6.255 Optimization Methods

Problem Set 4

Due: November 18, 2021

1 Due exercises

Problem 1: Lagrangean dual – based on Bertsimas & Tsitsiklis, Exercise 11.1. (10 points) Consider the integer programming (IP) problem:

$$\begin{array}{ll} \text{maximize} & 3x_1 + 2x_2 \\ \text{subject to} & 4x_1 + 2x_2 \leq 17 \\ & -2x_1 + 4x_2 \leq 9 \\ & x_1, x_2 \geq 0 \\ & x_1, x_2 \text{ integer.} \end{array}$$

Using figures, justify the answers to the following questions. Highlight or write in a different color the answer for each question.

1. What is the optimal cost of the linear programming relaxation? What is the optimal cost of the integer programming problem?
2. Draw the convex hull of the set of all solutions to the integer programming problem. How does the solution to the IP compare to the solution when the feasibility region is replaced by this convex hull?
3. Give the first Gomory cut for this problem. (Write the problem in standard form, specify the constraints where you are performing the cut and the resulting constraint after the cut.)
4. Suppose you dualize the constraint $4x_1 + 2x_2 \leq 17$ (this means X in your Lagrangian dual is composed of all the other constraints except $4x_1 + 2x_2 \leq 17$). What is the optimal value Z_D of your Lagrangian dual?
5. Suppose you dualize the constraint $-2x_1 + 4x_2 \leq 9$ (this means X in your Lagrangian dual is composed of all the other constraints except $-2x_1 + 4x_2 \leq 9$). What is the optimal value Z_D of your Lagrangian dual?

Solution:

1. From the graphical representation (Figure 1), the optimal solution of the LP relaxation is $(2.5, 3.5)$ with value 14.5. In addition, the optimal solution of the IP is $(3, 2)$ with value 13.
2. The convex hull of the IP solutions is given below in Figure 2. When we replace the feasibility region by this convex hull, we get the same optimal solution as the solution of the IP (taking the convex hull does not change the optimal solution nor the optimal cost).
3. The standard form of the problem is

$$\begin{array}{ll} \min & -3x_1 - 2x_2 \\ \text{subject to} & 4x_1 + 2x_2 + s_1 = 17 \\ & -2x_1 + 4x_2 + s_2 = 9 \\ & x_1, x_2, s_1, s_2 \geq 0 \end{array}$$

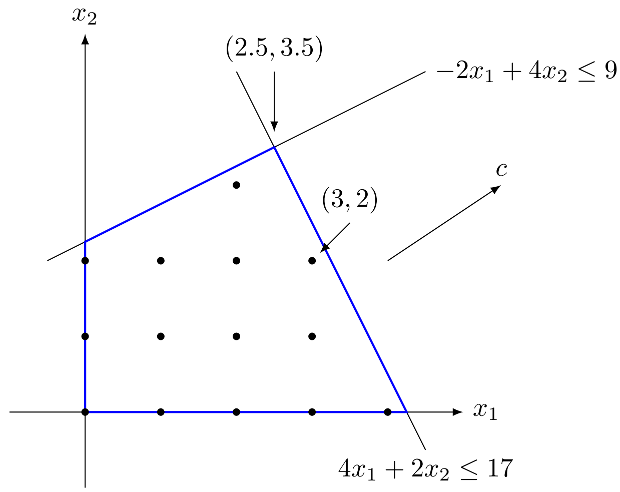

Figure 1

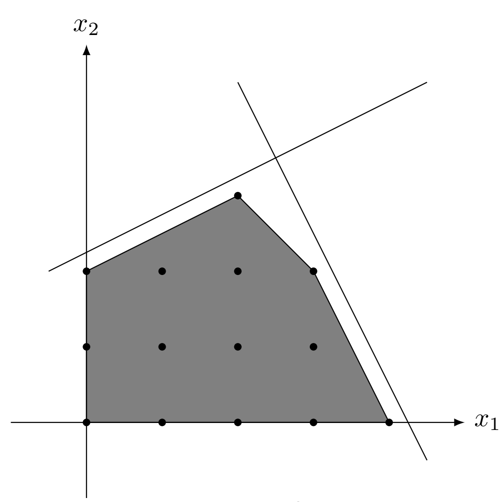

Figure 2

and the corresponding solution is $(2.5, 3.5, 0, 0)$. An optimal basis is $\{x_1, x_2\}$. With B the corresponding basis matrix, the constraints are $B^{-1}Ax = B^{-1}b$:

$$\begin{cases} x_1 + \frac{1}{5}s_1 - \frac{1}{10}s_2 = \frac{5}{2} \\ x_2 + \frac{1}{10}s_2 + \frac{1}{5}s_2 = \frac{7}{2}. \end{cases}$$

The corresponding Gomory cuts are $\begin{cases} x_1 - s_2 \le 2 \\ x_2 \le 3 \end{cases}$ i.e. $\begin{cases} -x_1 + 4x_2 \le 11 \\ x_2 \le 3 \end{cases}$ Graphically, depending on which cut is used we have one of the two pictures: with the first cut, the optimal solution is $(23/9, 61/18)$ with value $130/9 \approx 14.444$ (Figure 3). With the second cut, the optimal solution is $(2.75, 3)$ with value 14.25 (Figure 4.

4. By Theorem 11.4, we can solve the problem graphically represented in the next Figure 5. The region delimited by the red line is $conv(X)$ and the gray region is the feasible region. The optimal solution of the Lagrangian dual is $(2.6, 3.3)$ with value 14.4.
5. Similarly, the region delimited by the red line is $conv(X)$ and the gray region is the feasible region (Figure 6). The optimal solution of the Lagrangian dual is $(2.3, 3.4)$ with value 13.7.

Problem 2: Dynamic programming. (15 points) Consider a set S of n non-negative integers $\{s_1, s_2, \dots, s_n\}$. We want to find, if it exists, a partition of the set S into three subsets A, B, C, in such a way that the sum of the numbers in each partition is the same.

For instance, if $n = 8$ and $S = \{1, 1, 1, 2, 2, 4, 5, 5\}$, then a possible partition could be $A = \{1, 2, 4\}$, $B = \{2, 5\}$, and $C = \{1, 1, 5\}$, each of which adds up to 7.

1. What necessary condition must the data $\{s_1, \dots, s_n\}$ satisfy for the problem to have a feasible solution?
2. Give a dynamic programming algorithm to solve the partition problem. Explain how to compute the partition (if it exists) from your solution.
3. Assume that we modify the problem to incorporate an objective function, where we want to find a solution that maximizes the cardinality of the smallest subset. Give an integer programming formulation for this problem.

Solution:

1. Necessary condition: $\sum_{i=1}^n s_i$ has to be a multiple of 3, i.e. $\sum_{i=1}^n s_i = 3p$ for some $p \ge 0$.
2. Note that this is a feasibility problem. For a DP solution, we wish to conpute the tables $W_m(a, b)$ with 0/1 entries, that will indicate whether there exists a partition of the first m items, in such a way that $\sum_{i\in A} s_i = a$ and $\sum_{i\in A} s_i = b$ (value 1 if it is possible, 0 if not possible). We do not need to keep track of C since the total sum is constant. By considering where to put the $(m + 1)$-th item, we can write the following recursion:

   $$W_{m+1}(a, b) = \max(W_m(a - s_{m+1}, b), W_m(a, b - s_{m+1}), W_m(a, b)).$$

   This recursion says that the given tuple (a, b) is feasible with $m + 1$ items if there exists at least one partition with one item fewer. The initial conditions are simple: $W_1(s_1, 0) = W_1(0, s_1) = W_1(0, 0) = 1$ and all other entries of W_1 are 0. We also have $W_i(a, b) = 0$ if $a < 0$ or $b < 0$ or $a + b > \sum_{j\le i} s_j$. To solve the given instance, we check the entry $W_n(s, s)$ where $s = \frac{1}{3}\sum_{i=1}^n s_i$. The corresponding optimal solution can be computed from these tables by backtracking.
3. We introduce variables a_i, b_i, c_i for $1 \le i \le n$, representing the elements in sets A, B and C: $a_i = 1$ (resp. $b_i = 1$, $c_i = 1$) i.if. s_i is in set A (resp. B, C). Denote $s = \frac{1}{3}\sum_{i=1}^n s_i$. The problem can be

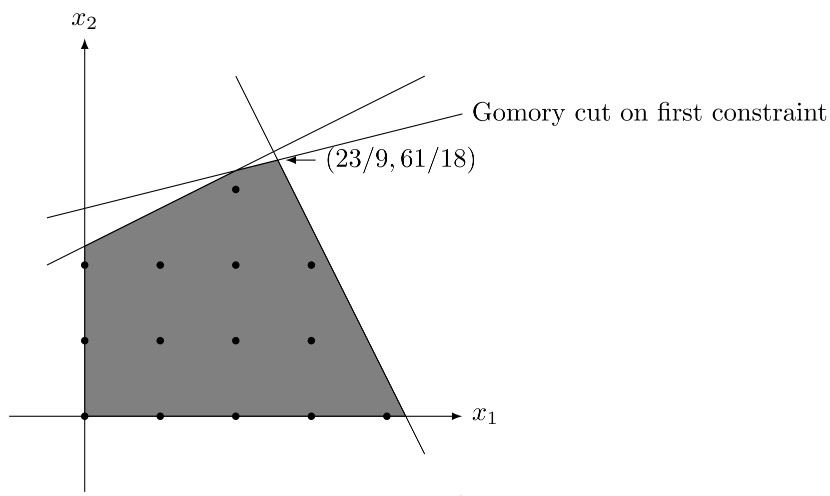

Figure 3

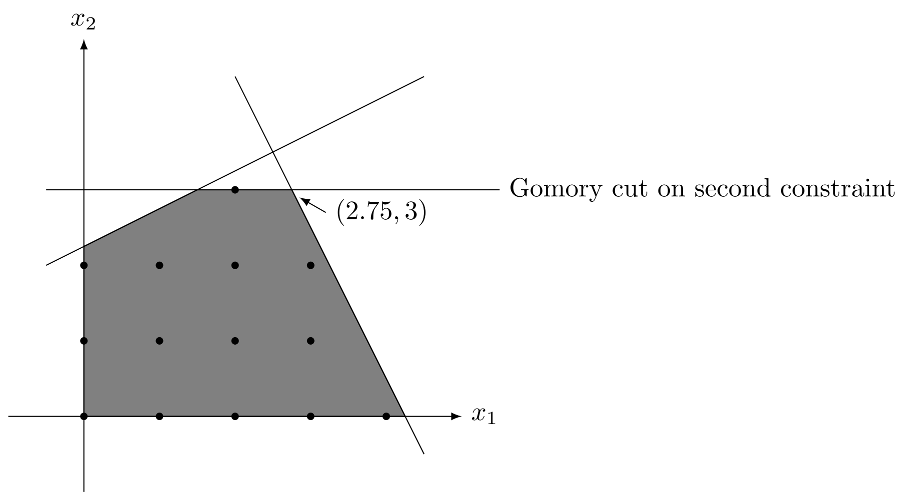

Figure 4

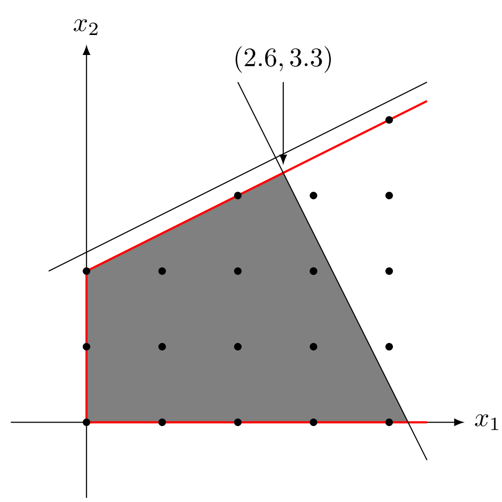

Figure 5

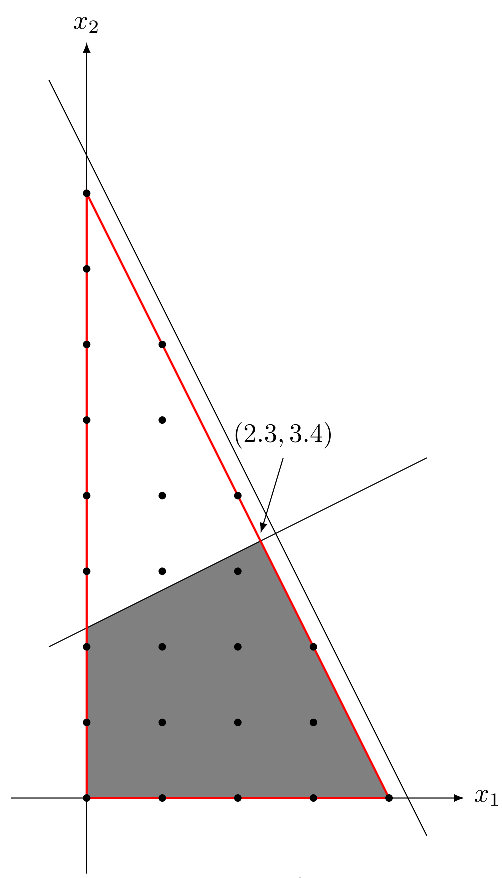

Figure 6

formulated as

$$\begin{array}{lll} \max & z & \\ \text{subject to} & a_i + b_i + c_i = 1 & i = 1, \dots n \\ & \sum_{i=1}^n a_i s_i = s & \\ & \sum_{i=1}^n b_i s_i = s & \\ & \sum_{i=1}^n c_i s_i = s & \\ & z \leq \sum_{i=1}^n a_i & \\ & z \leq \sum_{i=1}^n b_i & \\ & z \leq \sum_{i=1}^n c_i & \\ & a_i, b_i, c_i \in \{0, 1\} & \end{array}$$

Problem 3: Comparisons of relaxations for an assignment problem with a side constraint – Bertsimas & Tsitsiklis, Exercise 11.12. (15 points) We would like to assign n machines to n jobs in order to minimize the total cost of the assignment (it costs c_{ij} to assign machine i to job j. In addition, there is a value d_{ij} if machine i is assigned to job j. We would like the total value from the assignment to be above a threshold b. We formulate the problem as follows:

$$\begin{array}{lll} \text{minimize} & \displaystyle\sum_{1 \leq i,j \leq n} c_{ij} x_{ij} & \\ \text{subject to} & \displaystyle\sum_{i=1}^n x_{ij} = 1, & 1 \leq j \leq n, \\ & \displaystyle\sum_{j=1}^n x_{ij} = 1, & 1 \leq i \leq n, \\ & \displaystyle\sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ij} \geq b, & \\ & x_{ij} \in \{0, 1\} & 1 \leq i, j \leq n. \end{array}$$

We consider the following alternative relaxations

1. Relax the third constraint.
2. Relax the first and second sets of constraints.
3. Relax the first set of constraint.
4. Relax the first set of constraints and the third constraint.

Let Z_{Di} be the value of the corresponding Lagrangean dual problem, $i = 1, \dots, 4$. Let Z_{LP} be the optimal cost of the linear programming relaxation. Prove that these relaxations are ordered as follows:

$$Z_{LP} = Z_{D1} = Z_{D4} \leq Z_{D2} \leq Z_{D3} \leq Z_{IP}.$$

Solution: If we relax the third constraint, the remaining constraints form a polyhedron with integer extreme points: the remaining constraints form the assignment polyhedron $P = \{x, \sum_i x_{ij} = 1, \forall j, \sum_j x_{ij} = 1, \forall i, 0 \leq x_{ij} \leq 1, \forall i, j\}$. (The constraint $x_{ij} \in \{0, 1\}$ first has to be reformulated as $0 \leq x_{ij} \leq 1$ and x_{ij} integer). Similarly, if we relax the first and third set of constraints, the remaining constraints form an integral polyhedron $P' = \{x, \sum_j x_{ij} = 1, \forall i, 0 \leq x_{ij} \leq 1, \forall i, j\}$. To see this, you can e.g. see that it corresponds to the network flow of the assignment problem: remove the outflow 1 of all right nodes and link all these nodes to a sink node with infinite capacity. Last, the outflow at sink node is n. Hence, $Z_{LP} = Z_{D1} = Z_{D4}$.

Using the Lagrangian theorem, we have

$$Z_{D2} = \begin{array}{ll} \min & \displaystyle\sum_{1 \le i,j \le n} c_{ij} x_{ij} \\ \text{s.t.} & \displaystyle\sum_{i=1}^{n} x_{ij} = 1, \quad 1 \le j \le n, \\ & \displaystyle\sum_{j=1}^{n} x_{ij} = 1, \quad 1 \le i \le n, \\ & x \in CH \left(x \text{ s.t.} \begin{cases} \displaystyle\sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij} \ge b, \\ x_{ij} \in \{0,1\}, \quad 1 \le i,j \le n. \end{cases} \right) \end{array} \tag{1}$$

and

$$Z_{D3} = \begin{array}{ll} \min & \displaystyle\sum_{1 \le i,j \le n} c_{ij} x_{ij} \\ \text{s.t.} & \displaystyle\sum_{i=1}^{n} x_{ij} = 1, \quad 1 \le j \le n, \\ & x \in CH \left(x \text{ s.t.} \begin{cases} \displaystyle\sum_{j=1}^{n} x_{ij} = 1, \quad 1 \le i \le n, \\ \displaystyle\sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij} \ge b, \\ x_{ij} \in \{0,1\}, \quad 1 \le i,j \le n. \end{cases} \right) \end{array} \tag{2}$$

To show that $Z_{D2} \le Z_{D3}$ we will show that the feasibility set of problem (2) is included in the feasibility set of problem (1). Let x^* feasible for problem (2). By hypothesis, it satisfies constraint 1: $\sum_{i=1}^{n} x^*_{ij} = 1, 1 \le j \le n$. Further, we have by increasing property of convex hulls

$$CH \left(x \text{ s.t.} \begin{cases} \displaystyle\sum_{j=1}^{n} x_{ij} = 1, \quad 1 \le i \le n, \\ \displaystyle\sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij} \ge b, \\ x_{ij} \in \{0,1\}, \quad 1 \le i,j \le n. \end{cases} \right) \subset CH \left(x \text{ s.t. } \sum_{j=1}^{n} x_{ij} = 1, \quad 1 \le i \le n \right)$$

where the right hand term is simply $\{x \text{ s.t. } \sum_{j=1}^{n} x_{ij} = 1, \quad 1 \le i \le n\}$. This shows that the convex hull of a set which satisfy a linear constraint, still satisfy this constraint. In particular x^* satisfies the constraint $\sum_{j=1}^{n} x^*_{ij} = 1, \quad 1 \le i \le n$. Last,

$$CH \left(x \text{ s.t.} \begin{cases} \displaystyle\sum_{j=1}^{n} x_{ij} = 1, \quad 1 \le i \le n, \\ \displaystyle\sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij} \ge b, \\ x_{ij} \in \{0,1\}, \quad 1 \le i,j \le n. \end{cases} \right) \subset CH \left(x \text{ s.t.} \begin{cases} \displaystyle\sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij} \ge b, \\ x_{ij} \in \{0,1\}, \quad 1 \le i,j \le n. \end{cases} \right)$$

which ends the proof that x^* is feasible for problem (1). Therefore $Z_{D2} \le Z_{D3}$. Also $Z_{D2} \le Z_{LP}$, and all relaxations have cost which is smaller than or equal to Z_{IP}.

Note. This is true in general: relaxing additional constraints only worsens the quality of your Lagrangian relaxation. E.g. if no constraints are relaxed we have Z_{IP} and if all of them are relaxed we have Z_{LP}. The same proof as above proves this claim. Consider

$$Z_1 = \begin{array}{ll} \min & c^\top x \\ \text{s.t.} & Ax \geq a, \\ & Bx \geq b \\ & x \in CH\,(x \text{ integer s.t. } Cx \geq c) \end{array} \tag{3}$$

and

$$Z_2 = \begin{array}{ll} \min & c^\top x \\ \text{s.t.} & Ax \geq a, \\ & x \in CH\,(x \text{ integer s.t. } Bx \geq b, Cx \geq c) \end{array} \tag{4}$$

We can prove that $Z_1 \leq Z_2$ by showing that the feasibility set of (4) is included in that of (3). Take x^* in the feasible set of (4). Then $Ax^* \geq a$. Also, because $CH\,(x \text{ integer s.t. } Bx \geq b, Cx \geq c) \subset CH(x \text{ s.t. } Bx \geq b) = \{x, \text{ s.t. } Bx \geq b\}$ we have $Bx^* \geq b$. Last, we have $CH\,(x \text{ integer s.t. } Bx \geq b, Cx \geq c) \subset CH\,(x\text{integer s.t. } Cx \geq c)$. Therefore x^* is feasible for (3).

Problem 4: An approximation algorithm for maximum satisfiability – Bertsimas & Tsitsiklis, Exercise 11.16. (15 points) This exercise shows the use of randomization in constructing approximation algorithms for the following problem in logic called the maximum satisfiability problem (MAXSAT).

Given a collection $\mathcal{C} = \{C_1, \dots, C_m\}$ of boolean clauses, where each clause is a disjunction of literals (a literal is either a boolean variable x or its negation $\bar{x}$ from a set of variables $\{x_1, \dots, x_n\}$, and positive weights w_i for each clause C_i, the goal in MAXSAT is to assign truth values to the variables $x_1, \dots, x_n$ in order to maximize the sum of weights of the satisfied clauses.

Let us formulate the problem. Let $y_i = 1$ if we set x_i to be true, and $y_i = 0$ otherwise. Let $z_j = 1$ if clause C_j is satisfied. Let I_j^+, (resp. I_j^-) be the set of literals that are not (resp. are) negated in clause C_j. Then, MAXSAT can be formulated as follows:

$$\begin{array}{ll} \text{maximize} & \displaystyle\sum_{j=1}^{m} w_j z_j \\ \text{subject to} & \displaystyle\sum_{i \in I_j^+} y_i + \sum_{i \in I_j^-} (1 - y_i) \geq z_j, \quad C_j \in \mathcal{C}, \\ & y_i, z_j \in \{0, 1\}. \end{array}$$

We denote the optimal cost by Z_{IP} and the optimal cost of the linear programming relaxation by Z_{LP}. Consider the following heuristic:

(a) Solve the linear programming relaxation and find optimal values y_i^*, z_j^*.
(b) Interpret the numbers y_i^* as probabilities. Set $\tilde{y}_i$ to 0 or 1, randomly and independently, with probability

$$\mathbb{P}(\tilde{y}_i = 1) = y_i^*.$$

(c) Set the values $\tilde{z}_j$ to be 0 or 1, with preference given to 1 when possible, so that the resulting solution is feasible i.e. $\tilde{z}_j = 1$ if $\sum_{i \in I_j^+} \tilde{y}_i + \sum_{i \in I_j^-} (1 - \tilde{y}_i) \geq 1$ and $\tilde{z}_j = 0$ if $\sum_{i \in I_j^+} \tilde{y}_i + \sum_{i \in I_j^-} (1 - \tilde{y}_i) < 1$.

The resulting solution is always feasible, but its value is a random variable. Let Z_H be the value of the solution produced by the heuristic and let $\mathbb{E}[Z_H]$ be its expected value.

1. For a given clause C_j, show that

$$\log \mathbb{P}[\tilde{z}_j = 0] \leq - \sum_{i \in I_j^+} y_i^* - \sum_{i \in I_j^-} (1 - y_i^*)$$

2. Show that for any clause $C_j \in \mathcal{C}$,

$$\mathbb{E}[\tilde{z}_j] \geq \frac{e-1}{e} z_j^*$$

3. Prove that

$$Z_{LP} \geq Z_{IP} \geq \mathbb{E}[Z_H] \geq \frac{e-1}{e} Z_{LP},$$

where $e = 2.71..$ is the base of the natural logarithm.

Hint: $\log(1-x) \leq -x$ *for all* $0 \leq x \leq 1$. *Also,* $e^x \leq 1 + x(1-1/e)$ *for all* $-1 \leq x \leq 0$. *(Both inequalities are convexity inequalities.)*

Solution:

1. Observe that $\tilde{z}_j = 0$ only if $\tilde{y}_i = 0$ for all $i \in I_j^+$ and $\tilde{y}_i = 1$ for all $i \in I_j^-$. All these events are independent by construction. Therefore,

$$\mathbb{P}[\tilde{z}_j = 0] = \prod_{i \in I_j^+} (1 - y_i^*) \prod_{i \in I_j^-} y_i^*.$$

Thus,

$$\begin{aligned} \log \mathbb{P}[\tilde{z}_j = 0] &= \sum_{i \in I_j^+} \log(1 - y_i^*) + \sum_{i \in I_j^-} \log(y_i^*) \\ &\leq - \sum_{i \in I_j^+} y_i^* - \sum_{i \in I_j^-} (1 - y_i^*), \end{aligned}$$

where in the last inequality we used the fact $\log(1-x) \leq -x$.

2. Using 1. we have

$$\begin{aligned} \mathbb{E}[\tilde{z}_j] = \mathbb{P}[\tilde{z}_j = 1] = 1 - \mathbb{P}[\tilde{z}_j = 0] &\geq 1 - e^{-\sum_{i \in I_j^+} y_i^* - \sum_{i \in I_j^-} (1 - y_i^*)} \\ &\geq (\sum_{i \in I_j^+} y_i^* + \sum_{i \in I_j^-} (1 - y_i^*))(1 - 1/e) \\ &\geq \frac{e-1}{e} z_j^*. \end{aligned}$$

In the last inequality we used the fact that y^*, z^* is a feasible solution to the relaxed problem.

3. Since this is a maximization problem, we have $Z_{LP} \geq Z_{IP}$. Further, the result of the heuristic is a feasible solution to the IP problem, so we always have $Z_{LP} \geq Z_H$. Thus, $Z_{IP} \geq \mathbb{E}[Z_H]$. Last, using 2,

$$\mathbb{E}[Z_H] = \sum_{j=1}^m w_j \mathbb{E}[\tilde{z}_j] \geq \frac{e-1}{e} \sum_{j=1}^m w_j z_j^* = \frac{e-1}{e} Z_{LP}.$$

2 Practice exercises

Problem 5: MIP vs Linear relaxation (Practice) Bertsimas & Tsitsiklis, Exercise 11.2.

Solution: a) For simplicity, let us pretend that the linear programming relaxation is in standard form, although in reality we will have to convert to standard form, apply the following argument, and convert back. If the optimal cost of the linear programming relaxation is $-\infty$, simplex will find a vector d such that $c'd < 0$ and $x + \theta d$ is feasible whenever x is feasible and $\theta \geq 0$. Remember that d consists of 0 entries, one entry 1, and the elements of one of the columns of the final tableau. Since A and b are integer, $B^{-1}A$ is rational (see Cramer's rule, p. 29). This shows that the elements of d are all rational. We denote by p the least common multiple of the denominators of entries in d. Note that pd has integer entries.

Since the IP is feasible, let x_0 be a feasible integer solution. We will now show that for any integer $\theta \geq 0$ integer, $x_0 + \theta pd$ is also feasible for the IP. Indeed, it still satisfies the linear constraints (because d is an unbounded direction of the LP relaxation, hence $x_0 + \theta d$ is feasible for the LP relaxation for all reals $\theta \geq 0$). Further, since θ is integer, $x_0 + \theta \cdot (pd)$ is integer as well. Finally, because $c'd < 0$, $c'(x_0 + \theta pd) \to -\infty$ as $\theta \to \infty$. Therefore the IP is unbounded.

b) The example in Exercise 11.4 has $Z_{IP} = 1$ and $Z_{LP} = 0$:

$$\begin{array}{ll} \min & x_3 \\ \text{s.t.} & 2x_1 + 2x_2 + x_3 = 3 \\ & x_1, x_2, x_3 \in \{0, 1\}. \end{array}$$

So, such an $a > 0$ does not always exist.

Problem 6: Typography. (Practice) Bertsimas & Tsitsiklis, Exercise 11.6.

Solution: Let $f(i)$ be the optimal attractiveness of the sequence $i, i+1, \ldots, n$. We want to find $f(1)$. A dynamic programming algorithm is as follows

$$f(i) = \max_{j \geq i} c_{ij} + f(j+1)$$

and initial condition $f(n+1) = 0$. We compute the values of f recursively for $n, n-1, \ldots, 1$.

Problem 7: A 1/2−approximation algorithm for TSP. (Practice) Formalize the proof seen in class. Bertsimas & Tsitsiklis, Exercise 11.14.

Solution: The final walk goes through the edges $T \cup M$ of the graph G' exactly once. Therefore, the cost of the walk is $Z_H = \sum_{(i,j)\in T} c_{ij} + \sum_{(i,j)\in M} c_{ij}$. We first prove that $\sum_{(i,j)\in T} c_{ij} \leq Z_{IP}$ Indeed, consider the optimal TSP path. It visits all nodes by making a cycle. Deleting any edge forms a spanning tree, which has higher cost than the cost of the minimum spanning tree which is $\sum_{(i,j)\in T} c_{ij}$ by definition. Deleting an edge only decreases the cost, therefore $Z_{IP} \geq \sum_{(i,j)\in T} c_{ij}$.

We now turn to the matching part and aim to show that $Z_{IP} \geq 2\sum_{(i,j)\in M} c_{ij}$. We denote the elements of $S = \{s_1, \ldots, s_p\}$ according to their order of visit in the optimal TSP solution i.e. the path visits s_1 then s_2 etc. until s_p then returns to s_1. Note that p is even. Indeed, in T, the sum of degrees double counts each edge (given that there are $n-1$ edges in a tree, this sum is equal to $2(n-1)$) which is even. Thus, there must be an even number of odd degrees. We can therefore write $p = 2k$. By triangular inequality we have

$$Z_{IP} \geq c_{s_1,s_2} + c_{s_2,s_3} + \ldots + c_{s_{2k},s_1} = (c_{s_1,s_2} + c_{s_3,s_4} + \ldots + c_{s_{2k-1},s_{2k}}) + (c_{s_2,s_3} + c_{s_4,s_5} + \ldots + c_{s_{2k},s_1})$$

Each of the two terms on the right hand side corresponds to the cost of a perfect matching, which is then larger than the cost of the minimum cost perfect matching $\sum_{(i,j)\in M} c_{ij}$. Thus,

$$Z_{IP} \geq 2 \sum_{(i,j)\in M} c_{ij}.$$

Putting everything together gives

$$Z_H = \sum_{(i,j)\in T} c_{ij} + \sum_{(i,j)\in M} c_{ij} \leq Z_{IP} + \frac{1}{2} Z_{IP} = \frac{3}{2} Z_{IP}. \tag{5}$$