FINDING LONG CHAINS IN KIDNEY EXCHANGE USING THE TRAVELING SALESMAN PROBLEM

Ross Anderson, Itai Ashlagi, David Gamarnik
and Alvin E. Roth
(Presented by Alireza Farhadi)

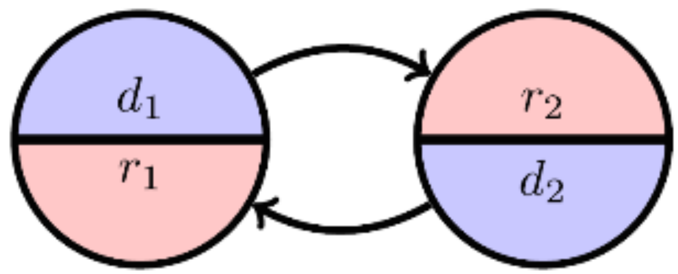
Kidney Exchange Problem

- Given donors and patients, maximize the total number of transplants performed

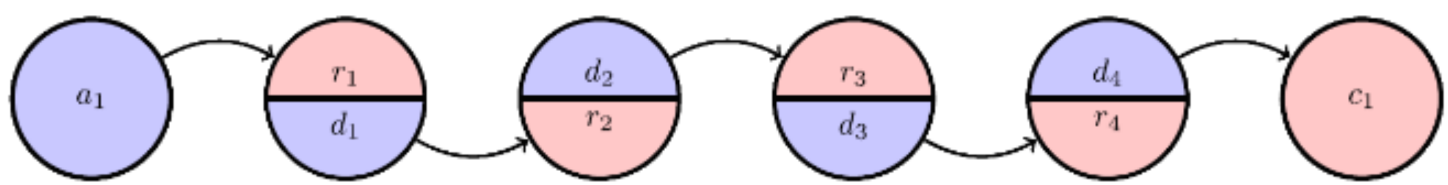
- Two types of donors
  - Altruistic donors: willing to donate their kidney without asking for anything
  - Patient-donor pairs

Kidney Exchange Problem

- Cycles



- Chains



Kidney Exchange Problem

- To exchange cycles, all the surgeries must be performed simultaneously
  - Very complicated for the large cycles

- Each cycle can have at most k nodes

Formal Definition

- $G = (V, E; w)$: a directed weighted graph
- k : maximum length of cycles
- N : altruistic donors
- P : patient-donor pairs
  - $V = N \cup P$

Recursive IP Formulation

$$
\begin{aligned}
\max \quad & \sum_{e \in E} w_e y_e \\
\text{s.t.} \quad & \sum_{e \in \delta^-(v)} y_e = f_v^i \quad v \in V && \textbf{[1]} \\
& \sum_{e \in \delta^+(v)} y_e = f_v^o \quad v \in V \\
& f_v^o \le f_v^i \le 1 \quad v \in P, && \textbf{[2]} \\
& f_v^o \le 1 \quad v \in N, && \textbf{[3]} \\
& \sum_{e \in C} y_e \le |C| - 1 \quad C \in \mathcal{C} \setminus \mathcal{C}_k, && \textbf{[4]} \\
& y_e \in \{0, 1\} \quad e \in E. && \textbf{[5]}
\end{aligned}
$$

- y_e : decision variable for each edge
- f_v^i : in-flow of vertex v
- f_v^o : out-flow of vertex v
- C : set of all cycles
- C_k : all the cycles with the length at most k

Recursive IP Formulation

- Number of constraints is exponential in $|E|$

$$\sum_{e \in C} y_e \leq |C| - 1 \quad C \in \mathcal{C} \setminus \mathcal{C}_k,$$ [4]

- Eliminate all the constraints for the cycles
  - Add violated constraints and resolve the IP

TSP-Based IP

- Using different decision variables for cycles and chains
- Finding chains is similar to prize-collecting TSP.
  - PC-TSP: visit each city (patient-donor pair) exactly once, but with the additional option to pay some penalty to skip a city (penalized for leaving pairs unmatched)

- We add a decision variable for every cycle of length at most k

TSP-Based IP

$$
\begin{aligned}
\max \quad & \sum_{e \in E} w_e y_e + \sum_{C \in \mathcal{C}_k} w_C z_C \\
\text{s.t.} \quad & \sum_{e \in \delta^-(v)} y_e = f_v^i \quad v \in V \\
& \sum_{e \in \delta^+(v)} y_e = f_v^o \quad v \in V \\
f_v^o + & \sum_{C \in \mathcal{C}_k(v)} z_C \leq f_v^i + \sum_{C \in \mathcal{C}_k(v)} z_C \leq 1 \quad v \in P,
\end{aligned}
\qquad [6]
$$

$$
\begin{aligned}
& f_v^o \leq 1 \quad v \in N, \\
& \sum_{e \in \delta^-(S)} y_e \geq f_v^i \quad S \subseteq P, \ v \in S \\
& y_e \in \{0,1\} \quad e \in E, \\
& z_C \in \{0,1\} \quad C \in \mathcal{C}_k.
\end{aligned}
\qquad [7]
$$

- z_C : decision variable for the cycles of length at most k

TSP-Based IP

- Every chain begins in N (altruistic donors)

- For every vertex v, and a partitioning of vertices into sets $S, V\backslash S$ such that $v \in S$ and $N \subseteq V/S$
  - In-flow of S must be at least f_v^i

$$\sum_{e \in \delta^-(S)} y_e \geq f_v^i \quad S \subseteq P, \ v \in S$$

[7]

- Number of constraints is exponential in $|E|$

Finding violated Constraints

- Violated constraints can be found using the max flow-min cut problem.

- We add a source vertex s to our graph and connect it to the vertices in N to get a new graph $\bar{G} = (\bar{V}, \bar{E}; \bar{w})$
  - $\bar{V} = V \cup \{s\}$
  - $\bar{E} = E \cup \{(s, n) \mid n \in N\}$
  - $\bar{w} = \begin{cases} y_e & e \in E \\ 1 & otherwise \end{cases}$
- For every vertex v, we solve the max flow-min cut problem with the source s and the sink v

Performance of TSP-IP

			Running time, s	
NDDs	Patient–donor pairs	Edges	Recursive	TSP
3	202	4,706	0.148	0.031
10	156	1,109	13.093	0.022
6	263	8,939	59.158	1.655
5	284	10,126	71.066	0.807
6	324	13,175	418.27	0.981
6	328	13,711	474.947	1.947
6	312	13,045	1,200*	0.157*
10	152	1,125	48.56	0.054
3	269	2,642	40.506	0.134
10	257	2,461	67.783	0.258
7	255	2,390	85.475	0.268
6	215	6,145	248.46	0.532
10	255	2,550	216.48	0.126
1	310	4,463	721.66	0.555
11	257	2,502	1,039.105	0.125
6	261	8,915	1,200	4.435
10	256	2,411	587.238	0.114
6	330	13,399	1,200	1.621
10	256	2,347	1,200	0.305
7	291	3,771	1,200*	0.221
8	275	3,158	1,200*	0.224
4	289	3,499	1,200*	0.2
3	199	2,581	1,200*	0.041
7	198	4,882	1,200*	8.204
2	389	8,346	1,200*	0.096

Timeouts (optimal solution not found) are indicated by an asterisk.

Thank You