CS 3510 Homework 3

Wenqi He, whe47

November 5, 2017

1

(a)

It's equivalent to

$$x_2 \geq x_1$$

$$x_2 \geq -x_1$$

(b)

Let $c = |a + b - 123|$ and $d = |4a + 3b - 234|$.
The first equation is equivalent to minimizing c subject to the constraint

$$c \geq |a + b - 123|$$

Similarly, the second equation is equivalent to minimizing d subject to

$$d \geq |4a + 3b - 234|$$

Combining these contraints and the conclusion from (a), the linear program is:

Minimize: $c + d$
Subject to:

$$c \geq a + b - 123$$

$$c \geq -a - b + 123$$

$$d \geq 4a + 3b - 234$$

$$d \geq -4a - 3b + 234$$

$$5a + 4b = 345$$

2

Minimize: $a - 10c$
Subject to:

$$a + 5b - c \leq 10$$

$$-a - 5b + c \leq -10$$

$$-5b + 4d \leq 2$$

Expressing each variable as a difference of two non-negative variables, it can be written as :
Minimize $a_+ - a_- - 10c_+ + 10c_-$:
Subject to:

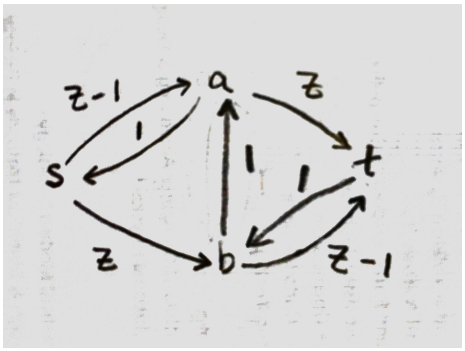
$$a_+ - a_- + 5b_+ - 5b_- - c_+ + c_- \leq 10$$

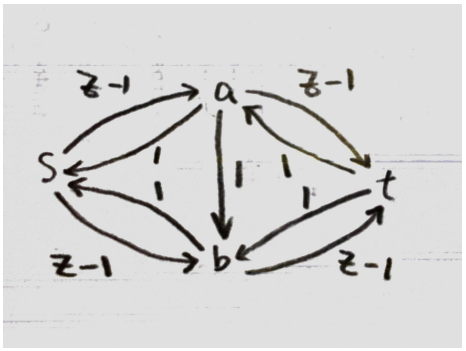
$$-a_+ + a_- - 5b_+ + 5b_- + c_+ - c_- \leq -10$$

$$-5b_+ + 5b_- + 4d_+ - 4d_- \leq 2$$

$$a_+, a_-, b_+, b_-, c_+, c_-, d_+, d_- \geq 0$$

3

(1) 

(2) 

After every 2 such iterations, the flow on path $s \to a \to t$ and $s \to b \to t$ each increases by 1, but the flow on edge $a \to b$ remains 0 because it was used twice in opposite directions. Following this algorithm, the maximum flow will be achieved after exactly $2Z$ iterations.

4

(a)

Suppose the vertices in graph G can be partitioned into A and B. For any $S_A \subset A$, suppose it's only incident to vertices in $S_B \subset B$, where $|S_B| < |S_A|$. Since the graph is d-regular, $d|S_A|$ edges come out of S_A while only $d|S_B| < d|S_A|$ edges go into S_B, therefore there must be some edges going from S_A to $B \setminus S_B$, which gives us a contradiction.

Thus, there does not exist any $S_A \subset A$ that is incident to a $S_B \subset B$ that has fewer than $|S_A|$ vertices. By Hall's Theorem, this is true if and only if such a bipartite graph has a perfect matching.

(b)

First of all, we can construct a max-flow problem from the perfect-matching finding problem if we connect a source s to each vertex in A and connect each vertex in B to a sink t. Suppose the edges only go from $\{s\}$ to A, from A to B and from B to $\{t\}$, and all edges have capacity 1. Then each perfect matching corresponds to one possible selection of edges that can be used to achieve the max flow $\frac{n}{2}$

Secondly, it can be shown that in a prefect matching if one pairing is changed all others must change as well, and therefore no two perfect matchings share any common edges.

Based on the above results, we can run Ford-Fulkerson algorithm to find one configuration that achieves max flow, and then safely remove all the edges between A and B that are used, and run the algorithm again on the remaining edges. Since exactly one edge is removed from each vertex, the remaining graph is still d-regular, and therefore all the nice properties are preserved. Repeat until all edges between A and B are removed, which would take exactly d iterations. Inside each iteration, the Ford-Fulkerson algorithm is performed on $\mathcal{O}(n)$ vertices and $\mathcal{O}(nd) \leq \mathcal{O}(n^2)$edges, and the maxflow is $\frac{n}{2} \leq \mathcal{O}(n)$. If we use Bellman-Ford to find each s-t path, the runtime of each iteration would be

$$\mathcal{O}(Fmn) \leq \mathcal{O}(n \cdot n^2 \cdot n) = \mathcal{O}(n^4)$$

In total, the runtime is

$$\mathcal{O}(dn^4) \leq \mathcal{O}(n^5)$$