1 The PC-TSP algorithm approach

The prize-collecting travelling salesperson is a variant of the familiar (NP) problem in which, instead of necessarily seeking a minimum weight cycle through every vertex in a graph (as required in the ordinary TSP), a penalty can be paid instead for missing vertices.

Anderson et al. propose an algorithm for solving KEPs which is inspired by the PC-TSP and which draws on algorithms for solving it. In particular, they suggest that this similarity is reasonable, given that finding longing chains in is useful in both the PC-TSP and KEPs.

1.1 Model

As under the Recursive Algorithm, prospective participants in a kidney exchange are associated with the vertices of a directed graph $\overrightarrow{G} = (V, \overrightarrow{E})$. Each donor-recipient pair is represented by a vertex $v \in P \subseteq V$ and each altruistic donor also by a vertex $v \in N \subseteq V$. A directed edge $e = (v, u)$ connects vertex v to a vertex u exactly when there is a compatible match from the donor at vertex v to the recipient at vertex u. Each such edge is also assigned a weight $w(e)$, intended to capture the importance of the (potential, compatible) transplant it represents.

For each vertex v, the set $\mathrm{in}(v)$ is of edges from compatible donors to v, while the set $\mathrm{out}(v)$ is of edges from v to compatible recipients. By construction, $\mathrm{in}(v)$ is empty for each $v \in N$.

The aim, as before, is to find the maximum weight subgraph of $\overrightarrow{G}$ subject to the constraint that cycles are of length at most $k \in \mathbf{Z}^{\geq 0}$.

Each edge in the $\overrightarrow{G}$ is associated with a binary decision variable y_e, which is equal to 1 if edge e is selected *unless* that edge forms part of a cycle of length no more than k. Denote by $\mathcal{C}_k$ the cycles in $\overrightarrow{G}$ of length no greater than k. Further binary decision variables z_C are thus created for each $C \in \mathcal{C}_k$, with $z_C = 1$ exactly when the cycle C is selected. The reason for this distinction between edges used in such cycles and those edges used which are not in cycles in $\mathcal{C}_k$ becomes apparent in the discussion of the constraints below.

1.1.1 Objective function

The objective function is given by

$$\sum_{e \in E} w_e y_e + \sum_{C \in \mathcal{C}_k} w_C z_C,$$

which we seek to maximize.

The weights w_C are simply calculated as the sum of the weights of the edges which make up the cycle C.

1.1.2 Constraints

Anderson et al. define (for every vertex $v \in V$) the *flow in* f_v^i and *flow out* f_v^o by

$$f_v^i = \sum_{e \in \text{in}(v)} y_e$$

and

$$f_v^o = \sum_{e \in \text{in}(v)} y_e.$$

These values are integer and will, in fact, be restricted to be binary.

Note particularly that when v is part of a cycle $C \in \mathcal{C}_k$, f_v^i and f_v^o are both set to 0. These values can therefore in general be interpreted as indicating whether an altruistic donor/donor-recipient pair is involved in kidney donation that is not part of a cycle up to length k.

The constraint

$$f_v^o + \sum_{C \in \mathcal{C}_k(v)} z_C \leq f_v^i + \sum_{C \in \mathcal{C}_k(v)} z_C \leq 1, \quad \forall v \in P,$$

where $\mathcal{C}_k(v)$ is the set of cycles of length at most k which include the vertex v, serves to ensure that each donor-recipient pair gives and receives at most one kidney and further that no such donation is made unless a kidney is received.

Likewise, imposing the constraint that $f_v^o \leq 1, \forall v \in N$ ensures that each altruistic donor gives at most one kidney.

The most involved constraint imposed requires that

$$\sum_{e \in \text{in}(S)} y_e \geq f_v^i, \quad \forall S \subseteq P, \quad \forall v \in S, \tag{1}$$

where in(S) is the set of directed edges ended at vertices in S. The authors indicate that this constraint is directly inspired by similar constraints for PC-TSP !INSERT REFERENCE!.

It permits cycles of length up to k and arbitrarily long chains but will make a proposed solution of the KEP infeasible if it contains a cycle longer than k:

- If v is part of a short cycle (i.e. part of $C \in \mathcal{C}_k$), then $f_v^i = 0$ and the constraint will be satisfied (**for this** v and any S) whatever the values of y_e.
- If $f_v^i = 1$, then either v is part of a long cycle $\bar{C}$ or a chain starting from an altruistic donor (since $v \in P$ cannot start a chain). If v is in a long cycle, setting $S = \bar{C}$[1], we see that the constraint is not satisfied.
- On the other hand, the constraint is not broken by v in a long chain: whatever selection of $S \subseteq P$ is made with $v \in S$, a vertex v in a chain requires a flow from an altruistic donor to S, so $\sum_{e \in \text{in}(S)} y_e \geq 1$ is guaranteed.

AJM NOTE: I think this might give us a different way to achieve the same thing without looking at every subset of P. More thought required.

[1]This satisfies $S \subseteq P$ as only donor-recipient pairs can be involved in cycles.

1.2 Implementation

It is not possible to directly introduce all of the constraints in Equation (1) to the solver, as there are exponentially many such constraints. Instead, a method is required to detect whether these constraints have been satisfied, which Anderson et al. propose achieving by solving a max-flow min-cut problem on an augmented graph.[2]

To form the augmented (weighted, directed) graph, an extra node s (a 'source') is added to the vertex set V and an edge connects s to every altruistic donor $n \in N$. Weights are associated with all the edges in this augmented graph. In particular, every new edge (s, n) gets a weight of 1, while the remaining edges are weighted by y_e.

In this way, the edges involved in short cycles are weighted 1, while the edges of chains and those of the potentially lurking long cycles are weighted 1.

If for any $v \in P$, a cut can be found to separate s from v which has weight less than f_v^i, then we have found a long cycle. Notice that we only actually have to check those v for which $f_v^i > 0$.

This approach works because every chain starts at an $n \in N$ and s was connected to every $n \in N$ and all the edges in these 'superchains' have weight 1 while the edges of long cycles are also weighted 1 but they are disconnected from s.

[2]AJM NOTE: This might actually only be explained in the PNAS paper, and in that case only really explained in the Supplemental Material - I haven't properly compared with the INFORMS paper just yet.