Kidney Exchange
With an emphasis on computation & work from CMU

John P. Dickerson
(in lieu of Ariel Procaccia)
Today’s lecture: kidney exchange

Al Roth
Tayfun Sönmez
Utku Ünver

Hmm ...
Hmm ...
Hmm ...

Next lecture!
This talk

- Motivation – sourcing organs for needy patients
- Computational dimensions of organ exchange
  - Dimension #1: Post-match failure
  - Dimension #2: Egalitarianism
  - Dimension #3: Dynamism
- FutureMatch framework
  - Preliminary results from CMU on real data
- Take-home message & future research

*This is a fairly CMU-centric lecture because some of it is on my thesis work, but I am happy to talk about anything related to kidney exchange!*
High-Level Motivation

Organ Failure

Kidney Failure
Kidney transplantation

- US waitlist: over 100,000
  - 36,157 added in 2014
- 4,537 people died while waiting
- 11,559 people received a kidney from the deceased donor waitlist
- 5,283 people received a kidney from a living donor
  - Some through kidney exchanges! [Roth et al. 2004]
  - Our software runs UNOS national kidney exchange

[Graph showing supply and demand of kidney transplants from 1988 to 2013]
Kidney exchange

Donors

Wife

Brother

Husband

Patient

D1 P1

D2 P2

(2- and 3-cycles, all surgeries performed simultaneously)
Non-directed donors & chains

• Not executed simultaneously, so no length cap required based on logistic concerns ...
• ... but in practice edges fail, so some finite cap is used!
Fielded exchanges around the world

- NEPKE (started 2003/2004, now closed)
- United Network for Organ Sharing (UNOS)
  - US-wide, 140+ transplant centers
  - Went live Oct. 2010, conducts biweekly matches
- Alliance for Paired Donation
- Paired Donation Network (now closed)
- National Kidney Registry (NKR)
- San Antonio
- Canada
- Netherlands
- England
- Portugal (just started!)
- Israel (about to start)
- Others ...?

(Around 1000 transplants in US, driven by chains!)

(Current as of late 2014)
Clearing problem

• $k$-cycle ($k$-chain): a cycle (chain) over $k$ vertices in the graph such that each candidate obtains the organ of the neighboring donor

• The clearing problem is to find the “best” disjoint collection consisting of cycles of length at most $L$, and chains
  – Typically, $2 \leq L \leq 5$ for kidneys (e.g., $L=3$ at UNOS)
Hardness & formulation

“Best” = maximum cardinality

- $L=2$: polynomial time
- $L>2$: NP-complete [Abraham, Blum, Sandholm 2007]
  - Significant gains from using $L>2$

- State of the art (national kidney exchange):
  - $L=3$
  - Formulate as MIP, one decision variable per cycle
  - Specialized branch-and-price can scale to 10,000 patient-donor pairs (cycles only) [Abraham, Blum, Sandholm 2007]
  - Harder in practice (+chains)
Basic IP formulation #1

“Best” = maximum cardinality

• Binary variable $x_{ij}$ for each edge from $i$ to $j$

Maximize

$$u(M) = \sum w_{ij} x_{ij}$$

Subject to

$$\sum_j x_{ij} = \sum_j x_{ji}$$

for each vertex $i$

$$\sum_{j} x_{ij} \leq 1$$

for each vertex $i$

$$\sum_{1 \leq k \leq L} x_{i(k)i(k+1)} \leq L-1$$

for paths $i(1)\ldots i(L+1)$

(no path of length $L$ that doesn’t end where it started – cycle cap)
Best Edge Formulation

“If: flow into $v$ from a chain
Then: at least as much flow across cuts from \{A\}"

Anderson et al. 15

$\text{"Best" = maximum cardinality}$
Basic IP formulation #2

“Best” = maximum cardinality

• Binary variable $x_c$ for each cycle/chain $c$ of length at most $L$

Maximize

$$\sum |c| x_c$$

Subject to

$$\sum_{c: i \in c} x_c \leq 1 \quad \text{for each vertex } i$$
Solving big integer programs

• Too big to write down full model
• Branch-and-price [Barnhart et al. 1998] stores reduced model, incrementally brings columns in via pricing:
  • Positive price $\rightarrow$ constraint in full model violated
  • No positive price variables $\rightarrow$ $OPT_{\text{reduced}} = OPT_{\text{full}}$
• Old pricing [Abraham et al. 07]:
  • DFS in compatibility graph, exponential in chain cap
• New pricing [Glorie et al. 14]:
  • Modified Bellman-Ford in reduced compatibility graph
  • Polynomial in graph size!
  • But not correct
The Right Idea

• Idea: solve structured optimization problem that implicitly prices variables

• Price: \( w_c - \sum_{v \in c} \delta_v = \sum_{e \in c} w_e - \sum_{v \in c} \delta_v = \sum_{(u,v) \in c} [w_{(u,v)} - \delta_v] \)

• Take \( G \), create \( G' \) s.t. all edges \( e = (u,v) \) are reweighted \( r_{(u,v)} = \delta_v - w_{(u,v)} \)
  – Positive price cycles in \( G \) = negative weight cycles in \( G' \)

• Bellman-Ford finds shortest paths
  – Undefined in graphs with negative weight
  – Adapt B-F to prevent internal looping during the traversal
    • Shortest path is NP-hard (reduce from Hamiltonian path:
      – Set edge weights to -1, given edge \((u,v)\) in \( E \), ask if shortest path from \( u \) to \( v \) is weight \( 1-|V| \) \( \rightarrow \) visits each vertex exactly once
    • We only need some short path (or proof that no negative cycle exists)
  – Now pricing runs in time \( O(|V||E|\text{cap}^2) \)
Note: Anderson et al.’s algorithm (CG-TSP) is very strong for uncapped aka “infinite-length” chains, but a chain cap is often imposed in practice.
Comparison
“Best” = maximum cardinality

- IP #1 is the most basic **edge formulation**
- IP #2 is the most basic **cycle formulation**
- Tradeoffs in number of variables, constraints
  - IP #1: $O(|E|^L)$ constraints vs. $O(|V|)$ for IP #2
  - IP #1: $O(|V|^2)$ variables vs. $O(|V|^L)$ for IP #2
- IP #2’s relaxation is weakly tighter than #1’s.
  Quick intuition in one direction:
  - Take a length L+1 cycle. #2’s LP relaxation is 0.
  - #1’s LP relaxation is $(L+1)/2 - \frac{1}{2}$ on each edge
The big problem

• What is “best”?
  – Maximize matches right now or over time?
  – Maximize transplants or matches?
  – Prioritization schemes (i.e. fairness)?
  – Modeling choices?
  – Incentives? Ethics? Legality?

• Optimization can handle this, but may be inflexible in hard-to-understand ways

Want humans in the loop at a high level (and then CS/Opt handles the implementation)
Dimension #1: Post-Match Failure
Matched ≠ Transplanted

• Only around 8% of UNOS matches resulted in an actual transplant
  – Similarly low % in other exchanges [ATC 2013]

• Many reasons for this. How to handle?

• One way: encode probability of transplantation rather than just feasibility
  – for individuals, cycles, chains, and full matchings
Failure-aware model

- Compatibility graph $G$
  - Edge $(v_i, v_j)$ if $v_i$’s donor can donate to $v_j$’s patient
  - Weight $w_e$ on each edge $e$
- Success probability $q_e$ for each edge $e$

- Discounted utility of cycle $c$
  
  $$u(c) = \sum w_e \cdot \prod q_e$$

Value of successful cycle

Probability of success
Failure-aware model

• Discounted utility of a $k$-chain $c$

\[
u(c) = \left[ \sum_{i=1}^{k-1} (1 - q_i)^i \prod_{j=0}^{i-1} q_j \right] + \left[ k \prod_{i=0}^{k-1} q_i \right]
\]

Exactly first $i$ transplants

Chain executes in entirety

• Cannot simply “reweight by failure probability”

• Utility of a match $M$: $u(M) = \sum u(c)$
Our problem

• *Discounted clearing problem* is to find matching $M^*$ with highest discounted utility
Theoretical result #1
• $G(n, t(n), p)$: random graph with
  – $n$ patient-donor pairs
  – $t(n)$ altruistic donors
  – Probability $\Theta(1/n)$ of incoming edges
• Constant transplant success probability $q$

Theorem

For all $q \in (0,1)$ and $\alpha, \theta > 0$, given a large $G(n, \alpha n, \theta/n)$, w.h.p. there exists some matching $M'$ s.t. for every maximum cardinality matching $M$, $u_q(M') \geq u_q(M) + \Omega(n)$
Brief intuition: Counting Y-gadgets

- For every structure $X$ of constant size, w.h.p. can find $\Omega(n)$ structures isomorphic to $X$ and isolated from the rest of the graph.
- Label them (alt vs. pair): flip weighted coins, constant fraction are labeled correctly $\rightarrow$ constant $\times \Omega(n) = \Omega(n)$
- Direct the edges: flip 50/50 coins, constant fraction are entirely directed correctly $\rightarrow$ constant $\times \Omega(n) = \Omega(n)$
In theory, we’re losing out on *expected actual transplants* by maximizing match cardinality.

... What about in practice?
Solving this new problem

- Real-world kidney exchanges are still small
  - UNOS pool: 281 donors, 260 patients [2 Feb 2015]
- Undiscounted clearing problem is NP-hard when cycle/chain cap $L \geq 3$ [Abraham et al. 2007]
  - Special case of our problem
- The current UNOS solver will not scale to the projected nationwide steady-state of 10,000
  - Empirical intractability driven by chains
We can’t use the current solver

• Branch-and-bound IP solvers use upper and lower bounds to prune subtrees during search
• Upper bound: cycle cover with no length cap
  – PTIME through max weighted perfect matching

Proposition:

The unrestricted discounted maximum cycle cover problem is NP-hard.

(Reduction from 3D-Matching)
Incrementally solving very large IPs

• #Decision variables grows linearly with #cycles and #chains in the pool
  – Millions, billions of variables
  – Too large to fit in memory
• Branch-and-price incrementally brings variables into a reduced model [Barnhart et al. 1998]
• Solves the “pricing problem” — each variable gets a real-valued price
  – Positive price $\rightarrow$ resp. constraint in full model violated
  – No positive price cycles $\rightarrow$ optimality at this node
Theorem:
Given a chain \( c \), any extension \( c' \) will not be needed in an optimal solution if the infinite extension has non-positive value.

\[
\left( \frac{q_{\max}}{1 - q_{\max}} \prod_{i=0}^{k-1} q_i \right) + u(c) + \ell - \left( d_{\min} + \sum_{i=0}^{k} d_i \right) \leq 0
\]
Scaling experiments

<table>
<thead>
<tr>
<th></th>
<th>V</th>
<th></th>
<th>CPLEX</th>
<th>Ours</th>
<th>Ours without chain curtailing</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>127 / 128</td>
<td>128 / 128</td>
<td>128 / 128</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>125 / 128</td>
<td>128 / 128</td>
<td>128 / 128</td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>105 / 128</td>
<td>128 / 128</td>
<td>125 / 128</td>
<td></td>
<td></td>
</tr>
<tr>
<td>75</td>
<td>91 / 128</td>
<td>126 / 128</td>
<td>123 / 128</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>1 / 128</td>
<td>121 / 128</td>
<td>121 / 128</td>
<td></td>
<td></td>
</tr>
<tr>
<td>150</td>
<td>114 / 128</td>
<td>95 / 128</td>
<td>95 / 128</td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>113 / 128</td>
<td>76 / 128</td>
<td>76 / 128</td>
<td></td>
<td></td>
</tr>
<tr>
<td>250</td>
<td>94 / 128</td>
<td>48 / 128</td>
<td>48 / 128</td>
<td></td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>107 / 128</td>
<td>1 / 128</td>
<td>1 / 128</td>
<td></td>
<td></td>
</tr>
<tr>
<td>700</td>
<td>115 / 128</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>900</td>
<td></td>
<td>38 / 128</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Runtime limited to 60 minutes; each instance given 8GB of RAM.
- |V| represents #patient-donor pairs; additionally, 0.1|V| altruistic donors are present.
In theory and practice, we’re helping the *global* bottom line by considering post-match failure ...  

... But can this hurt some *individuals*?
Dimension #2: Egalitarianism
Sensitization at UNOS

- Highly-sensitized patients: unlikely to be compatible with a random donor
- Deceased donor waitlist: 17%
- Kidney exchanges: much higher (60%+)

“Easy to match” patients

“Hard to match” patients
Price of fairness

• Efficiency vs. fairness:
  – *Utilitarian* objectives may favor certain classes at the expense of marginalizing others
  – *Fair* objectives may sacrifice efficiency in the name of egalitarianism

• **Price of fairness**: relative system efficiency loss under a fair allocation
  [Bertismas, Farias, Trichakis 2011]
  [Caragiannis et al. 2009]
Price of fairness in kidney exchange

- **Recall**: want a matching $M^*$ that maximizes utility function $u : \mathcal{M} \rightarrow \mathbb{R}$

  $$M^* = \arg\max_{M \in \mathcal{M}} u(M)$$

- **Price of fairness**: relative loss of match efficiency due to fair utility function $u_f$

  $$\text{POF}(\mathcal{M})(u_f) = \frac{u(M^*) - u(M_f^*)}{u(M^*)}$$
Theoretical result #2
Under the “most stringent” fairness rule:

\[ u_{H \succ L}(M) = \begin{cases} u(M) & \text{if } |M_H| = \max_{M' \in \mathcal{M}} |M'_H| \\ 0 & \text{otherwise} \end{cases} \]

**Theorem**

Assume “reasonable” level of sensitization and “reasonable” distribution of blood types. Then, almost surely as \( n \to \infty \),

\[ \text{POF}(\mathcal{M}, u_{H \succ L}) \leq \frac{2}{33}. \]

(And this is achieved using cycles of length at most 3.)
Linear efficiency loss

\[ \bar{p} \mu_{A\beta} \mu_{O} \]

Sublinear loss

\[ o(n) \]
From theory to practice

• Price of fairness is **low** in theory
• Fairness criterion: *extremely* strict.
• Theoretical assumptions (standard):
  – Big graphs ("n \(\to\) \(\infty\))
  – Dense graphs
  – Cycles (no chains)
  – No post-match failures
  – Simplified patient-donor features

What about the price of fairness *in practice*?
Toward usable fairness rules

• In healthcare, important to work within (or near to) the constraints of the fielded system
  – [Bertsimas, Farias, Trichakis 2013]
  – Our experience with UNOS

• We now present two (simple, intuitive) rules:
  – **Lexicographic**: strict ordering over vertex types
  – **Weighted**: implementation of “priority points”
Lexicographic fairness

Find the best match that includes at least $\alpha$ fraction of highly-sensitized patients.

• *Matching-wide* constraint:
  – Present-day branch-and-price IP solvers rely on an “easy” way to solve the pricing problem
  – Lexicographic constraints $\rightarrow$ pricing problem requires an IP solve, too!

• Strong guarantee on match composition ...
  – ... but harder to predict effect on efficiency
Weighted fairness

Value matching a highly-sensitized patient at (1+β) that of a lowly-sensitized patient, β>0

• Re-weighting is a preprocess → works with all present-day kidney exchange solvers

• Difficult to find a “good” β?
  – Empirical exploration helps strike a balance
Theory vs. “Practice”

*Lexicographic fairness*
Price of fairness: Generated data

<table>
<thead>
<tr>
<th>Size</th>
<th>Saidman (US)</th>
<th>Saidman (UNOS)</th>
<th>Heterogeneous</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.24% (1.98%)</td>
<td>0.00% (0.00%)</td>
<td>0.98% (5.27%)</td>
</tr>
<tr>
<td>25</td>
<td>0.58% (1.90%)</td>
<td>0.19% (1.75%)</td>
<td>0.00% (0.00%)</td>
</tr>
<tr>
<td>50</td>
<td>1.18% (2.34%)</td>
<td>1.96% (6.69%)</td>
<td>0.00% (0.00%)</td>
</tr>
<tr>
<td>100</td>
<td>1.46% (1.80%)</td>
<td>1.66% (3.64%)</td>
<td>0.00% (0.00%)</td>
</tr>
<tr>
<td>150</td>
<td>1.20% (1.86%)</td>
<td>2.04% (2.51%)</td>
<td>0.00% (0.00%)</td>
</tr>
<tr>
<td>200</td>
<td>1.43% (2.08%)</td>
<td>1.55% (1.79%)</td>
<td>0.00% (0.00%)</td>
</tr>
<tr>
<td>250</td>
<td>0.80% (1.24%)</td>
<td>1.86% (1.63%)</td>
<td>0.00% (0.00%)</td>
</tr>
<tr>
<td>500</td>
<td>0.72% (0.74%)</td>
<td>1.67% (0.82%)</td>
<td>0.00% (0.00%)</td>
</tr>
</tbody>
</table>

- Average (st.dev.) % loss in efficiency for three families of random graphs, under the strict lexicographic rule.
- **Good**: aligns with the theory
- **Bad**: standard generated models aren’t realistic
Real UNOS runs

Lexicographic fairness, varying failure rates
Real UNOS runs

Weighted fairness, varying failure rates
Contradictory goals

• Earlier, we saw **failure-aware** matching results in tremendous gains in #expected transplants
• Gain comes at a price – may further marginalize hard-to-match patients because:
  – Highly-sensitized patients tend to be matched in chains
  – Highly-sensitized patients may have higher failure rates (in APD data, not in UNOS data)
UNOS runs, weighted fairness, constant probability of failure (x-axis), increase in expected transplants over deterministic matching (y-axis)
Generated UNOS runs, weighted fairness, constant probability of failure (x-axis), increase in expected transplants over deterministic matching (y-axis)
Generated (top row) and real (bottom row) UNOS runs, weighted fairness (x-axis), bimodal failure probability (APD failures in left column, UNOS failures in right column), increase in expected transplants over deterministic matching (y-axis).
Fairness vs. efficiency can be balanced in theory and in practice *in a static model* ...

... But how should we match *over time*?
Dimension #3: Dynamism
Dynamic kidney exchange

• Kidney exchange is a naturally dynamic event
• Can be described by the evolution of its graph:
  – Additions, removals of edges and vertices

<table>
<thead>
<tr>
<th>Vertex Removal</th>
<th>Edge Removal</th>
<th>Vertex/Edge Add</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transplant, this exchange</td>
<td>Matched, positive crossmatch</td>
<td>Normal entrance</td>
</tr>
<tr>
<td>Transplant, deceased donor waitlist</td>
<td>Matched, candidate refuses donor</td>
<td></td>
</tr>
<tr>
<td>Transplant, other exchange (&quot;sniped&quot;)</td>
<td>Matched, donor refuses candidate</td>
<td></td>
</tr>
<tr>
<td>Death or illness</td>
<td>Pregnancy, sickness changes HLA</td>
<td></td>
</tr>
<tr>
<td>Altruist runs out of patience</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bridge donor reneges</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Our dynamic model
Dynamic matching via potentials

- Full optimization problem is very difficult
  - Realistic theory is too complex
  - Trajectory-based methods do not scale

- Approximation idea:
  - Associate with each “element type” its potential to help objective in the future
  - (Must learn these potentials)
  - Combine potentials with edge weights, perform myopic maximum utility matching
What’s a potential?

• Given a set of features $\Theta$ representing structural elements (e.g., vertex, edge, subgraph type) of a problem:
  – The potential $P_\vartheta$ for a type $\vartheta$ quantifies the future usefulness of that element

• E.g., let $\Theta = \{O-O, O-A, ..., AB-AB, \bullet-O, ..., \bullet-AB\}$
  – 16 patient-donor types, 4 altruist types
  – O-donors better than A-donors, so: $P_{\cdot O} > P_{\cdot A}$
Using potentials to inform myopia

- Using heavy one-time computation to learn potential of each type $\vartheta$
- Adjust solver to take them into account at runtime

- E.g., $P_{\cdot-O} = 2.1$ and $P_{O-AB} = 0.1$
  - Edges between O-altruist and O-AB pair has weight: $1 - 0.5(2.1 + 0.1) = -0.1$
  - Chain must be long enough to offset negative weight
Potentials: simple example

- Potentials assigned only on whether or not a vertex is an altruist
- Two time periods
Expressiveness tradeoff

• In kidney exchange:
  – 20 vertex types
  – 244 edge types (208 cyclic edges, 36 chain edges)
  – 1000s of 3-cycle types, et cetera.

• Allowing larger structural elements:
  – increases expressive power of potentials
  – increases size of hypothesis space to explore

Expressiveness Theory
Vertex vs. Edge: lose at least 1/3
Edge vs. Cycle: lose at least ½
Cycle vs. Graph: lose at least (L-1)/L

Is it that bad in practice?
Simulation results

Vertex potentials
Weighted myopic % improvement (relative to optimal)
We can learn to maximize a utility function over time (negative theory, positive experiments) ...

... But how should we choose an objective?
FutureMatch

A framework for learning to match in dynamic environments

[Dickerson Sandholm AAAI-2015]
Balancing failure and fairness

• Saw that we can strike a balance realizing gains of both matching methods
• Highly dependent on distribution of graphs
• Useful empirical visualization tool for policymakers needing to, e.g., define “acceptable” price of fairness

What about fairness-aware, failure-aware, dynamic matching?
FutureMatch: Learning to match in dynamic environments

**Offline (run once or periodically)**
1. Domain expert describes overall goal
2. Take historical data and policy input to learn a weight function $w$ for match quality
3. Take historical data and create a graph generator with edge weights set by $w$
4. Using this generator and a realistic exchange simulator, learn potentials for graph elements as a function of the exchange dynamics

**Online (run every match)**
1. Combine $w$ and potentials to form new edge weights on real input graphs
2. Solve maximum weighted matching and return match
Example objective: MaxLife

• Maximize the aggregate length of time donor organs will last in patients ...
  - ... with fairness “nobs”, failure-awareness, etc.

• Learn survival rates from all living donations in US since 1987 (~75k trans.)
• Translate to edge weight
• Learn potentials, then combine into new weights
The details are in the paper, but ...

- We show it is possible to:
  - Increase overall #transplants a lot at a (much) smaller decrease in #marginalized transplants
  - Increase #marginalized transplants a lot at no or very low decrease in overall #transplants
  - Increase both #transplants and #marginalized

- Again, sweet spot depends on distribution:
  - Luckily, we can generate – and learn from – realistic families of graphs!
Take-home message

• Contradictory wants in kidney exchange!

• In practice, can (automatically) strike a balance between these wants
  – Keeps the human in the loop

• Some improvements (e.g., failure-awareness) are *unilaterally good*, given the right balance with other wants
Lots left to do!

• Fairness:
  – Theoretical guarantees in better models
  – More general definitions

• Modeling:
  – More accurate models (multiple exchanges, legality, more features on patient/donor)

• Dynamics:
  – Better optimization methods
  – Faster “means vs. ends” loop with humans
Moving beyond kidneys

- **Chains are great!** [Anderson et al. 2015, Ashlagi et al. 2014, Rees et al. 2009]
- **Kidney transplants are “easy” and popular:**
  - Many altruistic donors
- **Liver transplants: higher mortality, morbidity:**
  - (Essentially) no altruistic donors

![Kidney transplantation diagram]

[Dickerson Sandholm AAAI-2014]
Would this help?

- **Theory:** adapted Erdős-Rényi models
- **Dense model** [Saidman et al. 2006]
  - Constant probability of edge existing
- **Sparse model** [Ashlagi et al. 2012]
  - $1-\lambda$ fraction is *highly-sensitized* ($p_H = c/n$)
  - $\lambda$ fraction is *lowly-sensitized* ($p_L > 0$, constant)
- **Not all kidney donors want to give livers**
  - Constant probability $p_{K\rightarrow L} > 0$
Sparse graph, many altruists

- $n_K$ kidney pairs in graph $D_K$
- $n_L = \gamma n_K$ liver pairs in graph $D_L$
- Number of altruists $t(n_K)$
- Constant cycle cap $z$

**Theorem**

Assume $t(n_K) = \beta n_K$ for some constant $\beta > 0$. Then, with probability 1 as $n_K \to \infty$,

Any efficient matching on $D = \text{join}(D_K, D_L)$ matches $\Omega(n_K)$ more pairs than the aggregate of efficient matchings on $D_K$ and $D_L$.

*Building on [Ashlagi et al. 2012]*
Intuition

• Find a linear number of “good cycles” in $D_L$ that are length $> z$
  – Good cycles = isolated path in highly-sensitized portion of pool and exactly one node in low portion
• Extend chains from $D_K$ into the isolated paths (aka can’t be matched otherwise) in $D_L'$ of which there are linearly many
  – Have to worry about $p_{K \to L'}$, and compatibility between vertices
• Show that a subset of the dotted edges below results in a linear-in-number-of-altruists max matching
  – $\Rightarrow$ linear number of $D_K$ chains extended into $D_L$
  – $\Rightarrow$ linear number of previously unmatched $D_L$ vertices matched
Sparse graph, few altruists

- \( n_K \) kidney pairs in graph \( D_K \)
- \( n_L = \gamma n_K \) liver pairs in graph \( D_L \)
- Number of altruists \( t \) – no longer depends on \( n_K \)!
- \( \lambda \) is frac. lowly-sensitized
- Constant cycle cap \( z \)

**Theorem**

Assume constant \( t \). Then there exists \( \lambda' > 0 \) s.t. for all \( \lambda < \lambda' \)

Any efficient matching on \( D = \text{join}(D_K, D_L) \) matches \( \Omega(n_K) \) more pairs than the aggregate of efficient matchings on \( D_K \) and \( D_L \).

With constant positive probability.  

*Building on [Ashlagi et al. 2012]*
Intuition

• For large enough $\lambda$ (i.e., lots of sensitized patients), there exist pairs in $D_K$ that can’t be matched in short cycles, thus only in chains
  – Same deal with $D_L$, except there are no chains
• Connect a long chain (+altruist) in $D_K$ into an unmatchable long chain in $D_L$, such that a linear number of $D_L$ pairs are now matched
FutureMatch + multi-organ exchange?

• Combination results in
  – Linear gain in theory
  – Big gains in simulation

• Equity problems
  – Kidneys ≠ livers
  – Hard to quantify cross-organ risk vs. reward

Let FutureMatch sort it out?

• 16.8% increase in total matches, combined pool vs. independent pools

• Independent samples $t$-test reveals statistical significance:
  • $T(46) = 31.37$, $p < 0.0001$

Also: lung exchange!
[Ergin Sönmez Ünver 2015]
Questions?

Pubs:  jpdickerson.com/pubs/dickerson15futurematch.pdf
       jpdickerson.com/pubs.html

Code:  github.com/JohnDickerson/KidneyExchange

Very incomplete list of CMU folks working on kidney exchange/matching:
{ Avrim Blum, John Dickerson, Alan Frieze, Anupam Gupta, Nika Haghtalab,
  Jamie Morgenstern, Ariel Procaccia, R. Ravi, Tuomas Sandholm }

Thanks to:
Kidney Exchange

Backup Slides
• Efficient matching with cycles and chains of length at most 3 in a dense kidney exchange ABO model [Dickerson Procaccia Sandholm AAMAS-2012]
Simulating dynamic kidney exchange (two time periods)
Generated UNOS runs, median number of transplants as $|V|$ increases (x-axis) for each of the objective functions.
Price of fairness: UNOS data

<table>
<thead>
<tr>
<th>Metric</th>
<th>Minimum</th>
<th>Average</th>
<th>Maximum</th>
<th>St. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loss % (Objective)</td>
<td>0.00%</td>
<td>2.76%</td>
<td>19.04%</td>
<td>4.84%</td>
</tr>
<tr>
<td>Loss % (Cardinality)</td>
<td>0.00%</td>
<td>4.09%</td>
<td>33.33%</td>
<td>8.18%</td>
</tr>
<tr>
<td>Loss (Cardinality)</td>
<td>0</td>
<td>0.55</td>
<td>4</td>
<td>1.1</td>
</tr>
</tbody>
</table>

- Minimum, average, and maximum loss in objective value and match size due to the strict lexicographic fairness rule, across the first 73 UNOS match runs, in a deterministic model.
Acknowledgments

• This material was funded by NSF grants IIS-1320620, CCF-1101668, CCF-1215883, and IIS-0964579, by an NDSEG fellowship, and used the Pittsburgh Supercomputing Center in partnership with the XSEDE, which is supported by NSF grant OCI-1053575. We thank Intel Corporation for machine gifts.

• Duke CPS 196.2 (Conitzer)