

Caleb Logemann

MATH 566 Discrete Optimization

Midterm II

1. *Girth* of a graph $G = (V, E)$ is the length of the shortest cycle. Notice cycle in undirected graph has at least 3 vertices. Formulate an integer program that is solving the problem of shortest cycle in a graph.

In order to create this integer program, I will create a binary variable for each vertex in the graph G . Let $x_v \in [0, 1]$, denote whether or not vertex v is in the shortest cycle. Note that if v is in the shortest cycle then at least two of v 's neighbors must also be in the cycle. If only one neighbor was in the cycle, then there would be a path into v but not out of v , so it could not be a cycle. If there weren't any neighbors in the cycle, then v would be disconnected from the cycle. This reasoning can be expressed in the following constraint.

$$\sum_{u \in N(v)} (x_u) \geq 2$$

where $N(v)$ is the set of vectors that are neighbors of v . This is also assuming that v is in the cycle so $x_v = 1$. If v is not in the cycle then $x_v = 0$ and there may or may not be neighbors of v in the cycle. This does not require a constraint but a trivial constraint can be used to describe this situation.

$$\sum_{u \in N(v)} (x_u) \geq 0$$

This is clearly true because all x_u are binary variables. Now it is necessary to be able to describe both of these situations depending on the value of x_v . Note that if $x_v = 0$, then $2x_v = 0$ and if $x_v = 1$, then $2x_v = 2$. Using this fact both constraints can be expressed as

$$\sum_{u \in N(v)} (x_u) \geq 2x_v$$

This constraint forces there to be two neighbors in the cycle if v is in the cycle, but doesn't enforce anything if v is not in the cycle.

Now in order to find the girth of a graph, we wish to detect the smallest cycle. Therefore the integer program should minimize the number of vertices in the cycle or equivalently the objective function should be

$$\text{minimize } \sum_{v \in V} x_v$$

Just using the constraints given above and this objective function the integer program will simply set all x_v equal to zero. In order to force the program to find a nontrivial cycle we must force that at least 3 vertices are in the cycle. No cycle can be created with 0, 1, or 2 vertices. This constraint can be written as

$$\sum_{v \in V} x_v \geq 3$$

Therefore the full integer program is

$$(P) = \begin{cases} \text{minimize} & \sum_{v \in V} x_v \\ \text{subject to} & \sum_{v \in V} (x_v) \geq 3 \\ & \sum_{u \in N(v)} (x_u) \geq 2x_v \quad \forall v \in V \\ & x_v \geq 0 \quad \forall v \in V \\ & x_v \leq 1 \quad \forall v \in V \\ & x_v \in \mathbb{Z} \quad \forall v \in V \end{cases}$$

2. Recall that in Minimum Cost Flow, we were augmenting on minimum mean cycle. Show that if we allow augmentation on any augmenting cycle, the algorithm will not work right (bad convergence).
3. Suppose you have a graph $G = (V, E)$. Every edge is assigned a cost $c : E \rightarrow \mathbb{R}$. How to pick edges, such that the sum of costs of the picked edges is maximized and the picked edges do not contain a cycle (i.e., the set of picked edges forms a forest)? More formally,

$$\text{maximize} \left\{ \sum_{e \in X} c(e) : X \subseteq E, X \text{ has no cycle} \right\}.$$

Describe how to use one/some of the algorithms we had in class to solve this problem.

(I mean, how to modify the input to this problem such that it can be solved by some other algorithm? Alternatively, you can also describe an algorithm.)

Example: Notice that the edges can have negative weight. The thick edges in the following graph show one of the optimal solutions. (There are two optimal solutions. Your algorithm should find one.)

