij}x_j$. (Ordinarily we have $A_{ij} \geq 0$, i.e., activity j consumes resource i. But we allow the possibility that $A_{ij} < 0$, which means that activity j actually *generates* resource i as a by-product.) Each resource consumption is limited: we must have $c_i \leq c_i^{\max}$, where $c_i^{\max}$ are given. Each activity generates revenue, which is a piecewise-linear concave function of the activity level:

$$r_j(x_j) = \begin{cases} p_j x_j & 0 \leq x_j \leq q_j \\ p_j q_j + p_j^{\text{disc}}(x_j - q_j) & x_j \geq q_j. \end{cases}$$

  Here $p_j > 0$ is the basic price, $q_j > 0$ is the quantity discount level, and p_j^{disc} is the quantity discount price, for (the product of) activity j. (We have $0 < p_j^{\text{disc}} < p_j$.) The total revenue is the sum of the revenues associated with each activity, i.e., $\sum_{j=1}^n r_j(x_j)$. The goal is to choose activity levels that maximize the total revenue while respecting the resource limits. Show how to formulate this problem as an LP.

  Solution The basic problem can be expressed as

$$\begin{array}{ll} \text{maximize} & \sum_{j=1}^n r_j(x_j) \\ \text{subject to} & x \succeq 0 \\ & Ax \preceq c^{\max}. \end{array}$$

  This is a convex optimization problem since the objective is concave and the constraints are a set of linear inequalities. To transform it to an equivalent LP, we first express the revenue functions as

$$r_j(x_j) = \min\{p_j x_j, p_j q_j + p_j^{\text{disc}}(x_j - q_j)\},$$

  which holds since r_j is concave. It follows that $r_j(x_j) \geq u_j$ if and only if

$$p_j x_j \geq u_j, \qquad p_j q_j + p_j^{\text{disc}}(x_j - q_j) \geq u_j.$$

  We can form an LP as

$$\begin{array}{ll} \text{maximize} & \mathbf{1}^T u \\ \text{subject to} & x \succeq 0 \\ & Ax \preceq c^{\max} \\ & p_j x_j \geq u_j, \quad p_j q_j + p_j^{\text{disc}}(x_j - q_j) \geq u_j, \quad j = 1, \ldots, n, \end{array}$$

  with variables x and u.

  To show that this LP is equivalent to the original problem, let us fix x. The last set of constraints in the LP ensure that $u_i \leq r_i(x)$, so we conclude that for every feasible x, u in the LP, the LP objective is less than or equal to the total revenue. On the other hand, we can always take $u_i = r_i(x)$, in which case the two objectives are equal. ■

5.5 *Dual of general LP.* Find the dual function of the LP

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Gx \preceq h \\ & Ax = b. \end{array}$$

Give the dual problem, and make the implicit equality constraints explicit.

Solution

1. The Lagrangian is

$$\begin{aligned} L(x, \lambda, \nu) &= c^T x + \lambda^T (Gx - h) + \nu^T (Ax - b) \\ &= (c^T + \lambda^T G + \nu^T A)x - h\lambda^T - \nu^T b, \end{aligned}$$

which is an affine function of x. It follows that the dual function is given by

$$g(\lambda, \nu) = \inf_x L(x, \lambda, \nu) = \begin{cases} -\lambda^T h - \nu^T b & c + G^T \lambda + A^T \nu = 0 \\ -\infty & \text{otherwise.} \end{cases}$$

2. The dual problem is

$$\begin{array}{ll} \text{maximize} & g(\lambda, \nu) \\ \text{subject to} & \lambda \succeq 0. \end{array}$$

After making the implicit constraints explicit, we obtain

$$\begin{array}{ll} \text{maximize} & -\lambda^T h - \nu^T b \\ \text{subject to} & c + G^T \lambda + A^T \nu = 0 \\ & \lambda \succeq 0. \end{array}$$

■

5.13 *Lagrangian relaxation of Boolean LP.* A *Boolean linear program* is an optimization problem of the form

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \preceq b \\ & x_i \in \{0, 1\}, \quad i = 1, \ldots, n, \end{array}$$

and is, in general, very difficult to solve. In exercise we studied the LP relaxation of this problem,

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \preceq b \\ & 0 \le x_i \le 1, \quad i = 1, \ldots, n, \end{array} \tag{3}$$

which is far easier to solve, and gives a lower bound on the optimal value of the Boolean LP. In this problem we derive another lower bound for the Boolean LP, and work out the relation between the two lower bounds.

1. *Lagrangian relaxation.* The Boolean LP can be reformulated as the problem

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \preceq b \\ & x_i(1-x_i) = 0, \quad i = 1, \dots, n, \end{array}$$

which has quadratic equality constraints. Find the Lagrange dual of this problem. The optimal value of the dual problem (which is convex) gives a lower bound on the optimal value of the Boolean LP. This method of finding a lower bound on the optimal value is called *Lagrangian relaxation.*

2. Show that the lower bound obtained via Lagrangian relaxation, and via the LP relaxation (3), are the same. *Hint.* Derive the dual of the LP relaxation (3).

Solution

1. The Lagrangian is

$$\begin{aligned} L(x,\mu,\nu) &= c^T x + \mu^T(Ax-b) - \nu^T x + x^T \mathbf{diag}(\nu) x \quad (4) \\ &= x^T \mathbf{diag}(\nu) x + (c + A^T\mu - \nu)^T x - b^T \mu. \quad (5) \end{aligned}$$

Minimizing over x gives the dual function

$$g(\mu,\nu) = \begin{cases} -b^T\mu - (1/4)\sum_{i=1}^n (c_i + a_i^T\mu - \nu_i)^2/\nu_i & \nu \succeq 0 \\ -\infty & \text{otherwise} \end{cases}$$

where a_i is the ith column of A, and we adopt the convention that $a^2/0 = \infty$ if $a \neq 0$, and $a^2/0 = 0$ if $a = 0$.

The resulting dual problem is

$$\begin{array}{ll} \text{maximize} & -b^T\mu - (1/4)\sum_{i=1}^n (c_i + a_i^T\mu - \nu_i)^2/\nu_i \\ \text{subject to} & \mu \succeq 0, \end{array}$$

with implicit constraint $\nu \succeq 0$.

In order to simplify this dual, we optimize analytically over ν, by noting that

$$\begin{aligned} \sup_{\nu_i \geq 0} \left(-\frac{(c_i + a_i^T\mu - \nu_i)^2}{\nu_i} \right) &= \begin{cases} 4(c_i + a_i^T\mu) & c_i + a_i^T\mu \leq 0 \\ 0 & c_i + a_i^T\mu \geq 0 \end{cases} \quad (6) \\ &= \min\{0, 4(c_i + a_i^T\mu)\}. \quad (7) \end{aligned}$$

This allows us to eliminate ν from the dual problem, and simplify it as

$$\begin{array}{ll} \text{maximize} & -b^T\mu + \sum_{i=1}^n \min\{0, c_i + a_i^T\mu\} \\ \text{subject to} & \mu \succeq 0. \end{array}$$

2. We follow the hint. The Lagrangian and dual function of the LP relaxation are

$$\begin{aligned} L(x,u,v,w) &= c^T x + u^T(Ax-b) - v^T x + w^T(x - \mathbf{1}) \quad (8) \\ &= (c + A^T u - v + w)^T x - b^T u - \mathbf{1}^T w \quad (9) \\ g(u,v,w) &= \begin{cases} -b^T u - \mathbf{1}^T w & A^T u - v + w + c = 0 \\ -\infty & \text{otherwise.} \end{cases} \quad (10) \end{aligned}$$

The dual problem is

$$\begin{array}{ll} \text{maximize} & -b^T u - \mathbf{1}^T w \\ \text{subject to} & A^T u - v + w + c = 0 \\ & u \succeq 0, v \succeq 0, w \succeq 0, \end{array}$$

which is equivalent to the Lagrange relaxation problem derived above. We conclude that the two relaxations give the same value.

■

- *Reformulating constraints in CVX.* Each of the following CVX code fragments describes a convex constraint on the scalar variables `x`, `y`, and `z`, but violates the CVX rule set, and so is invalid. Briefly explain why each fragment is invalid. Then, rewrite each one in an equivalent form that conforms to the CVX rule set. In your reformulations, you can use linear equality and inequality constraints, and inequalities constructed using CVX functions. You can also introduce additional variables, or use LMIs. Be sure to explain (briefly) why your reformulation is equivalent to the original constraint, if it is not obvious.

  Check your reformulations by creating a small problem that includes these constraints, and solving it using CVX. Your test problem doesn't have to be feasible; it's enough to verify that CVX processes your constraints without error.

  Remark. This *looks* like a problem about 'how to use CVX software', or 'tricks for using CVX'. But it really checks whether you understand the various composition rules, convex analysis, and constraint reformulation rules.

  1. `norm([x + 2*y, x - y]) == 0`
  2. `square(square(x + y)) <= x - y`
  3. `1/x + 1/y <= 1; x >= 0; y >= 0`
  4. `norm([max(x,1), max(y,2)]) <= 3*x + y`
  5. `x*y >= 1; x >= 0; y >= 0`
  6. `(x + y)^2/sqrt(y) <= x - y + 5`
  7. `x^3 + y^3 <= 1; x >= 0; y >= 0`
  8. `x + z <= 1 + sqrt(x*y - z^2); x >= 0; y >= 0`

  Solution

  1. The lefthand side is correctly identified as convex, but equality constraints are only valid with affine left and right hand sides. Since the norm of a vector is zero if and only if the vector is zero, we can express the constraint as `x + 2*y == 0; x - y == 0`, or simply `x == 0; y == 0`.
  2. The problem is that `square()` can only accept affine arguments, because it is convex, but not increasing. To correct this use `square_pos()` instead:

     ```
     square_pos(square(x + y)) <= x - y
     ```

     We can also reformulate this constraint by introducing an additional variable.

```
variable t
square(x + y) <= t
square(t) <= x - y
```

Note that, in general, decomposing the objective by introducing new variables doesn't need to work. It works in this case because the outer `square` function is convex and monotonic over $\mathbb{R}_+$.

Alternatively, we can rewrite the constraint as

```
(x + y)^4 <= x - y
```

3. $1/x$ isn't convex, unless you restrict the domain to $\mathbb{R}_{++}$. We can write this one as `inv_pos(x) + inv_pos(y) <= 1`. The `inv_pos` function has domain $\mathbb{R}_{++}$ so the constraints $x > 0$, $y > 0$ are (implicitly) included.
4. The problem is that `norm()` can only accept affine argument since it is convex but not increasing. One way to correct this is to introduce new variables `u` and `v`:

   ```
   norm([u, v]) <= 3*x + y
   max(x,1) <= u
   max(y,2) <= v
   ```

   Decomposing the objective by introducing new variables works here because `norm` is convex and monotonic over $\mathbb{R}^2_+$, and in particular over $[1, \infty) \times [2, \infty)$.
5. xy isn't concave, so this isn't going to work as stated. But we can express the constraint as `x >= inv_pos(y)`. (You can switch around `x` and `y` here.) Another solution is to write the constraint as `geo_mean([x, y]) >= 1`. We can also give an LMI representation:

   ```
   [x 1; 1 y] == semidefinite(2)
   ```
6. This fails when we attempt to divide a convex function by a concave one. We can write this as

   ```
   quad_over_lin(x + y, sqrt(y)) <= x - y + 5
   ```

   This works because `quad_over_lin` is monotone decreasing in the second argument, so it can accept a concave function here, and `sqrt` is concave.
7. The function $x^3 + y^3$ is convex for $x \geq 0$, $y \geq 0$. But x^3 isn't convex for $x < 0$, so CVX is going to reject this statement. One way to rewrite this constraint is

   ```
   quad_pos_over_lin(square(x),x) + quad_pos_over_lin(square(y),y) <= 1
   ```

   This works because `quad_pos_over_lin` is convex and increasing in its first argument, hence accepts a convex function in its first argument. (The function `quad_over_lin`, however, is not increasing in its first argument, and so won't work.)

   Alternatively, and more simply, we can rewrite the constraint as

   ```
   pow_pos(x,3) + pow_pos(y,3) <= 1
   ```
8. The problem here is that xy isn't concave, which causes CVX to reject the statement. To correct this, notice that

   $$\sqrt{xy - z^2} = \sqrt{y(x - z^2/y)},$$

   so we can reformulate the constraint as

```
x + z <= 1 + geo_mean([x - quad_over_lin(z,y), y])
```

This works, since `geo_mean` is concave and nondecreasing in each argument. It therefore accepts a concave function in its first argument.

We can check our reformulations by writing the following feasibility problem in CVX (which is obviously infeasible)

```
cvx_begin
    variables x y u v z
    x == 0;
    y == 0;
    (x + y)^4 <= x - y;
    inv_pos(x) + inv_pos(y)  <= 1;
    norm([u; v]) <= 3*x + y;
    max(x,1) <= u;
    max(y,2) <= v;
    x >= inv_pos(y);
    x >= 0;
    y >= 0;
    quad_over_lin(x + y, sqrt(y)) <= x - y + 5;
    pow_pos(x,3) + pow_pos(y,3) <= 1;
    x+z <= 1+geo_mean([x-quad_over_lin(z,y), y])
cvx_end
```

■

- *Optimal activity levels.* Solve the optimal activity level problem described in exercise 4.17 in *Convex Optimization*, for the instance with problem data

$$A = \begin{bmatrix} 1 & 2 & 0 & 1 \\ 0 & 0 & 3 & 1 \\ 0 & 3 & 1 & 1 \\ 2 & 1 & 2 & 5 \\ 1 & 0 & 3 & 2 \end{bmatrix}, \qquad c^{\max} = \begin{bmatrix} 100 \\ 100 \\ 100 \\ 100 \\ 100 \end{bmatrix}, \qquad p = \begin{bmatrix} 3 \\ 2 \\ 7 \\ 6 \end{bmatrix}, \qquad p^{\mathrm{disc}} = \begin{bmatrix} 2 \\ 1 \\ 4 \\ 2 \end{bmatrix}, \qquad q = \begin{bmatrix} 4 \\ 10 \\ 5 \\ 10 \end{bmatrix}.$$

You can do this by forming the LP you found in your solution of exercise 4.17, or more directly, using CVX. Give the optimal activity levels, the revenue generated by each one, and the total revenue generated by the optimal solution. Also, give the average price per unit for each activity level, i.e. the ratio of the revenue associated with an activity, to the activity level. (These numbers should be between the basic and discounted prices for each activity.) Give a *very brief* story explaining, or at least commenting on, the solution you find.

Solution The following Matlab/CVX code solves the problem. (Here we write the problem in a form close to its original statement, and let CVX do the work of reformulating it as an LP.)

```
A=[ 1 2 0 1;
    0 0 3 1;
    0 3 1 1;
    2 1 2 5;
    1 0 3 2];

cmax=[100;100;100;100;100];
p=[3;2;7;6];
pdisc=[2;1;4;2];
q=[4; 10 ;5; 10];

cvx_begin
  variable x(4)
  maximize( sum(min(p.*x,p.*q+pdisc.*(x-q))) )
  subject to
    x >= 0;
    A*x <= cmax
cvx_end

x
r=min(p.*x,p.*q+pdisc.*(x-q))
totr=sum(r)
avgPrice=r./x
```

The result of the code is

```
x =

    4.0000
   22.5000
   31.0000
    1.5000


r =

   12.0000
   32.5000
  139.0000
    9.0000


totr =

  192.5000
```

```
avgPrice =

    3.0000
    1.4444
    4.4839
    6.0000
```

We notice that the 3rd activity level is the highest and is also the one with the highest basic price. Since it also has a high discounted price its activity level is higher than the discount quantity level and it produces the highest contribution to the total revenue. The 4th activity has a discounted price which is substantially lower then the basic price and its activity is therefore lower than the discount quantity level. Moreover it requires the use of a lot of resources and therefore its activity level is low. ■