CS 532 Homework 2

Qilin Ye

October 4, 2024

Solution to problem 1[1]. First, let us assume that a given maximum horizon τ is given. Our goal is to find whether there exists a plan that allows all agents to move from S to T within M steps.
We will construct a time-dependent graph $G_\tau = (V_\tau, E_\tau)$ as follows:

1. Introduce a super source s and a super sink t.
2. For each node $v \in V$, create 2τ copies of it, labeled as $v_0, v_0', v_1, v_1', \cdots, v_\tau, v_\tau'$.
3. Set G_τ to be capacitated with edge capacities 1 for each of the following edges.
   (a) For each $v_i \in V_\tau$ and $i \le \tau$, add a directed edge (v_i, v_i') with capacity 1. *This prevents conflict at node v at time i. More on this later.*
   (b) For each $v_i' \in V_\tau$ and $i < \tau$, draw a directed edge (v_i', v_{i+1}). *This represents a "no move" at node $v \in V$.*
   (c) For each edge $(u, v) \in E$ and $i < \tau$, draw an edge (u_i', v_{i+1}). *This represents moving from u to v in E.*
   (d) For each $s_i \in S$, draw an edge from the super source s to $(s_i)_0$. Likewise, draw $(t_j)_\tau'$ to super sink t.

The reason for such construction is that **G_τ admits a flow of value k if and only if the original *makespan* can be achieved in under τ steps**. To see this, first observe that s is incident on k edges. Thus, G_τ has a max flow of value of k iff all of them are saturated. And by flow conservation, all k incoming edges into t must also be saturated. Since the edge capacities are integral, there exists an integer max flow, and because capacities are 1, the flow along all edges is binary. Any $s-t$ path in G_τ assumes the form

$$s \to \underbrace{(s_i)_0 \to (s_i)_0'}_{\text{corresponds to } s_i} \to \cdots \to \underbrace{(t_j)_\tau \to (t_j)_\tau'}_{\text{corresponds to } t_j} \to t$$

which, by grouping the intermediate nodes (excluding s and t) into pairs, allows us to recover a walk on G. Thus, decomposing an $s-t$ flow on G_τ translates to decoding a sequence of actions for each agent.
Furthermore, by $3(a)$, any node of form v_i can only have a total in-flow of 1 since the only edge leaving it is $v_i \to v_i'$, which has capacity 1. This ensures that at any time i and any node $v \in V$, at most one agent is allowed to visit it. This prevents collision.
Having described the core of our algorithm, it remains to repeat it with different values of max horizon. Thus, we perform a binary search on τ in $[0, |V|]$, since the agent cannot visit more than $|V|$ nodes.
Overall, we have $\mathcal{O}(\log|V|)$ iterations, where in each iteration new solve a flow problem on a graph whose size is polynomial w.r.t. $|V|$ and $|E|$, and whose runtime is polynomial w.r.t. the graph size when using appropriate algorithms (e.g. push-relabel). Hence the runtime overall is polynomial. We omit the exact bound.

[1] Rico Zhu and I had a similar approach, and we had a brief discussion on the optimization of our solutions.

Solution to problem 2. (i) Following the notation used in-class, we define a function $T : U \times V \to \mathbb{R}_{\geqslant 0}$ and formulate the LP as

$$\max_{(u,v)} w(u,v)T(u,v) \qquad \text{subject to} \qquad \begin{cases} \sum_j T(u,v_j) = 1 & \text{for each } u \in U \\ \sum_i T(u_i,v) = 1 & \text{for each } v \in V. \end{cases}$$

(ii) Let T_f be a fractional optional solution to the above LP. We show that **if T_f is not integral, then there exists an equally optimal solution, T', with at least one more edge attaining integer values, i.e.,**

$$|\{(u,v) : T'(u,v) \in \mathbb{N}\}| \geqslant |\{(u,v) : T_f(u,v) \in \mathbb{N}\}|.$$

Assuming the graph is finite, this ensures the existence of an integral solution.

To prove this claim, we view edges $(u,v) \in U \times V$ as undirected. The idea is that we consider $S = \{(u,v) : T(u,v) \notin \mathbb{N}\}$ and show that by tweaking the values of some edges, we obtain another feasible solution with more edges assigned to integer values.

First, suppose S has a cycle C. Clearly, by the nature of bipartite graphs, C has an even number of edges. Let us index the edges as $e_1, \cdots, e_k$. Consider the following modifications on the coupling function T:

$$\epsilon \leftarrow \min_{1 \leqslant i \leqslant k} [\min\{T(e_i), 1 - T(e_i)\}]$$

$$T' : \begin{cases} T'(e_k) \leftarrow T(e_k) + \epsilon & \text{for } k \text{ odd} \\ T'(e_k) \leftarrow T(e_k) - \epsilon & \text{for } k \text{ even,} \end{cases}$$

$$T'' : \begin{cases} T''(e_k) \leftarrow T(e_k) + \epsilon & \text{for } k \text{ even} \\ T''(e_k) \leftarrow T(e_k) - \epsilon & \text{for } k \text{ odd,} \end{cases}$$

(This is analogous to the augmentation we have performed on flow problems.) By keeping ϵ sufficiently small, the modified T-values are still nonnegative, and by the nature of a cycle, each node in C is adjacent to one even- and one odd-indexed edge, so the $\pm\epsilon$ cancel, and the constraints still hold for both T' and T''. It is also clear that both T' and T'' are optimal: for each e_k,

$$T'(e_k) + T''(e_k) = 2T(e_k),$$

so $\sum_k w(e_k)T(e_k) = (\sum_k w(e_k)T'(e_k) + \sum_k w(e_k)T''(e_k))/2$. Finally, among all variables modified in *both T' and T'', at least one of them now becomes zero*, so either T' or T'' (or maybe both, but we don't know) now contains more integer-valued edges than T. This concludes the first half of the claim.

It remains to notice that S cannot be acyclic: every vertex v associated with some edge $e \in S$ must have degree $\geqslant 2$ in order to satisfy the LP constraints. Thus the claim is proven, and we are done.

(iii)

$$\max\left[\sum_i \varphi(u_i) + \sum_j \varphi(v_j)\right] \qquad \text{subject to} \qquad \begin{cases} \varphi(u_i) + \varphi(v_j) \geqslant w(u_i,v_j) & \text{for all } (u_i,v_j) \in E \\ \varphi(v) \in \mathbb{R}. \end{cases}$$

Solution to problem 3. Recall that in class we showed that a greedy k-`center` algorithm is 2-approximate. We will construct our k-supplier algorithm based on it.

Algorithm 1: 3-Approximation Algorithm for the k-Suppliers Problem

For each customer $c \in C$, find nearest supplier $s(c) \in S$ and record $d(c) \leftarrow d(c, s(c))$.
Initialize the set of centers $K \leftarrow \varnothing$.
Choose an arbitrary customer $c_1 \in C$ as the first center; update $K \leftarrow K \cup \{c_1\}$.
while $|K| < k$ **do**
    For each customer $c \in C$, compute $d(c, K) = \min_{c' \in K} d(c, c')$.
    Select the farthest customer $c^* = \arg\max_{c \in C} d(c, K)$.
    Update $K \leftarrow K \cup \{c^*\}$.
For each center $c_i \in K$, assign it to its nearest supplier $s(c_i) \in S$.
Assign each $c \in C$ to the closest center in K and the corresponding supplier.
Return: The set of suppliers $\{s(c_i) \mid c_i \in K\}$.

To see that this algorithm achieves, let r^* be the optimal value, meaning that for each $c \in C$, there exists $s \in S$ with $d(c, s) \leqslant r^*$. Therefore, if c_1, c_2 share the same supplier in the optimal solution, then $d(c_1, c_2) \leqslant 2r^*$. On the other hand, since $s(c_i)$ is *the* closest supplier to c_i, we certainly have $d(c_i, s(c_i)) \leqslant r^*$.
Combining the observations above, for any customer c, we will assign it to a center $c_i \in K$, and by triangle inequality, $d(c, s(c_i)) \leqslant d(c, c_i) + d(c_i, s(c_i)) \leqslant 3r^*$.

Solution to problem 4. (i) For each $c \in C$ we define a binary variable $x_c \in \{0, 1\}$, where $x_c = 1$ means the call is routed clockwise. We let L_i represent the load for each edge e_i, and L the overall $\min L_i$. Finally, for each call (i, j) and each edge e, define

$$\varphi_e^{\text{ccw}}(c) = \mathbf{1}[e \text{ is on the CW path from } i \text{ to } j]$$

and likewise for $\varphi_e^{\text{cw}}(c)$. The LP defined in this way is messy but it gets the job done:

$$\min L \text{ subject to } \begin{cases} L_i = \sum_{c \in C} [\varphi_{e_i}^{\text{cw}}(c) x_c + \varphi_{e_i}^{\text{ccw}}(c)(1 - x_c)] & \text{for } i \in [n] \\ L_i \leqslant L & \text{for } i \in [n] \\ x_c \in \{0, 1\}, L_i \geqslant 0. & \end{cases}$$

(ii) We will use **threshold rounding** with threshold 0.5, i.e., round x_c to 1 if the fractional optimal $x_c^* \geqslant 0.5$, and 0 otherwise. Let x_c^* and L_i^* denote the optimal fractional values, and x_c' and L_i' the post-rounding values. Observe that for each c and each e_i, $\{\varphi_e^{\text{ccw}}(c), \varphi_e^{\text{cw}}(c)\} = \{0, 1\}$, so we can write one as 1 minus the other. Fix any e_i, $i \in [n]$ and any $c \in C$. For convenience we let $\gamma = \varphi_{e_i}^{\text{cw}}(c)$. The term contributing to L_i^* is

$$L_{i,c}^* := \varphi_{e_i}^{\text{cw}}(c) x_c^* + \varphi_{e_i}^{\text{ccw}}(c)(1 - x_c^*) = \gamma x_c^* + (1 - \gamma)(1 - x_c^*).$$

If $x_c^* \geqslant 0.5$, then we round up to $x_c' = 1$, and the new contribution is

$$L_{i,c}' = \gamma \cdot 1 + (1 - \gamma)(1 - 1) = \gamma \leqslant 2x_c^* \gamma \leqslant 2(\gamma x_c^* + (1 - \gamma)(1 - x_c^*)) = 2L_{i,c}^*$$

and likewise, if $x_c^* < 0.5$, then we round down to $x_c' = 0$, and

$$L_{i,c}' = \gamma \cdot 0 + (1 - \gamma)(1 - 0) = 1 - \gamma \leqslant 2(1 - x_c^*)(1 - \gamma) \leqslant 2(\gamma x_c^* + (1 - \gamma)(1 - x_c^*)) = 2L_{i,c}^*.$$

Summing over all $c \in C$ we obtain $L_i' \leqslant 2L_i^*$, and taking maximum over i we see threshold rounding is 2-approximate.