



THE UNIVERSITY OF
SYDNEY

The University of Sydney

School of Computer Science

COMP3530 Assignment 2

Author:

Gordon LIU (460328360)

An Assignment submitted for the UoS:

COMP3530: Discrete Optimization

September 21, 2022

Question 1

Task: Prove that the Strong Duality Theorem (SDT) also holds for Linear Programs with the following form:

$$\begin{array}{ll} \min. & c \cdot x \\ \text{subject to} & Ax = b \\ & x \text{ free} \end{array}$$

Solution Attempt: The dual program of (P) , (D) would have the following form:

$$\begin{array}{ll} \max. & b \cdot y \\ \text{subject to} & A^T y = c \\ & y \text{ free} \end{array}$$

Let us attempt to transform the above program (P) into an equivalent one in standard form (P') . We can do this by mapping the unrestricted variable (x) in (P) as a difference of restricted variables (x^+, x^-) (and vice-versa as solved in Week 1 ex 6):

$$x = (x^+ - x^-)$$

where

$$x^+ = \max(0, x), \quad x^- = -\min(0, x), \quad x^+ \geq 0, \quad x^- \geq 0$$

This gives the following expanded standard form of (P') :

$$\begin{array}{ll} \min. & c \cdot x^+ - c \cdot x^- \\ \text{subject to} & Ax^+ - Ax^- = b \\ & x^+ \geq 0 \\ & x^- \geq 0 \end{array}$$

Converting (P') into its dual (D') gives:

$$\begin{array}{ll} \max. & b \cdot y \\ \text{subject to} & A^T y \leq c \\ & -A^T y \leq -c \\ & y \text{ free} \end{array}$$

As the first and second $(-A^T y \leq -c \equiv A^T y \geq c)$ constraints of (D') constrain the solution from upper and lower bounds, they can be combined into the following singular constraint:

$$A^T y = c$$

Thus (D') has the form:

$$\begin{array}{ll} \max. & b \cdot y \\ \text{subject to} & A^T y = c \\ & y \text{ free} \end{array}$$

which is the same form as (D) . Thus, as the SDT holds for (P') and (D') , and we have established that (P) and (D) are equivalent to (P') and (D') respectively, the SDT must also hold for linear programs with forms like (P) .

Question 2

Given n jobs, each having a fixed processing time p , arrival times a_1, \dots, a_n , and deadlines d_1, \dots, d_n . For a given subset of jobs $X \subseteq [1, \dots, n]$, a schedule is an assignment $t : X \rightarrow \mathbb{R}$ with the following conditions:

1. For any distinct $i, j \in X$, $|t_i - t_j| \geq p$
2. For job $i \in X$, $t_i \geq a_i$
3. For job $i \in X$, $t_i \leq d_i - p$

Task: "to show an instance where the subset of system defined by schedulable subsets of jobs is not a matroid."

Solution attempt:

Let us consider a set of 3 schedulable jobs where $p = 1.5$: $X = \{(0.5, 2), (1.5, 3), (2, 4)\}$. Two feasible schedules would be the following subsets: $t_1 = \{(0.5, 2), (2, 4)\}$ and $t_2 = \{(1.5, 3)\}$. This instance would not be a matroid because there exists no job in t_1 that could be added to t_2 where it can still fulfill the conditions of a schedule.

Question 3

Let $X \subseteq \mathbb{R}^2$ be a set of n points on a plane such that no 3 points are co-linear. A X -polygon is a polygon whose vertices are a subset of points on X . A polygon cover of X is a collection of X polygons.

Consider the following Integer Program (IP) for finding a minimum length polygon cover:

$$\begin{aligned} \min. \quad & \sum_{u,v \in X} d_{uv} x_{u,v} \\ \text{s.t.} \quad & \sum_{v \neq u} x_{u,v} = 2 \quad \forall u \in X \\ & x_{u,v} \in \{0, 1\} \quad \forall \{u, v\} \in \binom{X}{2} \end{aligned} \quad (\text{IP})$$

d_{uv} is the straight-line distance between u and v , and $\binom{X}{2}$ are all the unordered pairs of points in X .

Task a: Argue that this is valid formulation for minimum length polygon cover.

Task b: Prove that the linear relaxation of (IP) is in fact integral.

Task a solution attempt:

From the the IP above, we understand that the variable x_{uv} is a edge between a pair of points (u, v) within X that may or may not exist. This also implies that loops are non-existent. The objective function of the IP states to minimise the total sum of the distance of edges. The constraint of the IP states that for every point in X , there are two edges connecting to that point may exist.

Let X be a set of 3 points consisting of points $\{(0,0), (0,1), (1,1)\}$. A feasible solution to the IP above would consist of a triangle connecting all three points. This solution satisfies the first constraint (as there are two edges for every distinct point), the integral constraint, and has a distance cost of $2 + \sqrt{2}$.

Similarly, for the feasible solution to the min-length polygon cover, there is only one polygon cover that can exist spanning the points in X , as there is only one X -polygon that can exist in X , that being a triangle spanning all the points in X . As a polygon is closed (i.e. the number of vertices equal the number of edges), each point in the solution will have two distinct edges. Furthermore, as straight lines are the shortest distance between two points, the feasible solution would also have the same cost as the IP.

Let X be a another set consisting of the following points: $\{(0,0), (0,1), (1,0), (1,1)\}$. An optimal solution to the minimum-length polygon cover of X would consist of a single X -polygon (i.e a square spanning all 4 points), which will have a cost of 4. Any additional polygons would only increase the edges, and thus the length, which discount the solution from being optimal.

With regards to the IP, though can exist 6 edges between the points in X , the shortest straight-line distance between any two distinct points in X would equal 1. The optimal solution of this IP will not include the two edges connecting points $\{(1,0), (0,1)\}$ and $\{(0,0), (1,1)\}$, as they have a distance of $\sqrt{2}$ which is greater than 1 and their inclusion

would violate the constraint of the IP (i.e. every point can only have two edges connecting it at a time). Thus the optimal solution to the IP, the distance of the sides of a square connecting all 4 points, would also have the same cost as the minimum-length polygon cover optimal solution, which is 4.

Thus the formulation of the IP as a minimum-length polygon cover is correct, as the cost is the same between feasible solutions and optimal solutions between the former and latter.

Generally, an optimal integral solution to the min-length polygon cover solution must always consist of a singular polygon as the definition of a polygon states there are equal number of edges to vertices. Therefore solutions with multiple polygons implies that there are greater number of edges than vertices, which would mean that the solution is not optimal.

Task b solution attempt:

As Julian stated that the LP would only be integral if we relax the integrality constraint as $x_{uv} \geq 0$ and not, $0 \leq x_{uv} \leq 1$ I will base my attempt on this, but the attempt is incomplete as I attempted to prove the LP is integral using a proof by contradiction, which is probably an incorrect method.

Let us construct a LP from the IP above by relaxing the integrality constraint.

$$\begin{aligned} \min. \quad & \sum_{u,v \in X} d_{uv} x_{u,v} \\ \text{s.t.} \quad & \sum_{v \neq u} x_{u,v} = 2 \quad \forall u \in X \\ & x_{u,v} \geq 0 \quad \forall \{u,v\} \in \binom{X}{2} \end{aligned}$$

For the LP to be integral, we will need to show that there will always be an optimal solution to the LP where all integral constraints x_{uv} are integral.

Let us attempt a proof by contradiction. Assuming there is optimal solution to the LP consisting of integral edges I and fractional edges F ($f \in F | 0 < f < 2, f \neq 1$). In order to follow the constraint that sum of edge weights to any given point must equal 2, if a vertex u some fractional edge $f_{uv} \in F$, then u must have at least some other fractional edge $f_{uw} \in F$ (given that $w, u, v \in X$) such that they sum to an integral value. However, this further implies that vertices as w, v must also have fractional edges.

Question 4

Given a subset $X \subseteq [0, 1]^2$ of n points of the unit square, let $f(X)$ be the value of (IP) for X . D_n is defined to be a probability distribution over subsets X with $|X| = n$ such that each point is chosen independently and uniformly at random from the unit square. Define $g(n) = E_{X_n}[f(X)]$.

Task:

- Evaluate $f(X)$ for a random $X \sim D_n$ for n increasing powers of 2 until the solver takes more than a few seconds to evaluate $f(X)$.
- For each of these feasible powers of n , sample enough X to reduce the variance of the sample mean estimate of $g(n)$
- plot the sample mean estimator of $g(n)$
- propose a function $h(n)$ such that $g(n) = \Theta(h(n))$.

Part 1: Using the Gurobi Linear Solver demonstration from week 3 as a base, I attempted to model IP into Gurobi. This was achieved by creating a complete graph consisting of n points within the unit square, randomly selected using uniform distribution for D_n . Edges between all the points were added, with a distance variable calculated using the Euclidean Distance between the two points. This was applied to the LP solver, following the objective function and constraints of (IP). The number of points n sampled $|X|$ ranged from $2^2 < n < 2^9$, with corresponding $f(X)$ as found in Table 1 below:

$ X = 2^n$	Num. Vertices	Num. Edges	$f(X)$	Runtime (s)
2^2	4	6	1.849	0.032
2^3	8	28	2.104	0.042
2^4	16	128	2.990	0.056
2^5	32	496	4.089	0.090
2^6	64	2016	6.290	0.1134
2^7	128	8128	8.153	0.374
2^8	256	32640	11.701	1.712
2^9	512	130816	16.25	5.3289

Table 1: $f(x)$ and Run times for increasing 2^n (3 d.p)

Part 23: For each $g(n)$ with exponentially increasing n , X was sampled 16 times and the resulting sample mean of the values was taken. The sample mean of $g(n)$ was then plotted against n to result in plot 1 in the next page:

Part 4: From plot 1 above, and given the values of n , we can conclude that the sample mean of $g(n)$ grows logarithmically to a certain aspect with regards to n . Plotting the sample mean of $g(n)$ against $\log_2(n)$ gives the following plot however:

This shows that the relationship between the sample mean of $g(n)$ and $\log_2(n)$ is not linear, however. The shape of the plot appears to indicate that there a higher-order

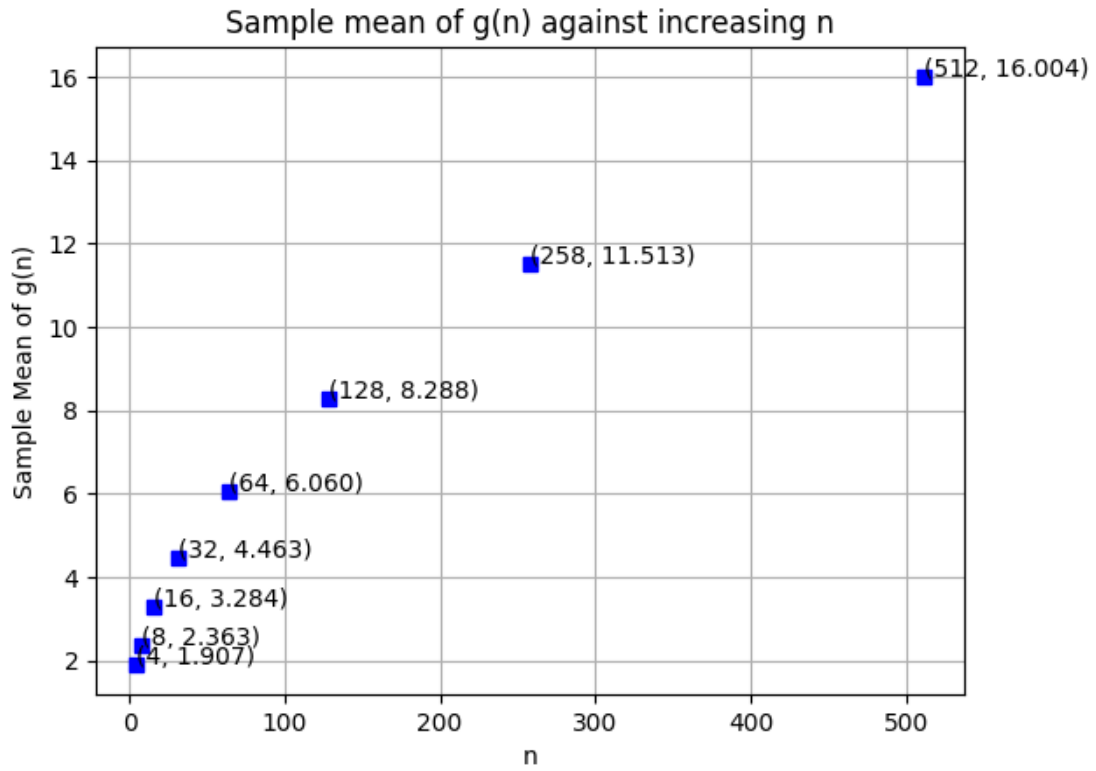


Figure 1: sample mean of $g(n)$ against n

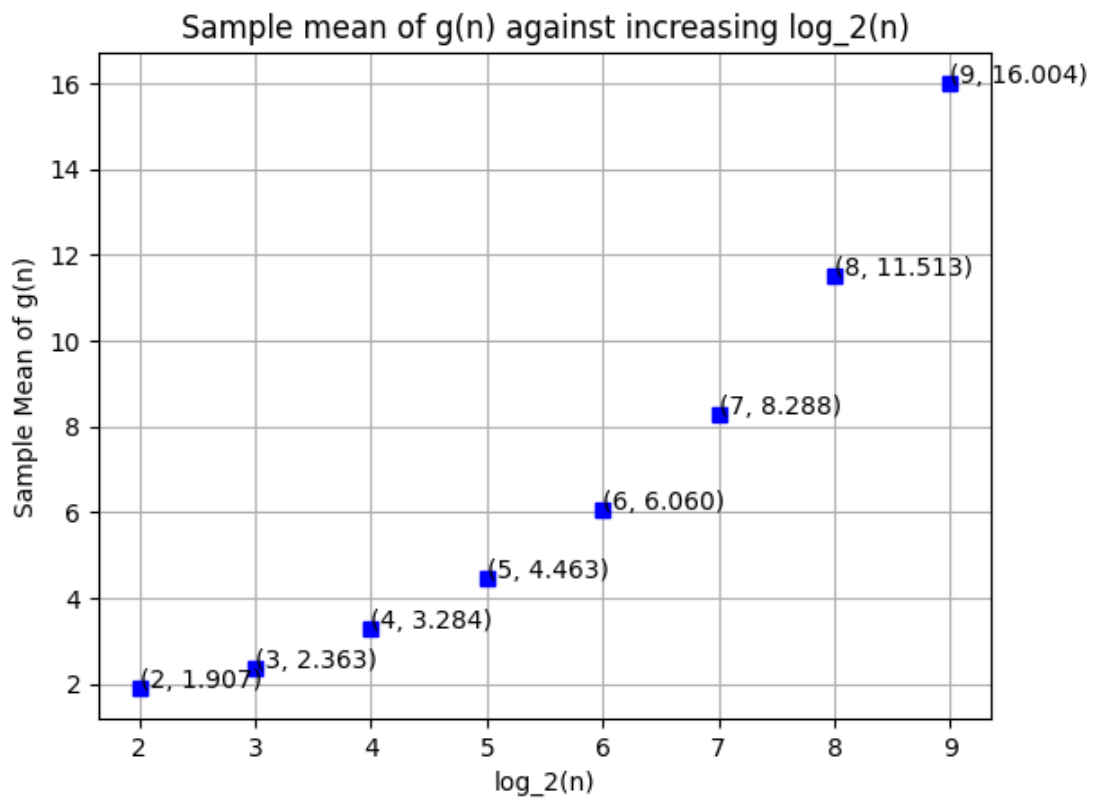


Figure 2: Sample mean of $g(n)$ against $\log_2(n)$

power within the relationship which leads me to the following formula for $h(n)$:

$$\begin{aligned}h(n) &= \log_2(n^m) \\ g(n) &= \Theta(\log_2(n^m))\end{aligned}$$

for some arbitrary m where $m \geq 1$.

The code, figures and sampling data can be found in the following github repository:
<https://github.com/Gordon-4389/COMP3530-A2>