Discrete Optimization
ISyE 6662 - Spring 2023
Homework 2

Instructor: Alejandro Toriello
TA: Filipe Cabral

February 6, 2023.

1. Consider the integer program

$$\begin{array}{ll} \inf & x_1\sqrt{3} - x_2 \\ \text{s.t.} & x_1\sqrt{3} - x_2 \geq 0, \\ & x_1 \geq 1, \\ & x \in \mathbb{Z}^2. \end{array}$$

Prove that no feasible solution has objective equal to zero, but that there are feasible solutions with objective value arbitrarily close to zero. Hint: Here is one possible strategy. Start with a feasible solution (x_1, x_2), such as $(1, 1)$. Show how you can construct a solution with a better objective value, where the new solution's values are functions of x_1 and x_2. Note that for any choice of x_1, the maximal choice of x_2 makes the objective of the corresponding solution less than 1; you may want to use this in your analysis.

Answer$_1$: Denote by ν the optimal value of this problem. It follows from the first constraint that $\nu \geq 0$. Because $(x_1, x_2) = (1, 1)$ is a feasible solution we also know that $\nu \leq \sqrt{3} - 1 < 0.75$. We will prove that given any feasible solution (x_1, x_2), we can always find another one with lower objective function.

Indeed, no solution (x_1, x_2) can have objective value 0 since this would imply that $\sqrt{3}$ is a rational number, i.e., $\sqrt{3} = x_2/x_1$. Let $\epsilon = x_1\sqrt{3} - x_2$ be the objective value of a feasible solution (x_1, x_2) such that $0 < \epsilon < 1$. This implies that x_2 is positive otherwise the objective value would be greater than 1.

Then,

$$\left.\begin{array}{rcl} \epsilon^2 & = & -2x_1x_2\sqrt{3} + (3x_1^2 + x_2^2) \\ \epsilon & = & x_1\sqrt{3} - x_2 \end{array}\right\} \implies \epsilon - \epsilon^2 = \underbrace{(x_1 + 2x_1x_2)}_{x_1' \geq 1}\sqrt{3} - \underbrace{(3x_1^2 + x_2^2 + x_2)}_{x_2'} < \epsilon.$$

Note that $x_1' := x_1 + 2x_1x_2$ and $x_2' := 3x_1^2 + x_2^2 + x_2$ define a feasible solution with lower objective cost. Therefore, there are feasible solutions with objective value arbitrarily close to zero.

Answer$_2$: The function $f : \mathbb{Z}_+ \to [0, 1]$ defined as $f(d) = d\alpha - \lfloor d\alpha \rfloor$ is dense in $[0, 1]$ if α is an irrational number. Thus, given any number $w \in [0, 1]$ and $\epsilon > 0$ there exist $d \in \mathbb{Z}_+$ such that $|w - f(d)| < \epsilon$. The case $\alpha = \sqrt{3}$ and $w = 0$ implies that the infimum of the integer program is 0 and there are feasible solutions with objective value arbitrarily close to zero.

2. Let $G = (N, E)$ be an undirected graph. We covered two formulations to model spanning trees:

$$\begin{array}{ll} \displaystyle\sum_{e \in E} x_e = |N| - 1 & \\ \displaystyle\sum_{e \in E; e \subseteq S} x_e \leq |S| - 1, & \emptyset \neq S \subsetneq N \\ x \in \{0, 1\}^E & \end{array} \qquad \begin{array}{ll} \displaystyle\sum_{e \in E} x_e = |N| - 1 & \\ \displaystyle\sum_{e \in \delta(S)} x_e \geq 1, & \emptyset \neq S \subsetneq N \\ x \in \{0, 1\}^E. & \end{array}$$

The left-hand formulation ensures the subgraph is acyclic, while the right-hand one ensures it is connected, and we showed that these properties are equivalent if the subgraph has $|N|-1$. Consider the linear relaxations of the two formulations, where we replace the binary restrictions with $x \in [0,1]^E$. Show that the left-hand relaxation is stronger than the right-hand one: That is, show that the left-hand polyhedron is contained in the right-hand one, and show that the inclusion can be strict for some graph.

Answer: Let x be a feasible solution of the left-hand side linear relaxation. Given any subset $S \subseteq N$, the set edges E can be decomposed into a disjoint union of edges within S, $E(S)$, edges within $S^{\complement}$, $E(S^{\complement})$, and edges across S and $S^{\complement}$, $\delta(S)$, that is,

$$E = E(S) \,\dot\cup\, E(S^{\complement}) \,\dot\cup\, \delta(S).$$

Note here that $E(S) = \{e \in E \mid e \subseteq S\}$. Thus,

$$\left.\begin{aligned} |N|-1 &= \sum_{e\in E(S)} x_e + \sum_{e \in E(S^{\complement})} x_e + \sum_{e\in\delta(S)} x_e \\ &\leq (|S|-1) + (|S^{\complement}|-1) + \sum_{e\in\delta(S)} x_e \\ &= |N| - 2 + \sum_{e\in\delta(S)} x_e \end{aligned}\right\} \quad\Longrightarrow\quad 1 \leq \sum_{e\in\delta(S)} x_e.$$

For the strict inclusion, consider the feasible solution defined as the weights of the following graph:

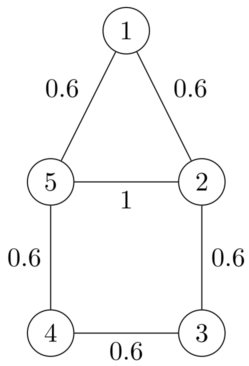

Figure 1: Strict inclusion graph example.

The solution satisfies the cut set constraint $\sum_{e\in\delta(S)} x_e \geq 1$ for every subset S such that $\emptyset \neq S \subsetneq \{1,\ldots,5\}$. However, the subtour elimination constraint is violated for $S = \{1,2,5\}$.

3. We've covered a couple of formulations for the TSP; here is another idea. For node set $N = \{1,\ldots,n\}$, consider a decision variable $x_{ik} \in \{0,1\}$ that equals one precisely when node i is in the tour's k-th position and is zero otherwise, for $k = 1,\ldots,n$. Give a linear integer programming formulation that uses these decision variables. You may need to also define additional decision variables. Prove your formulation's correctness.

   Answer: We create a constraint that assigns an order k for each node i, $\sum_{k=1}^{n} x_{ik} = 1$, for all i, and a constraint that assigns a node i to each order k, $\sum_{i\in N} x_{ik} = 1$, for all k.

   The TSP unit costs are for the tour's edges, so we need a variable to track that. Let $y_{ij}^k \in \{0,1\}$ be the binary variable that equal to 1 if the tour moves from node i to node j in the k-th position. We need to impose the constraint $y_{ij}^k = x_{jk} \cdot x_{i,k-1}$ since we can only traverse the edge ij if node i is in the $(k-1)$-th position and node j is in the k-th position. We use the standard linearization to convert the product of binary variables into linear constraints:

$$x_{i,k-1} \geq y_{ij}^k, \quad x_{jk} \geq y_{ij}^k, \quad y_{ij}^k \geq x_{i,k-1} + x_{jk} - 1.$$

Below is the complete formulation of the TSP:

$$
\begin{array}{llll}
\min_{x,y} & \sum_{k=1}^{n} \sum_{\substack{i,j\in N \\ i\neq j}} w_{ij} y_{ij}^k & & \\
\text{s.t.} & \sum_{k=1}^{n} x_{ik} = 1, & i \in N & \text{(Exact one order for each node)} \\
& \sum_{i\in N} x_{ik} = 1, & k = 1,\ldots,n, & \text{(Exact one node for each order)} \\
& x_{in} \geq y_{ij}^1, \quad x_{j1} \geq y_{ij}^1 & i,j \in N;\ i \neq j, & \text{(Move from } i \text{ to } j \text{ in 1st postion - UB)} \\
& y_{ij}^1 \geq x_{in} + x_{j1} - 1, & i,j \in N;\ i \neq j, & \text{(Move from } i \text{ to } j \text{ in 1st postion - LB)} \\
& x_{i,k-1} \geq y_{ij}^k, \quad x_{j,k} \geq y_{ij}^k & k = 2,\ldots,n, \quad i,j \in N;\ i \neq j, & \text{(Move from } i \text{ to } j \text{ in k-th postion - UB)} \\
& y_{ij}^k \geq x_{i,k-1} + x_{jk} - 1, & k = 2,\ldots,n, \quad i,j \in N;\ i \neq j, & \text{(Move from } i \text{ to } j \text{ in k-th postion - LB)} \\
& x_{ik} \in \{0,1\},\ y_{ij}^k \in \{0,1\} & k = 1,\ldots,n, \quad i,j \in N;\ i \neq j. &
\end{array}
$$

4. You are scheduling production runs in a plant over a planning horizon of T periods. In each period $t = 1,\ldots,T$, you have the following data:
   - $d_t \in \mathbb{Z}_+$: Units of demand that must be satisfied in period t.
   - $c_t \in \mathbb{R}_+$: Variable, per-unit production cost for period t.
   - $f_t \in \mathbb{R}_+$: Fixed production setup cost that is paid once if any production occurs in t.
   - $h_t \in \mathbb{R}_+$: Per-unit holding cost for units held in inventory at the end of period t.

   Your objective is to minimize total cost over the planning horizon while meeting demand.

   a) Consider the formulation

$$
\begin{array}{lll}
\min\limits_{q,s,z\geq 0} & \sum\limits_{t=1}^{T} f_t z_t + c_t q_t + h_t s_t & \\
\text{s.t.} & q_t + s_{t-1} - s_t = d_t, & t = 1,\ldots,T \\
& q_t \leq z_t \sum_{\tau=t}^{T} d_\tau, & t = 1,\ldots,T \\
& s_0 = 0; \quad z \in \{0,1\}^T. &
\end{array}
$$

   The variables q_t represent production quantity in t, s_t represents inventory at the end of t, and z_t indicates if production occurred in t. Prove its correctness.

   Answer: The correctness of the inventory balance equation is straightforward. The only non-trivial question is $\sum_{\tau=t}^{T} d_\tau$ is a valid lower bound for the production variable q_t. Indeed, if we sum the inventory balance equation $q_\tau + s_{\tau-1} - s_\tau = d_\tau$ for τ from t to T we get the following relations:

$$
\sum_{\tau=t}^{T} d_\tau = \sum_{\tau=t}^{t} q_\tau + s_{t-1} - s_T \geq q_t - s_T,
$$

   where the last inequality follows from the non-negativity constraints. Because the inventory cost is non-negative and the planning horizon ends at T, there exists an optimal solution with final inventory s_T equal to 0. Thus, $\sum_{\tau=t}^{T} d_\tau$ is a valid upper bound for q_t.

   b) Give a different formulation in which you do not use inventory variables. Instead, use variables $q_{t,t'}$ to indicate the amount of production in period t used to meet demand in period $t' \geq t$. Prove your formulation's correctness.

   Answer: Because the initial inventory s_0 is 0 and the final inventory s_T must be 0 at an optimal solution we consider the constraint $q_t = \sum_{\tau=t}^{T} q_{t\tau}$. This represents production at time t that will be sent to a future time period τ. Then, the demand d_t must be met by the sum of production quantities of previous time periods that were sent to t, i.e., $\sum_{\tau=1}^{t} q_{\tau t} = d_t$. The inventory cost to hold the production quantity $q_{t\tau}$ from time t to τ is $\sum_{l=t}^{\tau-1} h_l$. Below is the complete formulation:

$$
\begin{array}{llll}
\min\limits_{q,z\geq 0} & \sum_{t=1}^{T}\Big(f_t z_t + c_t q_t + \sum_{\tau=t+1}^{T} q_{t\tau}\cdot \sum_{l=t}^{\tau-1} h_l\Big) & & \\
\text{s.t.} & \sum_{\tau=1}^{t} q_{\tau t} = d_t, & t=1,\dots,T, & \text{(Demand constraint)} \\
& q_t = \sum_{\tau=t}^{T} q_{t\tau}, & t=1,\dots,T, & \text{(Production partition)} \\
& q_t \leq z_t \sum_{\tau=t}^{T} d_\tau, & t=1,\dots,T, & \text{(Production setup)} \\
& q_t \geq 0, \quad q_{t\tau}\geq 0, \quad z_t\in\{0,1\}, & t=1,\dots,T, \quad \tau=t,\dots,T. &
\end{array}
$$

5. In an undirected graph, a clique is a set of nodes in which every pair is connected by an edge. The CLIQUE problem in a graph G asks if there is a clique in G of size k or greater. Give a polynomial reduction of 3-SAT to CLIQUE.

   Answer: Recall that every 3-SAT instance is defined by a set of clauses $C = \{c_1, \dots, c_n\}$ on a finite set of variables U. Each clause c_i is a set containing 3 literals, that is, $c_i = \{l_{i1}, l_{i2}, l_{i3}\}$, and each literal l_{ij} is given by a Boolean variable x or its negation $\neg x$, where $x \in U$.

   A clause c_i represents the logical disjunction (or) $l_{i1} \vee l_{i2} \vee l_{i3}$, which can be true or false depending on the assignment of the Boolean variables. The 3-SAT is the problem of determining whether or not there is an assignment of the Boolean variables $x \in U$ that satisfies all the clauses $c_1, \dots, c_n$ at once, i.e., that satisfies the logical conjunction of clauses $c_1 \wedge \dots \wedge c_n$. We say that the 3-SAT is *satisfiable* when the answer to this question is yes.

   To create a polynomial reduction of 3-SAT to CLIQUE we need to construct a graph that represents a 3-SAT instance. Indeed, let a node be the pair (c_i, l_{ij}), where c_i is a clause and l_{ij} is a literal that belongs c_i:

$$N = \big\{(c_i, l_{ij}) \bigm| i = 1, \dots, n,\ j = 1, 2, 3\big\}.$$

   We construct an edge between nodes (c_i, l_{ij}) and (c_r, l_{rs}) if the clauses c_i and c_r are different and the literals l_{ij} and l_{rs} are not the negation of each other:

$$E = \Big\{\big\{(c_i, l_{ij}), (c_r, l_{rs})\big\} \Bigm| c_i \neq c_r,\ l_{ij} \neq \neg l_{rs},\ \forall i, j, r, s\Big\}.$$

   It is straightforward to see that this reduction is polynomial in the instance of the 3-SAT. The intuition here is that if two nodes (c_i, l_{ij}) and (c_r, l_{rs}) are connected by an edge then both clauses c_i and c_r are satisfiable if we make both literals l_{ij} and l_{rs} equal to true.

   Indeed, we prove that a 3-SAT instance (U, C) is satisfiable if, and only if, the graph $G = (N, E)$ has a clique of size $n = |C|$. If the 3-SAT instance (U, C) is satisfiable then there is an assignment of the Boolean variables such that some literal l_{ij} of each clause c_i is true. The set K of those pairs (c_i, l_{ij}) define a subset of nodes of cardinality n. Note that K is a clique since two different nodes (c_i, l_{ij}) and (c_r, l_{rs}) in K have true literals l_{ij} and l_{rs}, which implies that l_{ij} cannot be the negation of l_{rs}.

   Conversely, suppose the graph G has a clique K of size n. For any two nodes (c_i, l_{ij}) and (c_r, l_{rs}) in K, the literals l_{ij} and l_{rs} are the same or they are associated to different variables. In either case, the 3-SAT instance can be satisfied by assigning true to the literal l_{ij} of each clause c_i, where $(c_i, l_{ij}) \in K$.

6. Problem A: Let $a_i \in \mathbb{N}$ for $i \in N = \{1, \dots, n\}$, let $b \in \mathbb{N}$. Is there a set $S \subseteq N$ satisfying $\sum_{i \in S} a_i = b$? Problem B: Let $\alpha_i \in \mathbb{Q}$ for $i \in N$. Is there is some $S \subseteq N$ with $f\big(\sum_{i \in S} \alpha_i\big) \geq \beta$, where $f(x) = \sqrt{x} - rx$ for some arbitrary $r > 0$.

   a) Show that 3-SAT polynomially reduces to A. Hint: Use very large numbers, where each digit encodes a variable or clause.

      Answer: Consider an instance of the 3-SAT problem (U, C) such that $U := \{x_1, \dots, x_k\}$ is the set of variables and $C := \{c_1, \dots, c_n\}$ is the set of clauses. For each variable $x_i \in U$, define the numbers

$$y_i = 10^{i+n} + \sum_{\substack{l=1;\\ x_i \in c_l}}^{n} 10^l, \quad \text{and} \quad z_i = 10^{i+n} + \sum_{\substack{l=1;\\ \neg x_i \in c_l}}^{n} 10^l$$

and for each clause c_j, define the numbers

$$t_j = 10^j \quad \text{and} \quad s_j = 10^j.$$

The target number is defined as

$$\begin{aligned} b &= \sum_{i=1}^{k} 10^{i+n} + 3 \cdot \sum_{l=1}^{n} 10^l \\ &= \underbrace{11 \cdots 1}_{k} \underbrace{33 \cdots 3}_{n}. \end{aligned}$$

Suppose that the 3-SAT instance is satisfiable, i.e., there is an assignment of the Boolean variables $x_1, \ldots, x_k$ that satisfies the logical conjunction $c_1 \wedge \cdots \wedge c_n$. To define a solution to the subset sum, select y_i such that x_i is true otherwise select z_i:

$$\sum_{\substack{i=1; \\ x_i=\text{true}}}^{n} y_i + \sum_{\substack{i=1; \\ x_i=\text{false}}}^{n} z_i = \underbrace{11 \cdots 1}_{k} \underbrace{d_n d_{n-1} \cdots d_1}_{n}.$$

Because this Boolean assignment satisfies the logical conjunction $c_1 \wedge \cdots \wedge c_n$ we have that each d_l is greater than or equal to 1, for all $l = 1, \ldots, n$. Thus, we can sum the variables t_j and s_j to add up d_j to 3 and meet the target b.

Conversely, suppose that our instance of the subset sum problem has a solution. Because each of the variables t_j and s_j for the j-th digit can sum only up to 2 there must exist a variable y_i or z_i that completes the sum up to 3. This implies that either x_i or $\neg x_i$ belongs to the class c_j, for some $i \in \{1, \ldots, n\}$. Because the last k digits of b are all 1's then either x_i or $\neg x_i$ is in the subset sum representation but not both. Otherwise the $(i+n)$-th digit would be 2, instead of 1. Therefore, the 3-SAT instance is satisfiable.

b) Show that A polynomially reduces to B.

Answer: The idea of this question is to note that f is a strictly concave function, therefore has a unique maximizer. We can use the explicit formula for the maximizer x^* of f and $\nu = f(x^*)$ to define $\alpha_i \in \mathbb{Q}$ and β, respectively.

Indeed, the first and second derivative of f is given by:

$$\begin{aligned} f'(x) &= \frac{1}{2\sqrt{x}} - r \\ f''(x) &= -\frac{1}{4x^{3/2}} < 0, \end{aligned}$$

for every point $x > 0$. Because the second derivative is negative for every $x > 0$, we conclude that f is strictly concave. From the first derivative, we have that the maximizer x^* is such that $1/(2\sqrt{x^*}) - r = 0$, that is, $x^* = 1/4r^2$. So, the maximum is $\nu = f(x^*) = 1/4r$.

We scale b by a constant c to make it the maximizer of f:

$$cb = \frac{1}{4r^2} \quad \Longrightarrow \quad c = \frac{1}{4r^2 b}.$$

By scaling $\sum_{i \in S} a_i$ by c as well, we are able to define our instance for problem B:

$$\alpha_i = c \cdot a_i, \qquad \beta = \frac{1}{4r},$$

for every $i \in N$. Note that the reduction is polynomial in the size of the instance of problem A.

Suppose the answer to problem B is yes, i.e., there exists some $S \subseteq N$ with $f(\sum_{i \in S} \alpha_i) \geq \beta$. Then, $\sum_{i \in S} \alpha_i$ is the maximizer of f which is equal to $1/4r^2$:

$$\frac{1}{4r^2} = \sum_{i \in S} \alpha_i = \frac{\sum_{i \in S} a_i}{4r^2 b} \quad \Longrightarrow \quad b = \sum_{i \in S} a_i.$$

So, the answer to problem A is also yes. Conversely, suppose the answer to problem A is yes, i.e., there exists a set $S \subseteq N$ such that $\sum_{i \in S} a_i = b$. Then,

$$f\left(\sum_{i \in S} \alpha_i\right) = f(c \cdot b) = \frac{1}{4r} = \beta.$$

Thus, the answer to problem B is also yes.