Organ Exchanges:
A Fielded Testbed for AI & Healthcare

John P. Dickerson

Tuomas Sandholm

Tutorial outline

- Introduction & preliminaries
- Optimization models & state of the practice
- Deep dive into dimensions of kidney exchange:
  - Short-term uncertainty
  - Fairness vs economic efficiency
  - Long-term uncertainty & dynamic optimization
  - Incorporating human expert judgment in better ways
  - Incentives & mechanism design
- Other organ exchanges
- Conclusion & open research problems

Intermission

Tutorial outline

- **Introduction & preliminaries**
- Optimization models & state of the practice
- Deep dive into dimensions of kidney exchange:
  - Short-term uncertainty
  - Fairness vs economic efficiency
  - Long-term uncertainty & dynamic optimization
  - Incorporating human expert judgment in better ways
  - Incentives & mechanism design
- Other organ exchanges
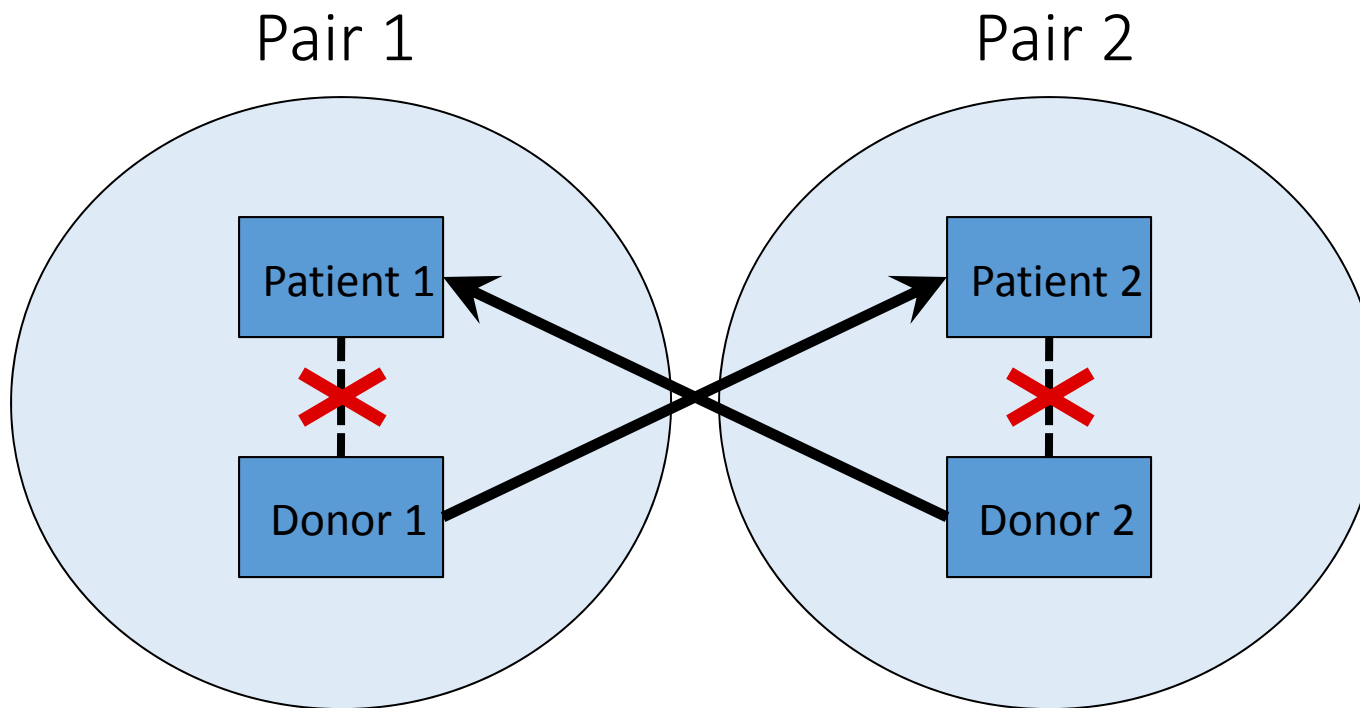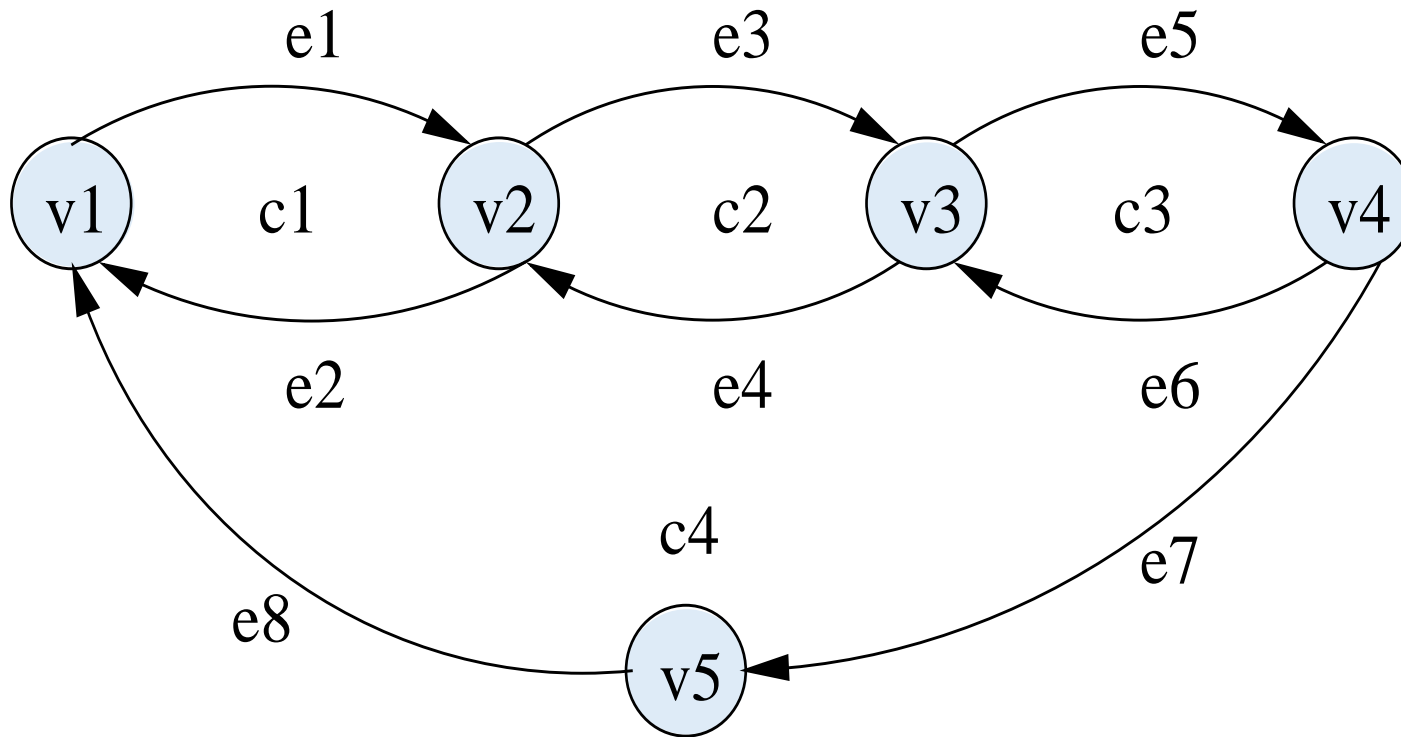- Conclusion & open research problems

GETTY IMAGES

Kidney exchange

Idea introduced in 1986 [Rapaport]
First exchange (NEPKE) started 2003-04 [Roth, Sönmez, Ünver, …]

Objective:

Maximum weight *combination* of disjoint cycles

Cap on cycle length

- Why a cap?
  - Transplants in a cycle must occur simultaneously
  - Cycle may fail

- Cap is typically 3

Complexity of batch optimization problem

- **Theorem** [Abraham, Blum, Sandholm *EC-07*]

  NP-complete for any cycle length cap ≥ 3

  - Solvable in polynomial time if cap=2 or cap=∞

- See also complexity results by Biró, Manlove, Rizzi *Disc. Math. 09*

Other barter-exchange markets

- Holiday Homes: Intervac
- Books: Read It Swap It
- General used goods: Netcycler / swap.com
- National Odd Shoe Exchange
- Room exchange (e.g. dorm rooms)
- Nurse shift exchange

Tutorial outline

- Introduction & preliminaries
- Optimization models & state of the practice
- Deep dive into dimensions of kidney exchange:
    - Short-term uncertainty
    - Fairness vs economic efficiency
    - Long-term uncertainty & dynamic optimization
    - Incorporating human expert judgment in better ways
    - Incentives & mechanism design
- Other organ exchanges
- Conclusion & open research problems

# Pre-2007 state of the art in kidney exchange clearing algorithms

- Manual matching / greedy algorithms

- (Weighted) maximum matching [Edmonds 1965]

- CPLEX

Algorithms for batch problem

- Algorithms find a provably optimal solution
  - Based on branch-and-price framework:
    - Branch-and-price, DFS pricing [Abraham, Blum, Sandholm *EC-07*]
    - Branch-and-price, B-F pricing [Glorie et al. *MSOM 14*, Plaut, Dickerson, Sandholm *AAAI-16*]
    - Branch-and-price, … [Klimentova et al. *ICCSA-14*, Manlove & O'Malley *ACM JEA 14*, Mak-Hau *J. Comb Opt 15*, …]
  - Based on constraint generation:
    - Basic, not scalable [Abraham, Blum, Sandholm *EC-07*]
    - Based on PC-TSP [Anderson et al. *PNAS 2015*]
  - Compact formulations:
    - Extended edge formulation [Constantino et al. *EJOR 2014*]
    - Position-indexed [ongoing work with CMU and Manlove & Trimble at U. Glasgow]

Kidney exchanges use designs, algorithms, and software from Prof. Sandholm's lab

- United Network for Organ Sharing (UNOS)
  - Our technology was selected
  - Exchange went live Oct. 2010
  - Match run twice a week
  - 143 transplant centers

- Previously:
  - Alliance for Paired Donation
  - Paired Donation Network
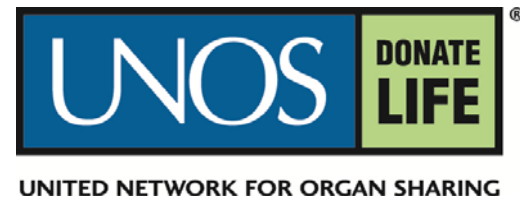
# UNOS has design constraints that private exchange don't have

- Must have transparent, broadly-agreed policies
- Clearing algorithm / priority points must be transparent
- AI must be autonomous (surgeons still have veto)
- Have to work also with small transplant centers => slower turnaround time
- …

A first approach:
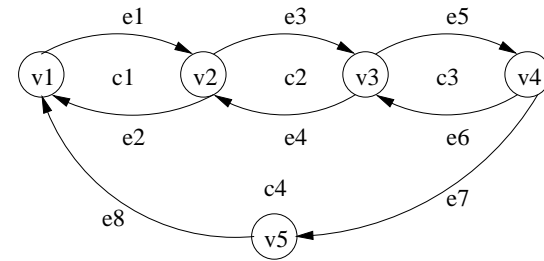Constraint (row) generation

[Abraham, Blum, Sandholm, *EC-07*]

Special Case: No limit on cycle length

- Polynomial-time solvable via max-weight perfect matching

- Red vertex for each patient, blue vertex for each donor

- Edge between each patient and their incompatible donor with weight 0

  (=> perfect matching)

- Edge between patient and compatible donor with weight 1

- Exchanges correspond to perfect matchings
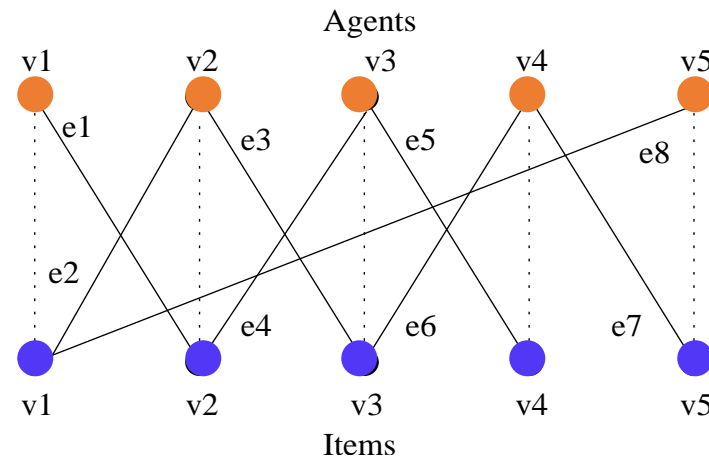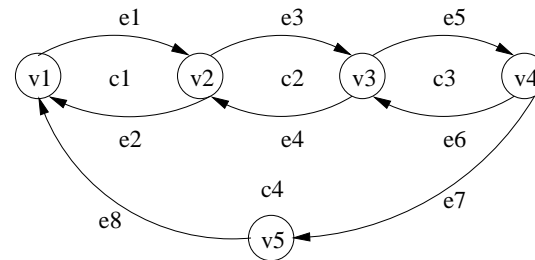
ILP: Edge Formulation (in original directed graph)

- ## Max $\sum_e w(e)*e$

- ## Subject to:
  - Conservation: $\sum_{e \in Out(vi)} e - \sum_{e \in In(vi)} e = 0$
  - Capacity: $\sum_{e \in Out(vi)} e \leq 1$ for all v_i
  - $e \in \{0,1\}$ for each edge e

- ## Limit L on cycle length:
  - For each (non-cycle) path p of length L:
    $$\sum_{e \in p} e \leq L - 1$$

Constraint (= Row) Generation

- ILP too large (too many constraints)
  - Even with only 1000 patients, there are 400M length-3 paths

- Incremental formulation
  - Begin with small subset of constraints
  - Repeat
    - Solve LP relaxation
    - Add subset of violated constraints, if any
  - Perform Branch-and-Bound, doing the repeat loop above at every node

Constraint Generation …

- Constraint Seeding:
  - Forbid any (non-cycle) path of length L-1 that has no edge closing the cycle from its tail to head
  - Or, seed with random constraints from the ILP

- Constraint Generation:
  - Find length-L path with value sum more than $L - 1$
    - A long cycle C contains |C| of these paths
  - Or, more space efficient to add only one constraint for such a cycle: Edge sum in a long fractional cycle can be at most floor(([L-1]/L)*|C|) => slower
  - So we went in the other direction: Add a constraint per violating path p, and each path with the same interior vertices => faster

A First Approach:
Constraint (Row) Generation in Edge Formulation

- Even after many improvements, could not clear markets with 100 nodes faster than our second approach with 10,000 nodes

- **Theorem.** LP relaxation of edge formulation not as tight as that of cycle formulation

A better algorithm (early version of what we fielded at UNOS)

[Abraham, Blum, Sandholm, *EC-07*]

Special Case: Cycle Length L ≤ 2
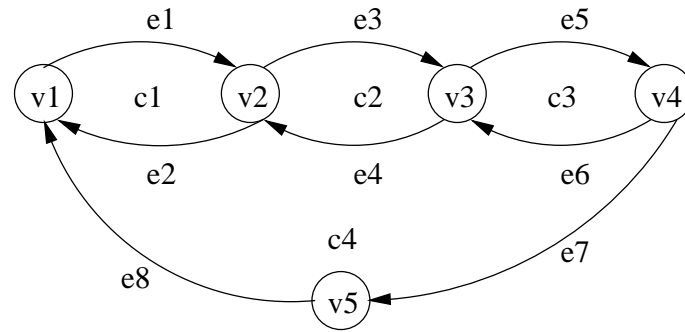
- Model using a different graph:
  - One undirected edge for each cycle of length at most 2
    - Edge weight is weight of cycle
  - Exchanges correspond to matchings
 - Polytime to find max-weight matching in general graph
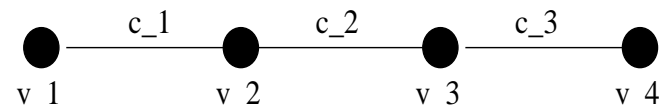
ILP: Cycle formulation

- $C(L) \equiv$ set of cycles of length $\leq L$
- One variable for each cycle c in $C(L)$
- Max $\sum_{c \in C(L)} w_c x_c$
- Subject to:
  - $\sum_{c : v_i \in c} x_c \leq 1$ (each v_i belongs to at most one cycle)
- $x_c \in \{0,1\}$ for all $c \in C(L)$

Solving the ILP

- Too large to write down
  - Unlike winner determination in combinatorial auctions

- Overall approach: Branch-and-Price
  - Branch: select fractional column and fix its value to 1 and 0 respectively
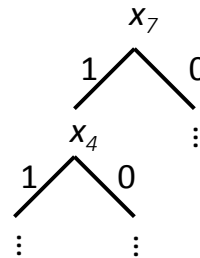
  - Fathom the search node if no better than incumbent
    - Solve LP relaxation using column generation

Column generation

- ***Master*** LP P has too many variables
  - Won't fit in memory
  - Would take too long to solve
- Begin with ***restricted*** LP P', which contains only a small subset of the variables (i.e., cycles)
  - OPT(P') ≤ OPT(P)
- Solve P' and, if necessary, add more variables to it
- Repeat until OPT(P') = OPT(P)

"Pricing" problem

- **Price** of a cycle (i.e., column) c is
  - $p(c) = w_c - \sum_{v \text{ in } c} \text{dual-val}(v)$
- Dual constraint c is violated if $p(c) > 0$
- **Pricing problem**: find a positive price cycle, or report that none exists (in which case OPT(P') = OPT(P))
- Key: Check the price of cycles one-by-one, without having all cycles in memory

Pricing problem:
Further techniques to enhance speed

- Generate cycles by DFS over input graph

- Vertices explored in non-decreasing order of dual value
  - Earlier vertices more likely to belong to positive-price cycle
  - Can prune DFS path early

- Avoid repeating parent's pricing problem work in child's pricing problem
  - If a vertex wasn't the root of any positive price cycle,
    and its dual value hasn't decreased,
    then it can't be the root of a positive-price cycle now either

"Tailing off" effect

- OPT(P') = OPT(P), but some columns still have positive price
  - Many iterations required to prove optimality
- Our technique for tackling this led to big speedup
  - Polytime upper bound by removing cycle length constraint
    - Edge formulation in bipartite perfect-matching graph (integrality gap 0, so we can just solve that LP instead of that ILP)
    - Column (edge here) generation
      - Fastest polynomial-time maximum-weight matching code didn't scale [Rothberg DIMACS implementation challenge 1990]
  - Optimal once incumbent value = upper bound
    - Length-3 cycles usually enough to match upper bound

Column seeding

- Pricing problem is expensive & improving OPT(P') is slow
  - => Want to begin with OPT(P') close to OPT(P)
  - => Select good (but small) set of initial columns

- Begin with columns from heuristic solutions
  - E.g. randomized greedy and max-weight matching
  - Perform well and introduce very few columns

- Also, random selection of larger set of cycles (400,000)

Column management

- Problems with too many columns in P'
  - Run out of memory, even with ≤ 4000 vertices
  - LPs take longer to solve

- Delete columns so P' is not too big
  - Only a small fraction of columns end up in the final solution for OPT(P) => unlikely to delete
    - Will always generate again if needed
  - Delete column with largest negative price first, as this is the most satisfied constraint in D
    - But some columns we never delete, e.g., those we have branched on and those with positive LP value

Primal heuristics

1. Rounding heuristic: include all cycles with LP value at least ½ ; greedily select remaining cycles
   - Rarely helps

2. Our heuristic:
   - P' at root node usually contains enough columns so that integral OPT can match fractional OPT
   - We use CPLEX MIP as a primal heuristic at nodes, *but only on the restricted ILP that corresponds to P'*
     - Constraint that ILP value has to match fractional target, and
     - Time limit, and
     - Only do this if node has sufficiently different set of cycles than its parent
   - Improves speed significantly

Experiment using data generator by [Saidman et al. 06]
Cycle cap = 3
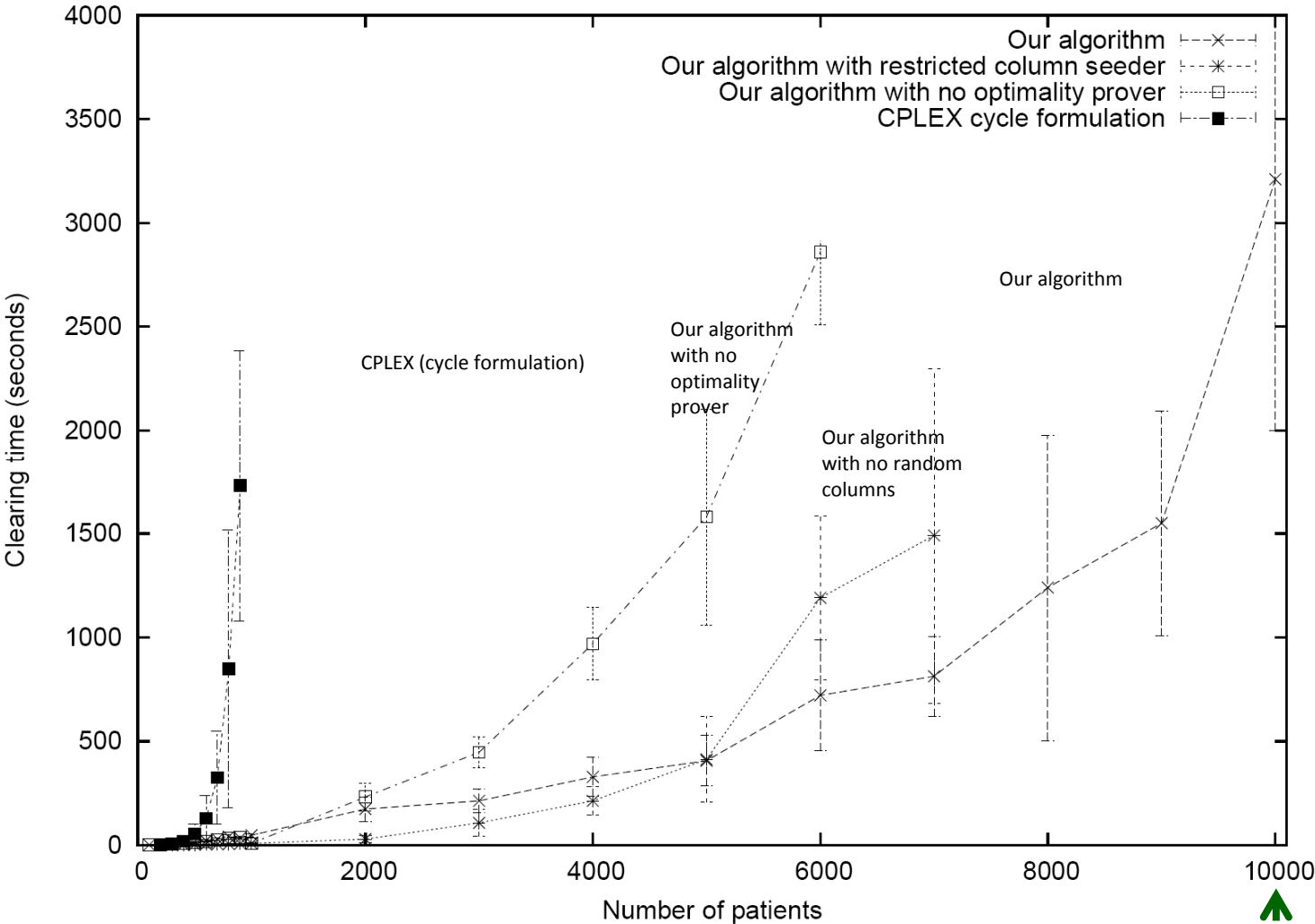
Recap: Main message

- This algorithm made modern kidney exchanges able to be cleared in a scalable way

    - Techniques that made this possible
        - Incremental problem formulation
        - Exploiting problem specific upper bounds in several ways
        - Other algorithmic ideas

Additional functionality
for modern kidney exchanges
supported by this algorithm
and its later enhancements

Side constraints

- Algorithm supports certain kinds of side constraints, e.g.,
  - Center A does not want to be in cycles longer than 2
  - Patient x does not want to be in a cycle longer than 2
  - Center B does not want to participate in altruistic donor chains of length greater than 3
  - …

Multiple willing donors per patient

- All their edges included in input graph
- Solver automatically uses at most one of the donors

Incorporating compatible pairs

- Why?
    - Patient can get a better kidney
    - Others get more/better matches

- Our algorithm supports this
    - Could preprocess so patient can't get worse kidney than her compatible donor brings
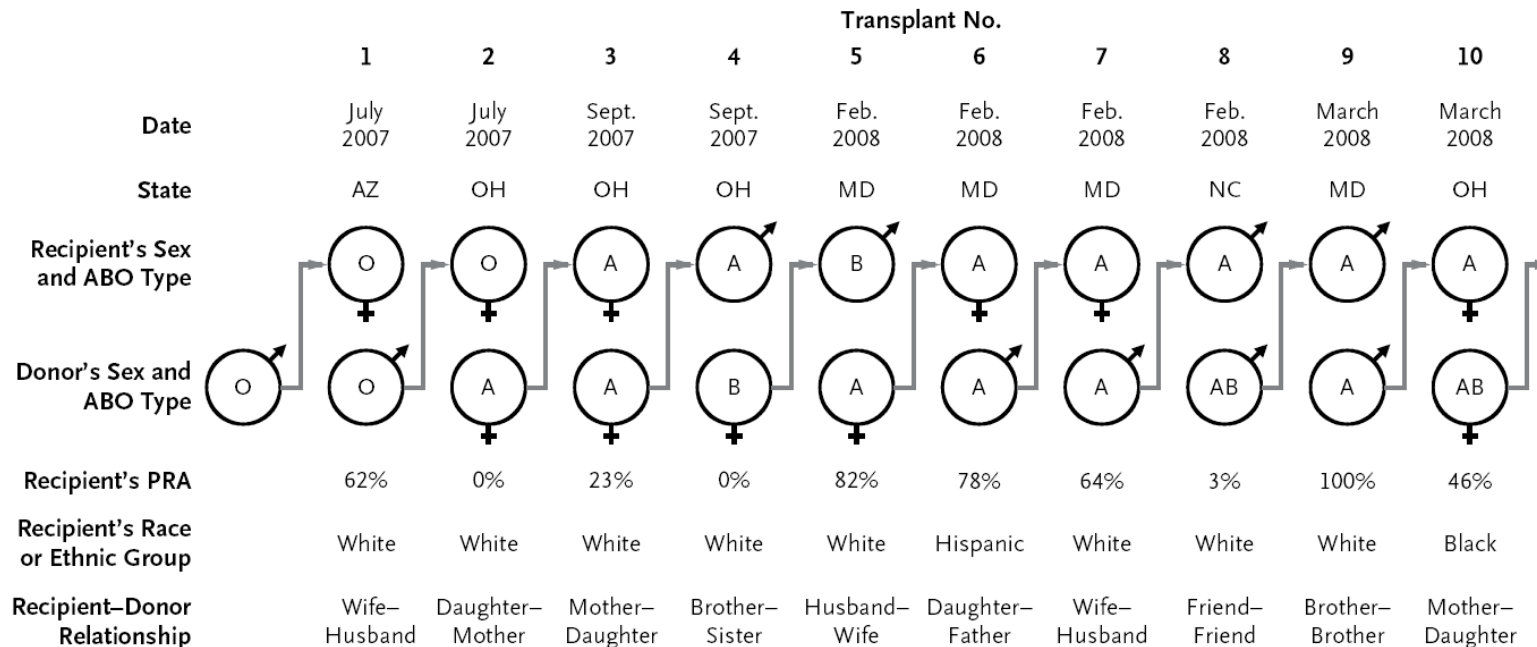
Incorporating list exchange(s)

- Algorithm can be used also if list exchanges were included in the optimization

Weights on edges

- Algorithm supports weights on edges (thus also on nodes)
- Weights can represent, e.g.,
    - Degrees of compatibility
    - Projected life years (potentially quality-adjusted)
    - Travel distance
    - Wait time
    - Transplanting children
    - Transplanting sensitized, hard-to-match patients
- Tradeoffs between efficiency and fairness
- AI autonomy

Never-ending altruistic donor (NEAD) chains
[Rees *et al. New England Journal of Medicine 2009*]

	Transplant No.									
	1	2	3	4	5	6	7	8	9	10
Date	July 2007	July 2007	Sept. 2007	Sept. 2007	Feb. 2008	Feb. 2008	Feb. 2008	Feb. 2008	March 2008	March 2008
State	AZ	OH	OH	OH	MD	MD	MD	NC	MD	OH
Recipient's Sex and ABO Type	O	O	A	A	B	A	A	A	A	A
Donor's Sex and ABO Type	O	O	A	A	B	A	A	A	AB	A	AB
Recipient's PRA	62%	0%	23%	0%	82%	78%	64%	3%	100%	46%
Recipient's Race or Ethnic Group	White	White	White	White	White	Hispanic	White	White	White	Black
Recipient–Donor Relationship	Wife–Husband	Daughter–Mother	Mother–Daughter	Brother–Sister	Husband–Wife	Daughter–Father	Wife–Husband	Friend–Friend	Brother–Brother	Mother–Daughter

- We started the first NEAD when working with APD

- Different from closed chains [Gentry,…]

- Incorporated chains at UNOS 4/2011

- Our early approaches handled chains as cycles via 0-weight back edges

- Weighted edges to value
  - Different "bridge donor" kidneys, and
  - Not using up different altruistic donors
  - Based on blood & tissue type, likelihood of pulling out, …
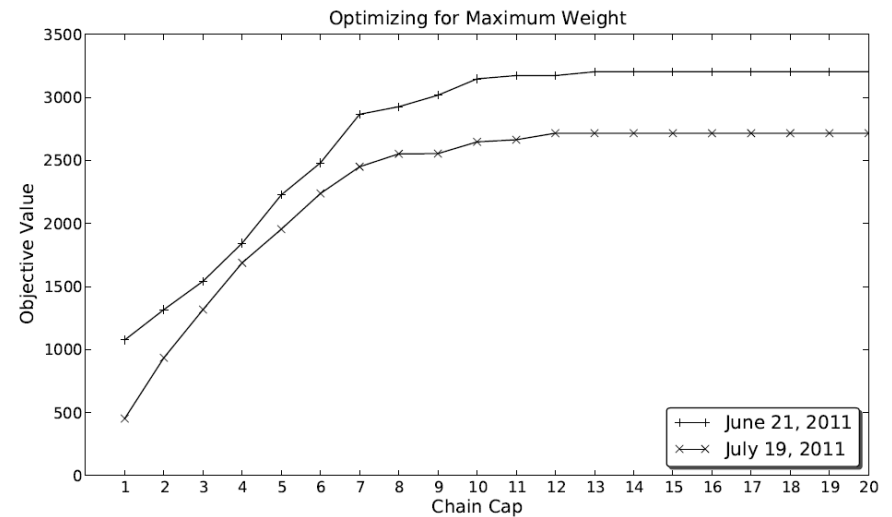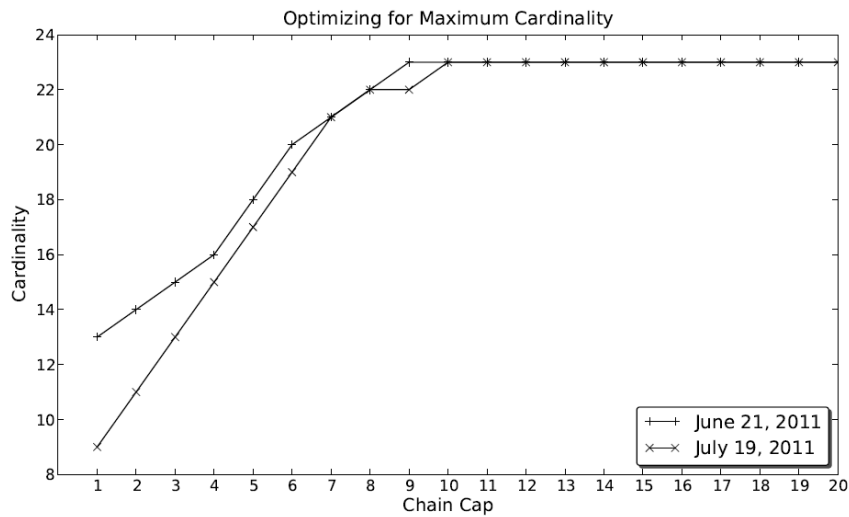
49

30-chain [*New York Times* 2/18/2012]

[National Kidney Registry]

50

Study of chains

[Dickerson, Procaccia & Sandholm, *AAMAS-12]*

Impact of within-batch chain cap on UNOS data

# Theory: Short chains suffice

- ABO-model with tissue type incompatibility
- Large, unweighted graph

THEOREM 1. *Assume that $\bar{\gamma} < 2/5$, $\mu_O < 3\mu_A/2$, and $\mu_O > \mu_A > \mu_B > \mu_{AB}$. Then with high probability $G(n)$ has an efficient allocation (i.e., one that saves as many patients as possible) that uses only cycles of length at most 3 and chains of length at most 3.*

Why the discrepancy?

- Possible reasons:
  - Unweighted – not the (only) reason
  - UNOS data not "large"
  - Uniform tissue type incompatibility model not realistic
- Experiments using Saidman et al. generator:
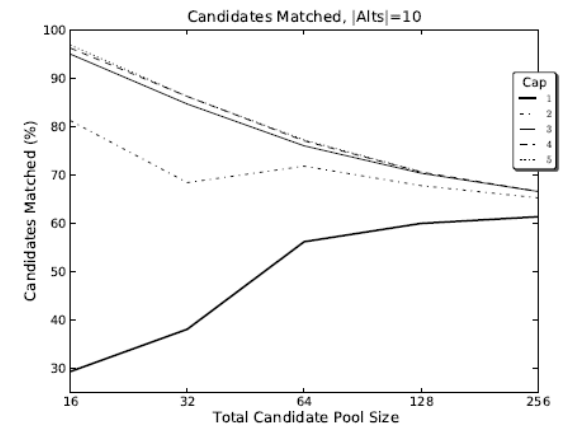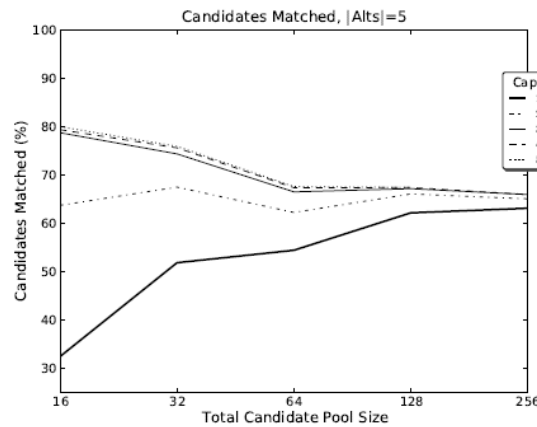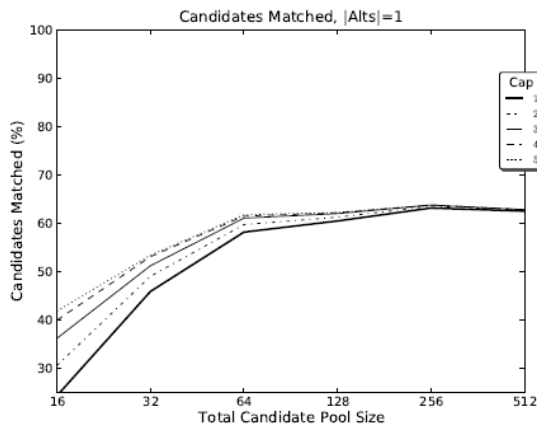
- In dynamic experiments, a chain cap of 4 was best

Newer scalable clearing algorithms

Better batch algorithms, infinite chain caps

[Anderson et al. *PNAS 2015*]

- Builds on the prize-collecting traveling salesperson problem [Balas Networks 1989]
  - 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)
- They maintain decision variables for all cycles of length at most L, but build chains in the final solution from decision variables associated with individual edges
- Then, an exponential number of constraints could be required to prevent the solver from including chains of length greater than K; these are generated incrementally until optimality is proved.
  - Leverage cut generation from PC-TSP literature to provide stronger (i.e. tighter) IP formulation

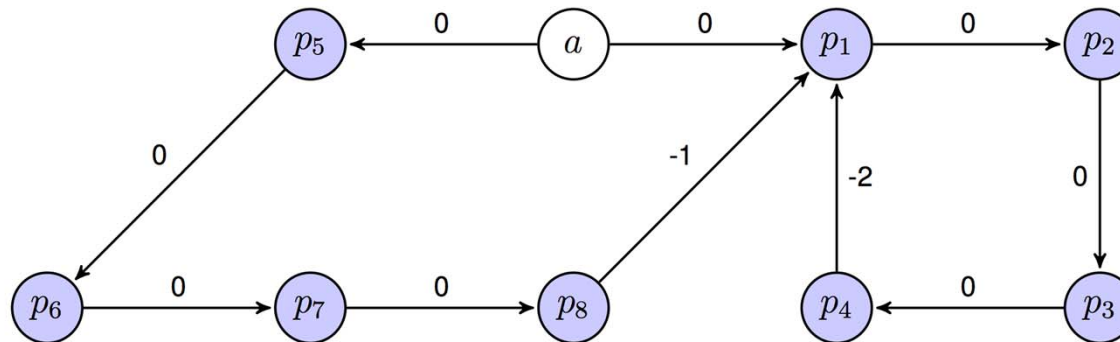# Better batch algorithms under within-batch chain caps

- Idea: solve a structured alternate optimization problem that implicitly prices variables
- Price (for max card, weight): $w_c - \Sigma_{v\,in\,c}\,\delta_v$
  - But: $w_c = \Sigma_{e\,in\,c}\,w_e$
- Take $G=(V,E)$, create $G'=(V,E)$ s.t. all edges $e = (u,v)$ are reweighted $r_e = \delta_v - w_e$
  - Positive price cycles in G = negative weight cycles in G'
- Bellman-Ford finds shortest paths
  - Undefined in graphs with negative weight
    - *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 $O(|V||E|\text{cap})$, but …

Better batch algorithms under within-batch chain caps

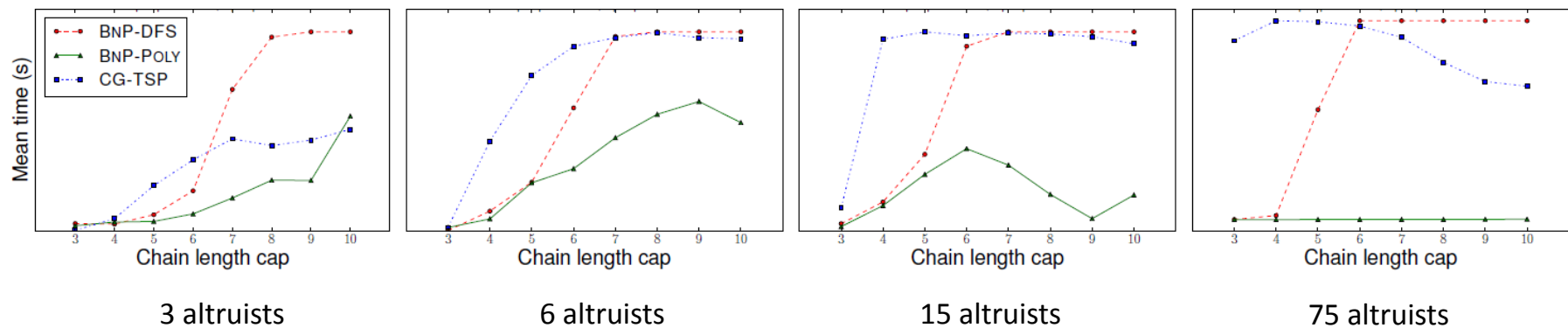- Necessary to prevent **internal looping** during modified B-F

- Now pricing runs in time $O(|V||E|\text{cap}^2)$

Better batch algorithms under within-batch chain caps

[Plaut, Dickerson, Sandholm *AAAI-16*]

Experiment with 300 vertices, 1 hour time limit
on realistic generated UNOS graphs

3 altruists   6 altruists   15 altruists   75 altruists

59

Compact formulations

- Previous models: exponential #constraints (CG methods)
  or #variables (B&P methods)

- Let L be upper bound on #cycles in a final matching

- Create L copies of compatibility graph

- Search for a *single* cycle or chain in each copy
  - (Keep cycles/chains disjoint across graphs)

Cycle 1   Cycle 2   Complete solution
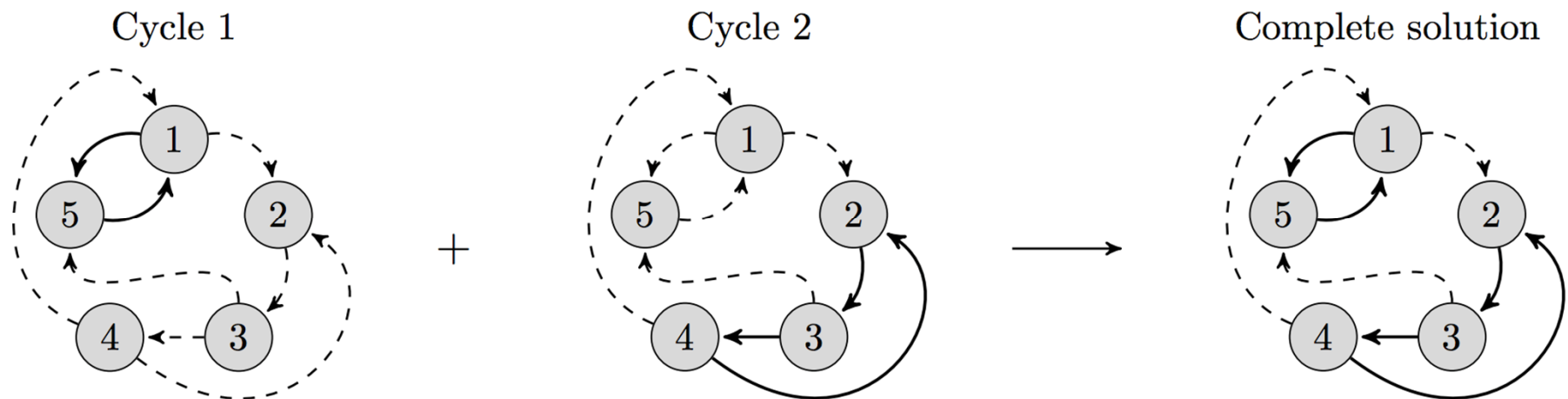
Compact formulations

$$x_{ij}^l = \begin{cases} 1 & \text{if arc } (i,j) \text{ is selected to be in copy } l \text{ of the graph,} \\ 0 & \text{otherwise.} \end{cases}$$

maximize
$$\sum_l \sum_{(i,j) \in A} w_{ij} x_{ij}^l \tag{7a}$$

subject to
$$\sum_{j:(j,i) \in A} x_{ji}^l = \sum_{j:(i,j) \in A} x_{ij}^l \qquad \forall i \in V, \forall l \in \{1, ...L\} \tag{7b}$$

$$\sum_l \sum_{j:(i,j) \in A} x_{ij}^l \leqslant 1 \qquad \forall i \in V \tag{7c}$$

$$\sum_{(i,j) \in A} x_{ij}^l \leqslant k \qquad \forall l \in \{1, ...L\} \tag{7d}$$

$$x_{ij}^l \in \{0, 1\}. \qquad \forall (i,j) \in A, \forall l \in \{1, ...L\} \tag{7e}$$

7a: max edge weights over all graph copies

7b: give a kidney <-> get a kidney *within that copy*

7c: only use a vertex once

7d: cycle cap

Poly #constraints **and** #variables!

Compact formulations for within-batch chain caps
[ongoing with David Manlove & James Trimble]

- Previous: edge is in a cycle/chain or not
  - Weak LP relaxation

- Idea: *where* in the cycle/chain does the edge exist?
  - "Position indexed"

Compact formulations for within-batch chain caps [ongoing with David Manlove & James Trimble]

$$\max \quad \sum_{(i,j)\in A}\sum_{k\in\mathcal{K}(i,j)} w_{ij}y_{ijk} + \sum_{c\in C} w_c z_c \tag{3a}$$

$$\text{s.t.} \quad \sum_{j:(j,i)\in A}\sum_{k\in\mathcal{K}(j,i)} y_{jik} + \sum_{c\in C:\, i \text{ appears in } c} z_c \leq 1 \qquad i \in P \tag{3b}$$

$$\sum_{j:(i,j)\in A} y_{ij1} \leq 1 \qquad i \in N \tag{3c}$$

$$\sum_{\substack{j:(j,i)\in A \wedge \\ k\in\mathcal{K}(j,i)}} y_{jik} \geq \sum_{j:(i,j)\in A} y_{i,j,k+1} \qquad \begin{array}{l} \forall\, i \in P, \\ \quad k \in \{1,\ldots,K-1\} \end{array} \tag{3d}$$

$$y_{ijk} \in \{0,1\} \qquad (i,j) \in A, k \in \mathcal{K}(i,j) \tag{3e}$$

$$z_c \in \{0,1\} \qquad c \in C \tag{3f}$$

3a: max weight of edges in chains + weight of cycles
3b: each pair is in at most chain/cycle
3c: each NDD has at most one used out edge
3d: if an edge is used at position k+1 in chain, there must be an appropriate edge used at position k in that chain

How to choose a formulation

- Comparison of LP relaxations
  - Column-generation-based tend to have tighter relaxations (so far!)
- Business constraints
  - Constraint-generation-based approaches outperform when feasible matching space is less constrained
    - E.g., no chain cap …
  - Column-generation-based approaches allow for more expressive objective functions (so far!)
    - E.g., general stochastic matching

Fielded kidney exchanges

- NEPKE (started 2003-04, now closed)
- United Network for Organ Sharing (UNOS)
- Alliance for Paired Donation
- Paired Donation Network (now closed)
- National Kidney Registry
- San Antonio
- Mayo Clinic
- St. Barnabas Compassionate Share
- Canada
- Netherlands
- UK
- Australia
- Portugal
- Israel (about to start)
- Sweden (about to start Q1 2016)
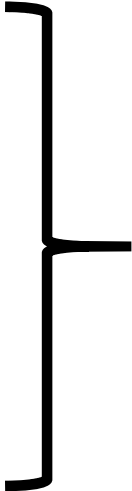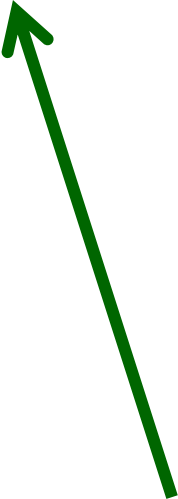
~600 transplants in US per year, mainly via open chains

Only US one that uses purely algorithmic matching

State of practice

- United States:
  - Started in the US in 2003-04 (NEPKE; pool now merged to UNOS pool), cycles
  - UNOS nationwide exchange (only one that is run by algorithms)
  - Several private exchanges (NKR, APD) and single-center "exchanges"
  - ~600 transplants per year, mostly via open chains
  - Multi-listing, competition, sniping
  - Cadence: twice a week or even multiple times per day

- Netherlands: national, chains, algorithms

- UK: national, quarterly, hierarchical algorithmic approach

- Canada: national, quarterly, CPLEX

- Nascent: Australia, Portugal, Israel, Sweden, …

- International: one swap at APD so far via manual matching

Tutorial outline

- Introduction & preliminaries
- Optimization models & state of the practice
- Deep dive into dimensions of kidney exchange:
  - Short-term uncertainty
  - Fairness vs economic efficiency
  - Long-term uncertainty & dynamic optimization
  - Incorporating human expert judgment in better ways
  - Incentives & mechanism design
- Other organ exchanges
- Conclusion & open research problems

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: maximize *expected* value of the (batch) transplants

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*

$$u(c) = \sum w_e \bullet \prod q_e$$

Value of successful cycle

Probability of success

Failure-aware model…

- Discounted utility of an (unweighted for simplicity) 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 execute   Chain executes in entirety

- These cycle and chain utilities are not the same one would get by simply replacing the weight of each edge by (weight * success probability)

- Utility of a solution M: $u(M) = \sum u(c)$

Our problem

- *Discounted clearing problem* is to find matching M^* with highest discounted utility

Maximum cardinality

Maximum expected transplants

3

1

2
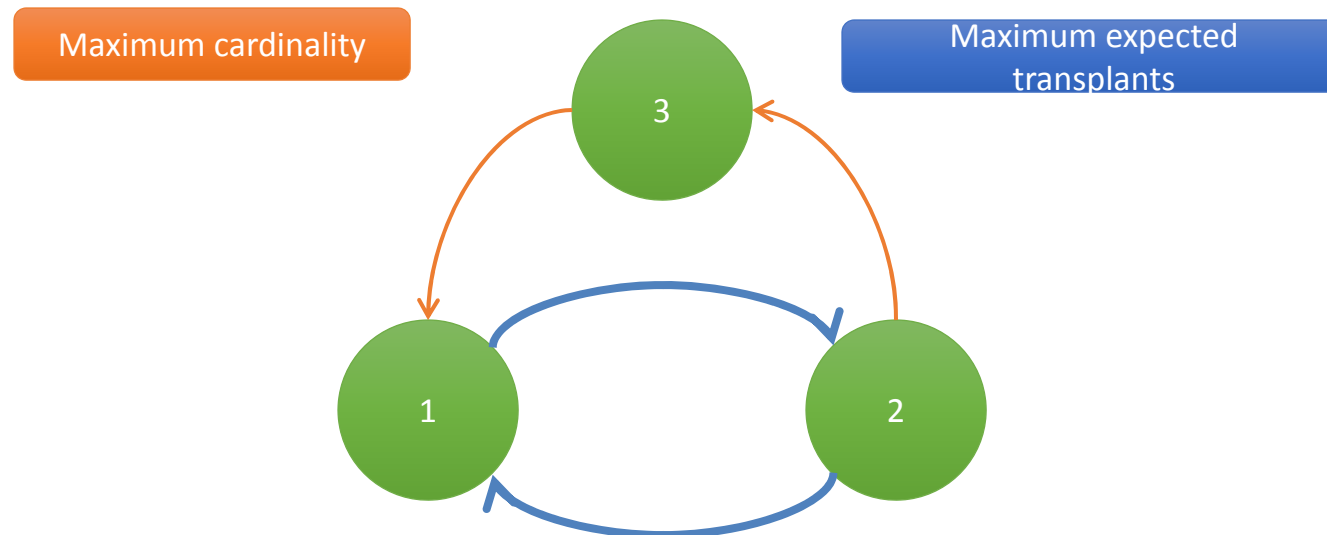
Motivating theorem

- 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 $\alpha, \beta > 0$, given a large $G(n, \alpha n, \beta/n)$, w.h.p. there exists some matching M' s.t. for every maximum cardinality matching M,
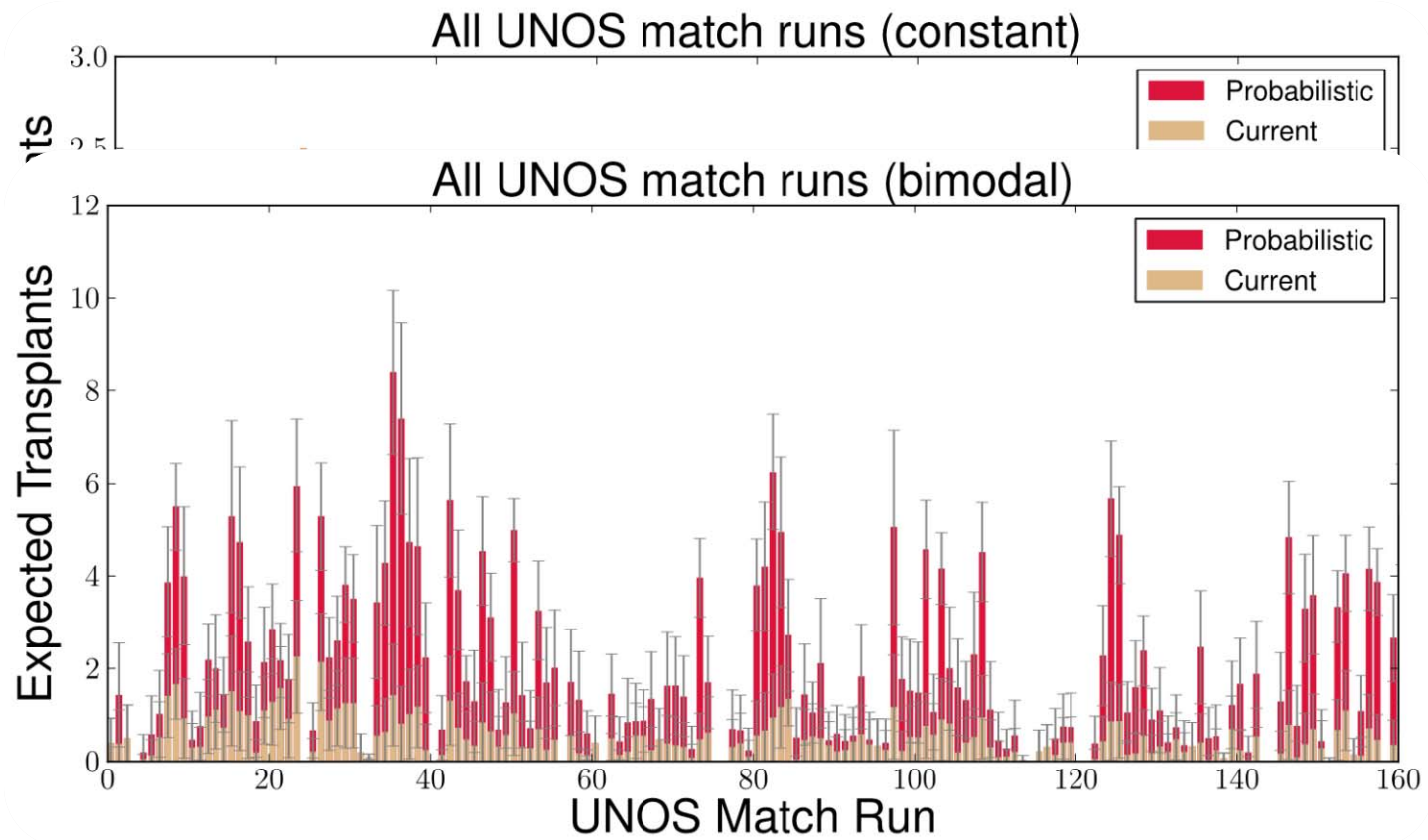>
> $$u_q(M') \geq u_q(M) + \Omega(n)$$

Proof sketch: Counting Y-gadgets

(a) A Y gadget. (b) The maximum cardinality matching M_Y. (c) The matching M_Y'.
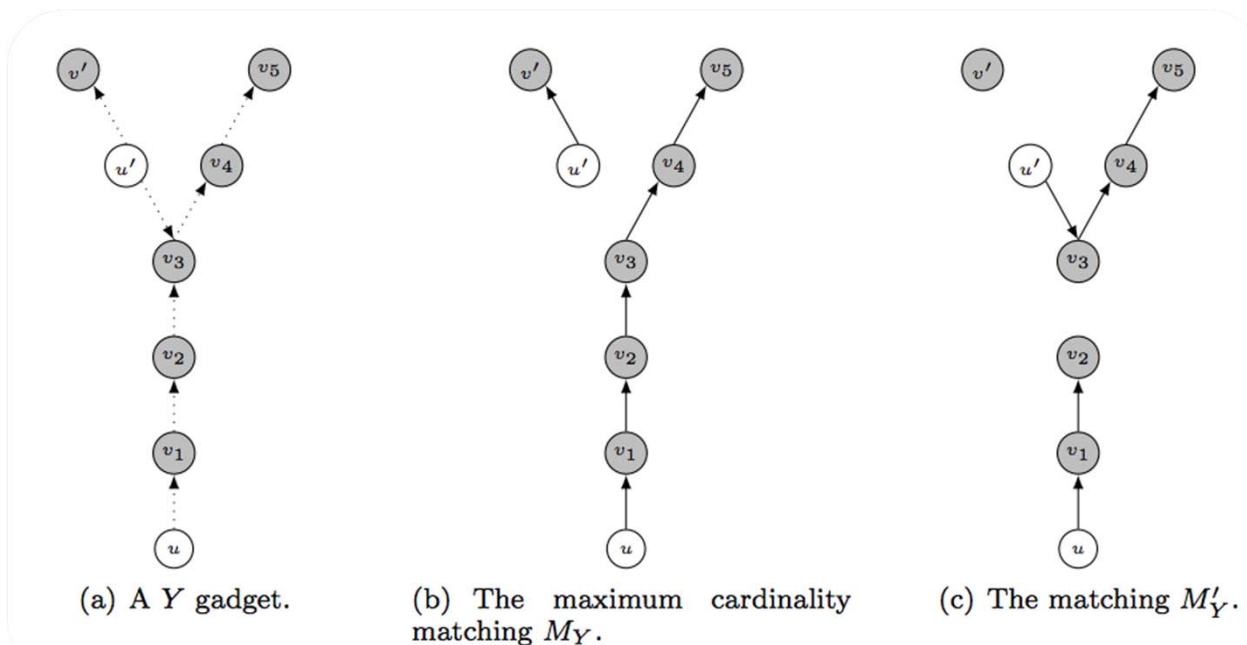
- For every structure X of constant size, w.h.p. can find $\Omega(n)$ structures isomorphic to X *and isolated from the rest of the graph*
- Label them (alt vs. pair): flip weighted coins, constant fraction are labeled correctly → constant × $\Omega(n) = \Omega(n)$
- Direct the edges: flip 50/50 coins, constant fraction are entirely directed correctly → constant × $\Omega(n) = \Omega(n)$

In theory, we're losing out on *expected actual transplants* by maximizing match cardinality.

… What about in practice?

All UNOS match runs (constant)

3.0

Probabilistic

Current

2.5

All UNOS match runs (bimodal)

Expected Transplants (y-axis, 0 to 12)

Legend: Probabilistic, Current

UNOS Match Run (x-axis, 0 to 160)

75

Solving this new problem

- Real-world kidney exchanges are still small
  - UNOS pool: 281 donors, 260 patients [2 Feb 2015]
- *Un*discounted clearing problem is NP-hard when cycle/chain cap $L \geq 3$ [Abraham et al. 2007]
  - Special case of our problem

- Current UNOS solver will not scale to this problem
  - Empirical intractability driven by chains

Algorithm changes for this probabilistic setting

- Use chain extension in pricing problem
  - **Theorem.** Don't have to extend a chain by any finite #steps if optimistic infinite extension has negative expected value:

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

Optimistic future value of infinite extension

Discounted utility of current chain

Donation to waitlist

Pessimistic sum of LP dual values

  - Ordering heuristics for cycle and chain generation
- Upper bound now hard
  - **Theorem.** Discounted clearing NP-complete (even with no chains or cycle length cap)
  - So, we use looser bound: solve with $w'_e = (1 - p_{fail}) w_e$
- Lower bound still easy
 - **Theorem.** Discounted clearing with 2-cycles polytime

Scaling experiments

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

- Runtime limited to 60 minutes; each instance given 8GB of RAM.
- |V| represents #patient-donor pairs; additionally, 0.1|V| altruistic donors are present.

Dynamic experiment with failures

24 weeks; Bimodal failure probability; #altruists = 0.1 * #pairs
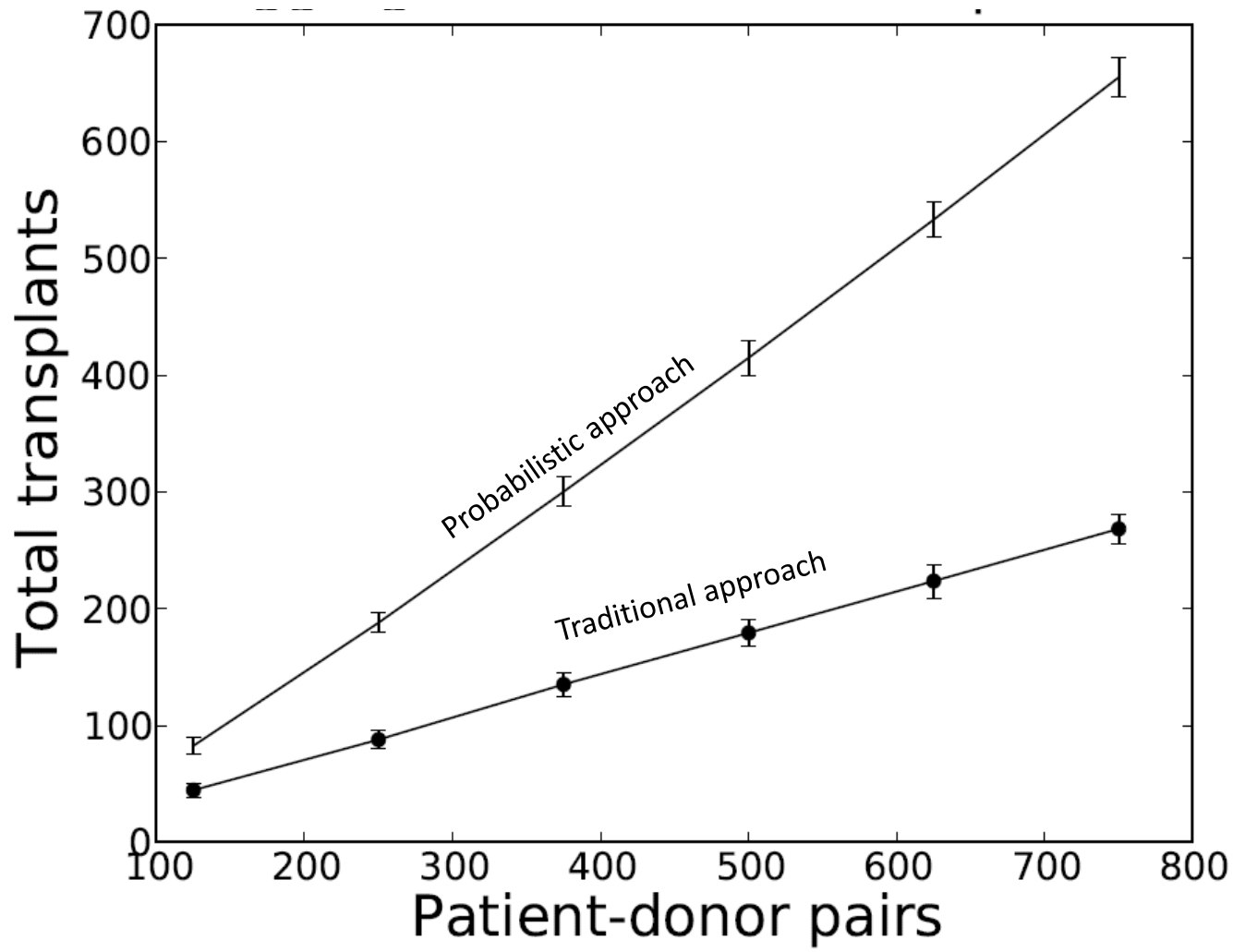
Pre-match edge testing

[Blum et al. *EC-13* and *EC-15*]

- Complementary idea: perform a *small amount* of testing before a match run to query for (non)existence of edges
  - more extensive medical testing
  - donor interviews
  - surgeon interviews, etc.

- For 2-cycles only: stochastic matching

The power of two crossmatches
[Blum et al. *EC-13*]

- Cast as a general *stochastic matching* problem:

Given a graph $G(V,E)$, choose subset of edges S such that:

$$|M(S)| \geq (1-\varepsilon)\ |M(E)|$$

Need: "sparse" *S*, where every vertex has O(1) incident tested edges

- Initially: 2-cycles only (= undirected), **at most 2** tests per vertex → polytime algorithm for this

Pre-match testing in rounds

[Blum et al. *EC-15*]

- What about testing a variable number of edges per vertex?

- What if we can test edges, get feedback, test more?

- What about 3-cycles, chains?

- Cast as an *adaptive* stochastic *k-set packing* problem:
  - Query edges in *rounds*, where each round tests at most one incident edge per vertex

General theoretical results

Adaptive: select one edge per vertex per *round*, test, repeat

Stochastic matching:
(1-ε) approximation with $O_\varepsilon(1)$ queries per vertex, in $O_\varepsilon(1)$ rounds

Stochastic k-set packing:
(2/k − ε) approximation with $O_\varepsilon(1)$ queries per vertex, in $O_\varepsilon(1)$ rounds

Non-adaptive: select O(1) edges per vertex, test all at once

Stochastic matching:
(0.5-ε) approximation with $O_\varepsilon(1)$ queries per vertex, in 1 round

Stochastic k-set packing:
$(2/k − ε)^2$ approximation with $O_\varepsilon(1)$ queries per vertex, in 1 round

Adaptive algorithm

For *R* rounds, do:
1. Pick a max-cardinality matching *M* in graph *G*, minus already-queried edges that do not exist
2. Query all edges in *M*

Input Graph
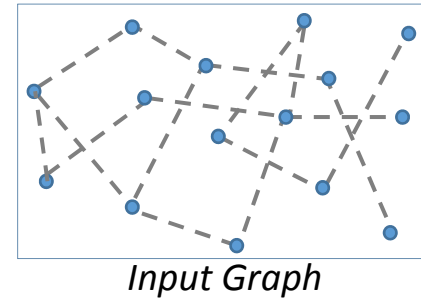
r	Base graph	Matching picked	Result of queries
1:			
2:			

Intuition for adaptive algorithm

- If at any round *r*, the best solution on edges queried so far is *small* relative to omniscient …
  - … then current structrure admits *large* number of unqueried, disjoint augmenting structures
  - For *k=2*, simply augmenting paths
- Augmenting structures might not exist, but can query in parallel in a single round
  - Structures are constant size $\rightarrow$ exist with constant probability
  - Structures are disjoint $\rightarrow$ queries are independent
  - $\rightarrow$ Close a constant gap per round

UNOS, 2- and 3-cycles, with chains

At p=0.5, **one** edge test per vertex → +21% OPT

Legend:
- $R = 0$
- $R = 1$
- $R = 2$
- $R = 3$
- $R = 4$
- $R = 5$

Y-axis: Fraction of Omniscient

X-axis: Edge Failure Rate
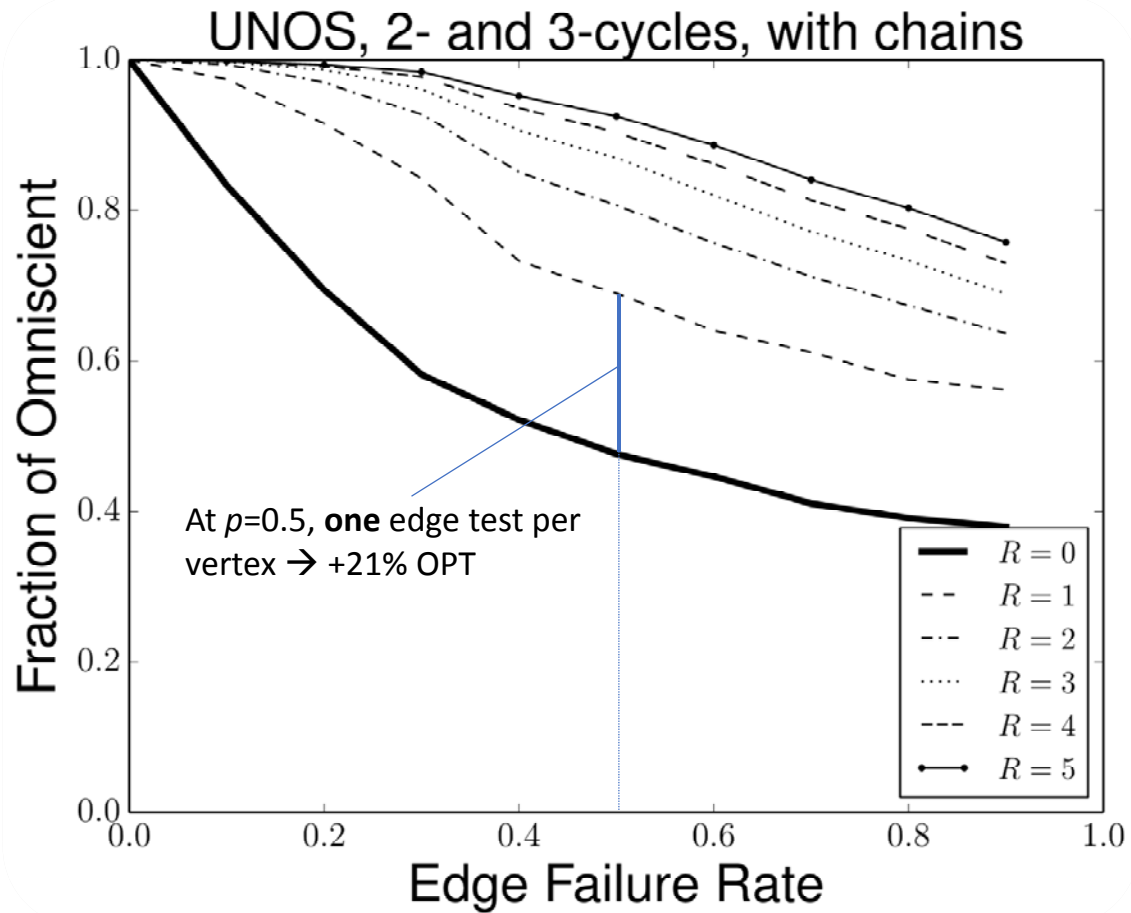
Even 1 or 2 extra tests would result in a huge lift

Tutorial outline

- Introduction & preliminaries
- Optimization models & state of the practice
- Deep dive into dimensions of kidney exchange:
  - Short-term uncertainty
  - Fairness vs economic efficiency
  - Long-term uncertainty & dynamic optimization
  - Incorporating human expert judgment in better ways
  - Incentives & mechanism design
- Other organ exchanges
- Conclusion & open research problems

An initial definition of fairness

[Roth, Sönmez, Ünver *JET 2006*]

- **Matching lottery**: distribution over possible matchings
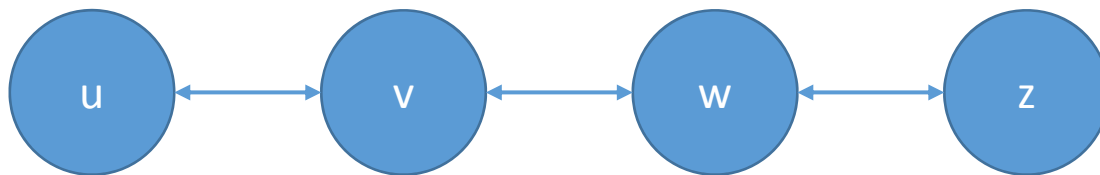
- Example lottery: $\ell_{\mu_1} = \ell_{\mu_3} = 0.5$

  $\ell_{\mu_2} = \ell_{\mu_4} = \ell_{\mu_5} = 0$
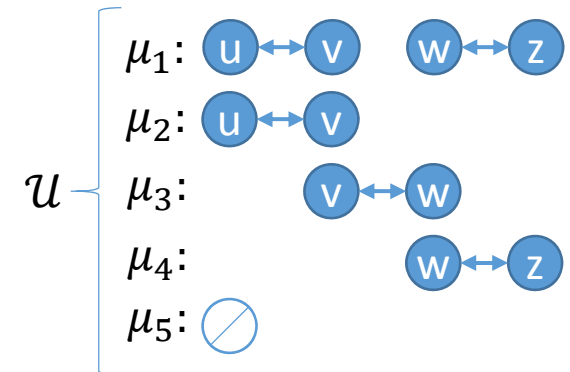
- **Utility profile**: total probability given to each vertex

$$\left(x_u^\ell, x_v^\ell, x_w^\ell, x_z^\ell \right) = (0.5, 1.0, 1.0, 0.5)$$

An initial definition of fairness

- One utility profile **x** is said to **Lorenz dominate** another utility profile **y** if:
  - Sort both profiles in increasing order
  - Condition 1: For $t = \{1..n\}$, $\sum_{s=1}^{t} x_s \geq \sum_{s=1}^{t} y_s$
  - Condition 2: There exists t s.t. $\sum_{s=1}^{t} x_s > \sum_{s=1}^{t} y_s$
- Previous graph: Lorenz **dominant** profile assigns all weight to μ_1: (u) ↔ (v) (w) ↔ (z)

Thm: There is a unique Lorenz-dominant utility profile.

- Algorithm is exponential in graph size (#odd components in Gallai-Edmonds Decomposition)
- Applies to 2-cycles only

A fast egalitarian mechanism

[Li et al. *AAMAS-14*]
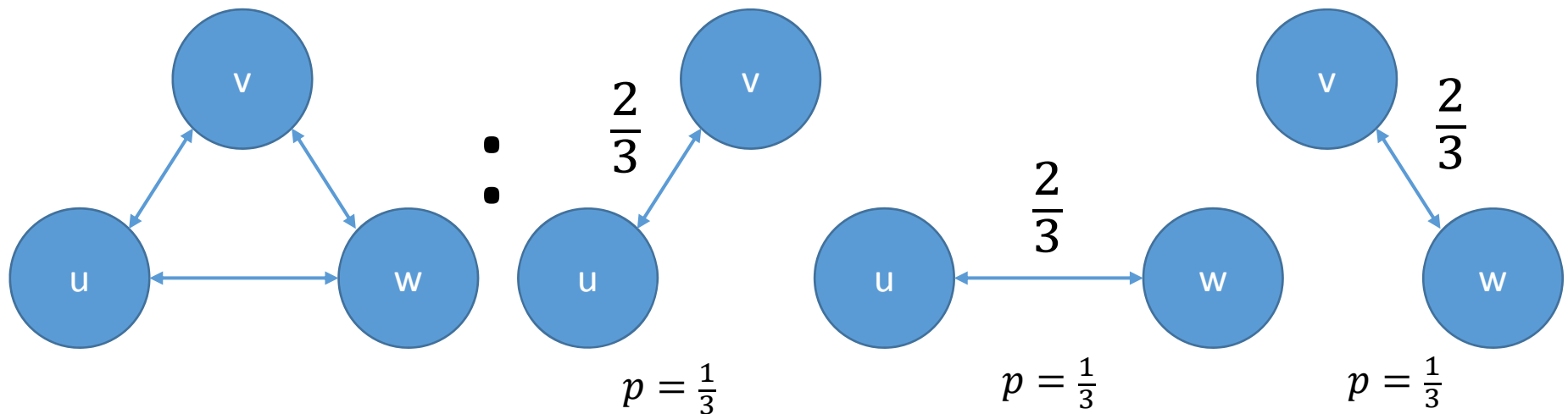
Thm: There is a polytime—$O(n^3)$—algorithm to find the Lorenz-dominant utility profile.

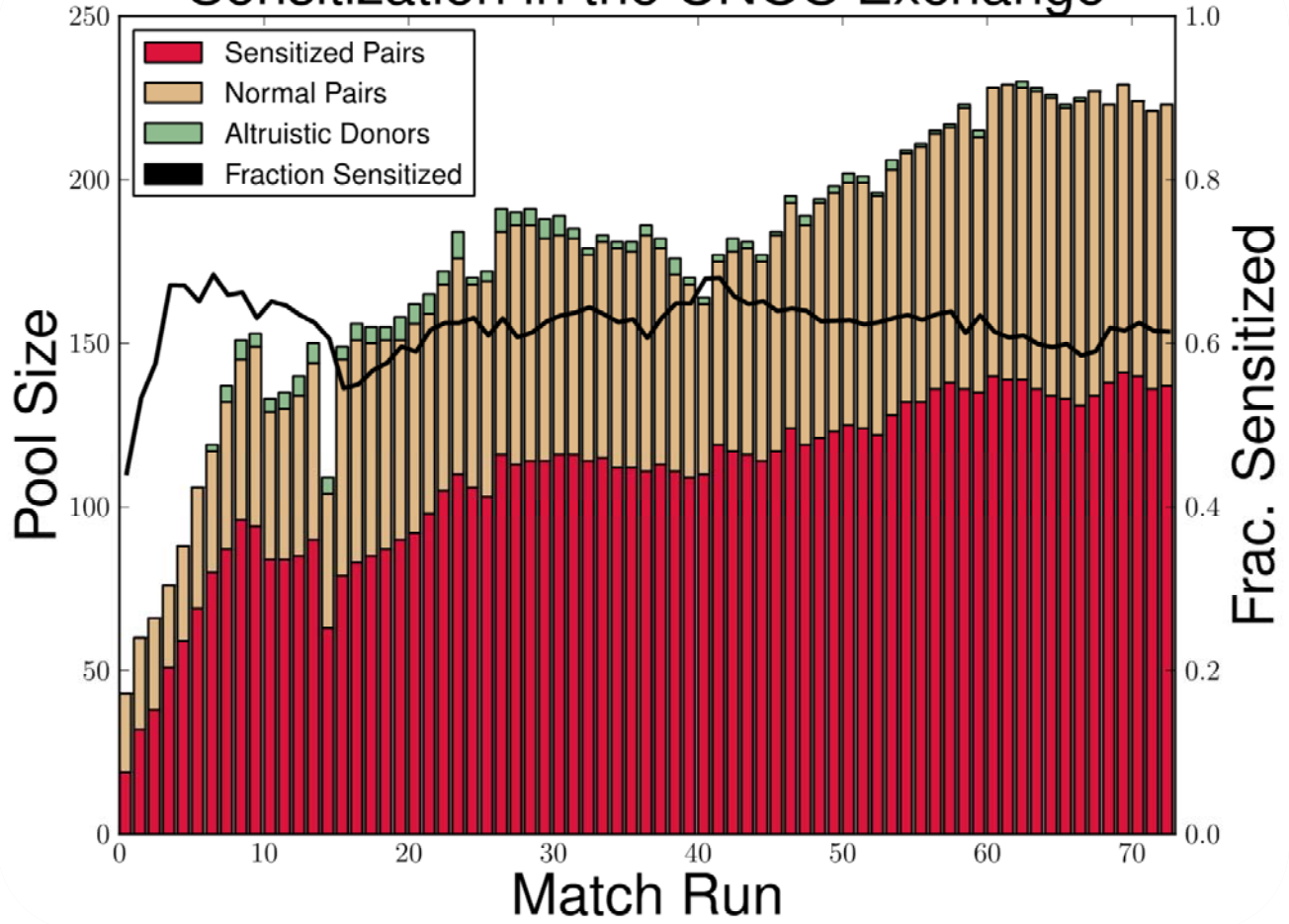- **Lorenz-dominant allocations not guaranteed to exist in >2-cycles and chains matching**

Balancing efficiency and equity

- Fielded kidney exchanges match under **utilitarian** or near-utilitarian rules:
  - i.e., "match as many as possible"
  - (often with some ad-hoc vertex weighting)

- This can marginalize hard-to-match pairs

Present-day marginalization

- Highly-sensitized patients have elevated antibody levels that negatively react to foreign tissue
  - Harder to find a matching donor

- CPRA score estimates % of incompatible donors
  - 0%: low sensitization, easy to find a match
  - 100%: high sensitization, hard to find a match
  - Typical definition of highly sensitized is 80% (and increasing); 80% is what we will use below

- 17% of adult deceased-donor kidney waitlist is highly-sensitized
  - ~60% in kidney exchange

Sensitization in the UNOS Exchange

Legend:
- Sensitized Pairs
- Normal Pairs
- Altruistic Donors
- Fraction Sensitized

X-axis: Match Run
Left Y-axis: Pool Size
Right Y-axis: Frac. Sensitized

"The needs of the many outweigh the needs of the few or the one."

- … Generally *not followed* in healthcare

Definition

Price of fairness: relative loss in system efficiency due to using a fair objective

[Bertismas, Farias, Trichakis *OR 2011*,
 Caragiannis et al. *WINE-09*]

98

"Price of fairness" in kidney exchange

[Dickerson, Procaccia, Sandholm *AAMAS-14, invited AIJ*]

- **Clearing problem**: find a matching M^* that maximizes utility function

$$M^* = \operatorname*{argmax}_{M \in \mathcal{M}} u(M)$$

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

$$POF(\mathcal{M}, u_f) = \frac{u(M^*) - u(M_f^*)}{u(M^*)}$$

- $V_{\{L,H\}}$: lowly-, highly-sensitized vertices
- λ : fraction of pool that is lowly-sensitized
- $p_{\{L,H\}}$: prob. ABO-compatible is tissue-type incompatible
- $p = \lambda p_L + (1-\lambda)p_H$: average level of sensitization

- "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 p < 2/5, $\lambda \geq$ 1-p, and "reasonable" distribution of blood types. Then, almost surely as $n \rightarrow \infty$,

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

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

In theory, the price of fairness is low
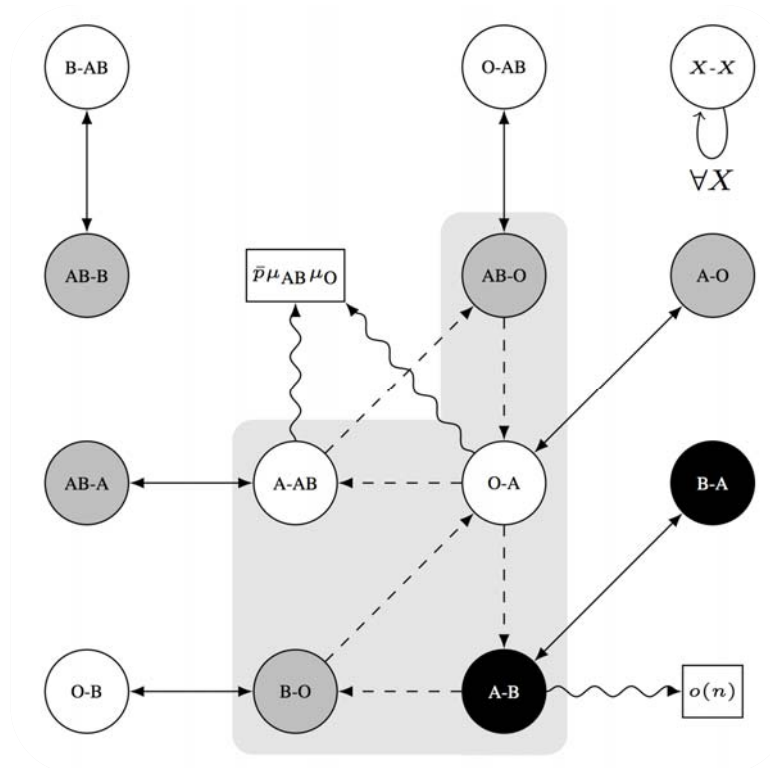
As many highly-sensitized patients as possible are matched; loss compared to the efficient matching of [Ashlagi & Roth EC-11] is shown with wavy lines.

From theory to practice

- Theoretical assumptions (standard):
  - Big graphs ("$n \rightarrow \infty$")
  - Dense graphs (constant p, p_L, p_H)
  - Cycles (no chains)
  - No post-match failures
  - Simplified patient-donor features

- Fairness criterion was *extremely* strict

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

Two fairness definitions

- Lexicographic:
  - Generalizes strict $u_{H>L}$ used in theoretical result
  - Requires fraction α of the maximum number of highly-sensitized patients that *could* be matched over *all* possible matchings
    - Chooses the largest matching among those

- Weighted:
  - A highly-sensitized patient counts for $(1+\beta)$ times as much as a lowly-sensitized patient

Implementing the two fairness notions within branch-and-price

- Lexicographic: <u>difficult</u>
  - Requires a *matching-wide* constraint
  - Finding a positive price cycle now requires solving an integer program (at every node in search tree)
  - In experiments we used CPLEX

- Weighted: <u>easy</u>
  - Re-weight edges according to fairness function
  - Match using our solver

Fairness experiments on UNOS data

- Our algorithm and code run the UNOS nationwide exchange
  - Algorithm computes a weighted efficient matching

- Applied both fairness definitions to 73 match runs, compared against fielded version
  - (All match runs from inception of exchange in Oct. 2010 through early Oct 2013)

Strict lexicographic fairness on UNOS data

Efficiency Loss (No Failure Prob)

Efficiency Loss (Constant Failure Prob.)

Efficiency Loss (Bimodal Failure Prob.)

Frac. Match Runs

Efficiency Loss (Frac. Objective)
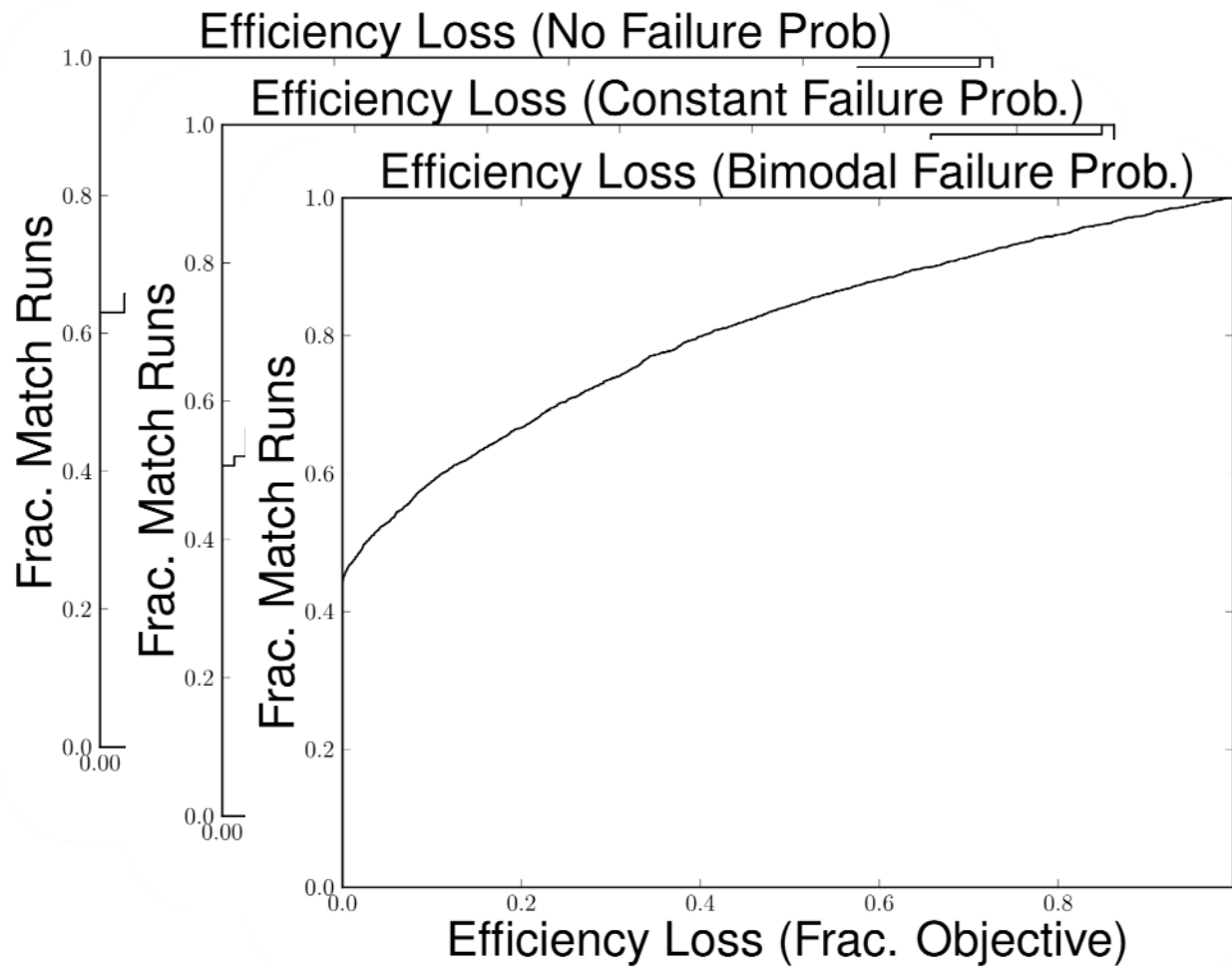
Weighted fairness on real UNOS data
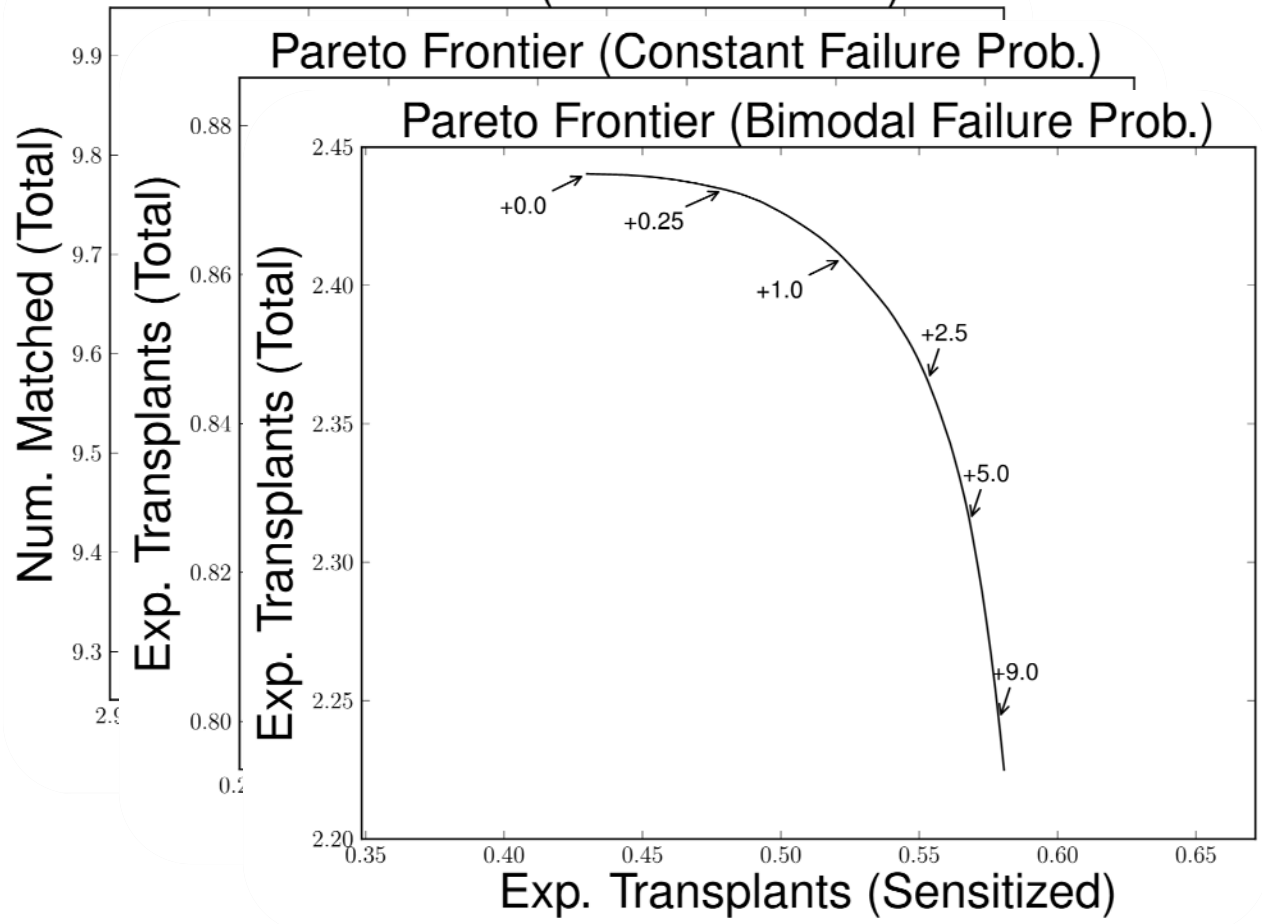
Experiments on generated data

- Two standard kidney exchange models:
    - [Saidman et al. 2006]: dense, parameterized by full US population, considers ABO and 3 levels of sensitization, etc.
    - [Ashlagi et al. *EC-12*+]: adaptation of sparse Erdos-Renyi graphs:
        - no blood type
        - *high* and *low* sensitization
            - constant prob of incoming edge to low-sensitized patients
            - $\Theta(1/n)$ prob to high-sensitized patients

- Created a third distribution, "Saidman-UNOS":
    - Saidman model parameterized by UNOS pool data (Oct 2013)

Avg (St.Dev.) efficiency loss under the strict fairness, for generated data

Size	Saidman (US)	Saidman (UNOS)	Ashlagi & Roth
10	0.24% (1.98%)	0.00% (0.00%)	0.98% (5.27%)
25	0.58% (1.90%)	0.19% (1.75%)	0.00% (0.00%)
50	1.18% (2.34%)	1.96% (6.69%)	0.00% (0.00%)
100	1.46% (1.80%)	1.66% (3.64%)	0.00% (0.00%)
150	1.20% (1.86%)	2.04% (2.51%)	0.00% (0.00%)
200	1.43% (2.08%)	1.55% (1.79%)	0.00% (0.00%)
250	0.80% (1.24%)	1.86% (1.63%)	0.00% (0.00%)
500	0.72% (0.74%)	1.67% (0.82%)	0.00% (0.00%)

- Distributions align more with our theoretical model (of a larger, stable exchange) than UNOS data does
 - Thus the price of fairness is much lower

Experiments on the interaction between failure-aware matching and fairness rules
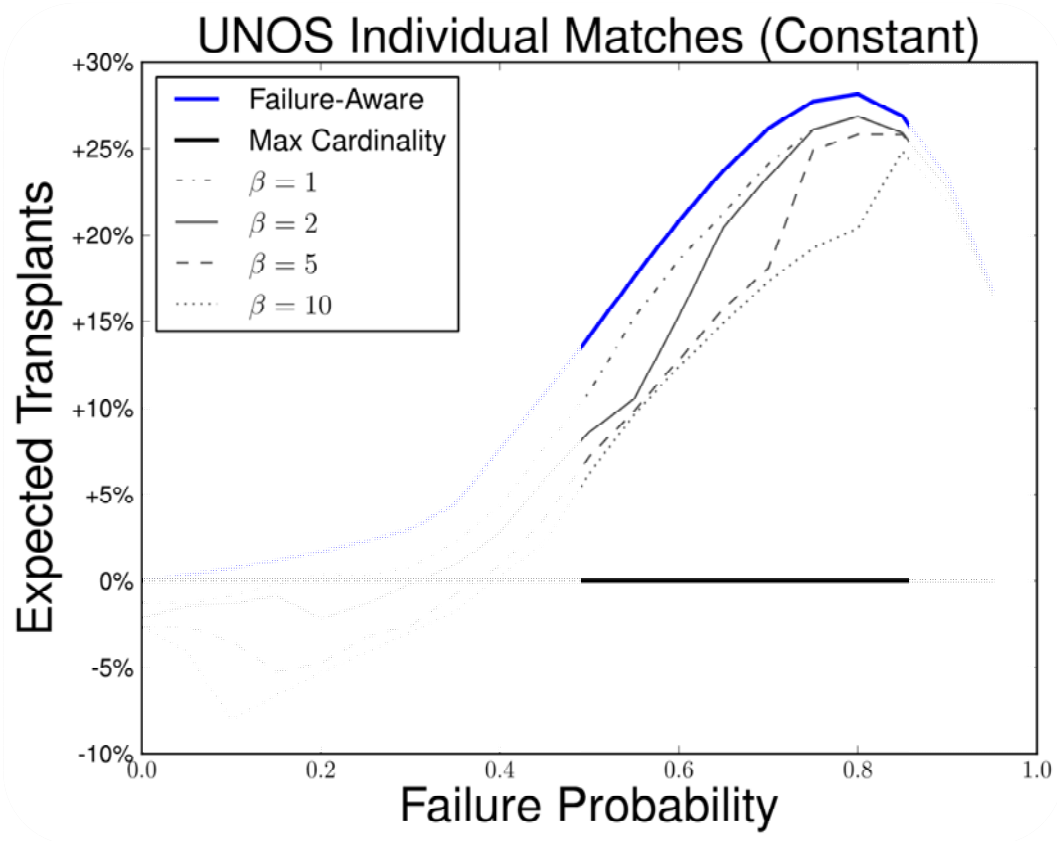
UNOS runs, weighted fairness, constant probability of failure (x-axis), increase in expected transplants over deterministic matching (y-axis)

114
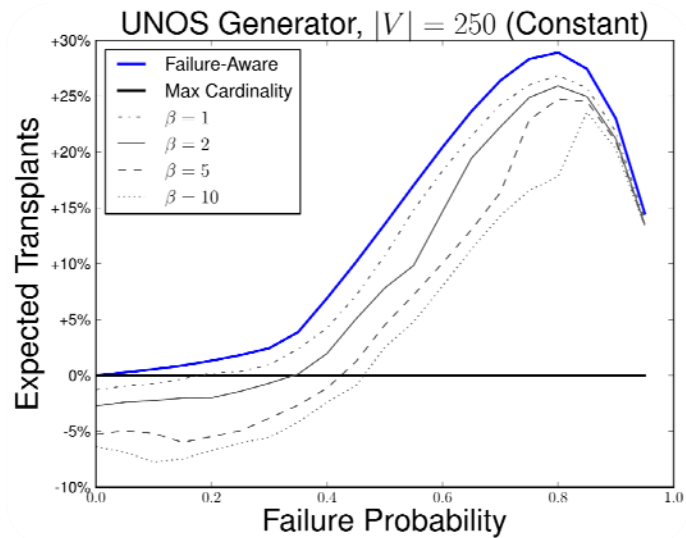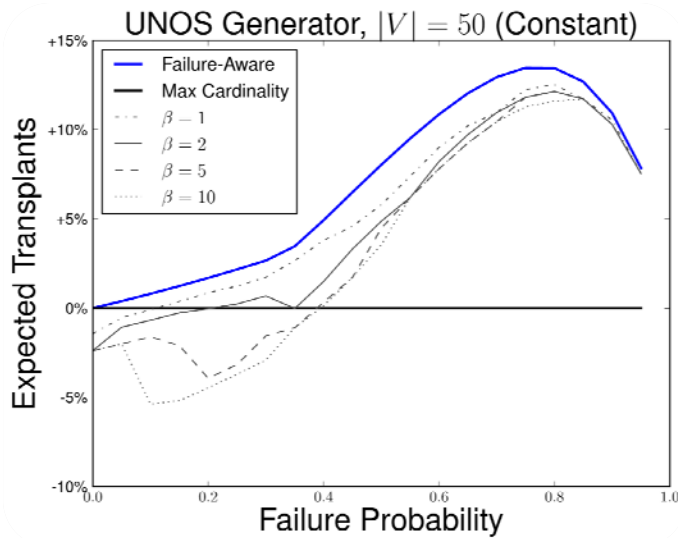
Generated UNOS runs, weighted fairness, constant probability of failure (x-axis), increase in expected transplants over deterministic matching (y-axis)
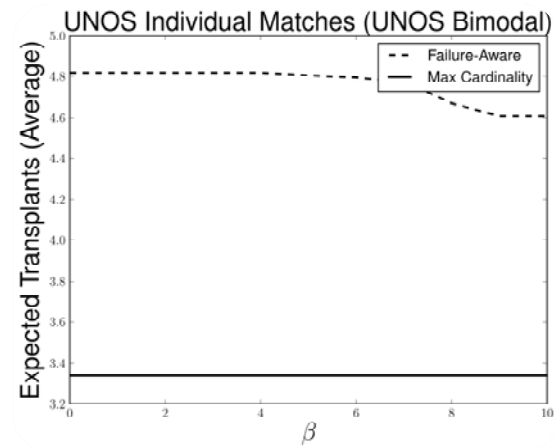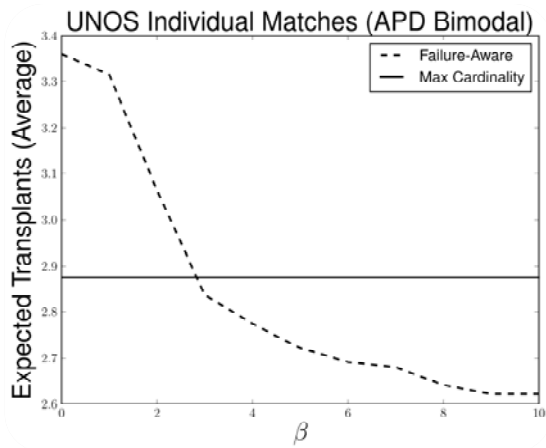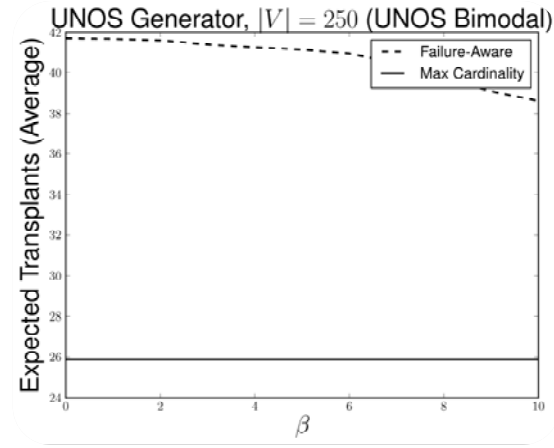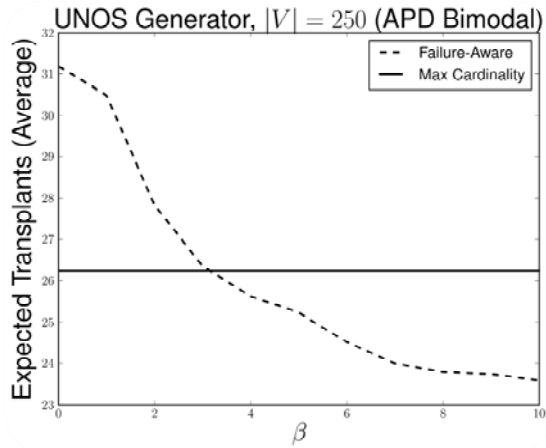
Generated (top row) and real (bottom row) UNOS runs, weighted fairness (x-axis), bimodal failure probability (APD failures in left column, UNOS failures in right column), increase in expected transplants over deterministic matching (y-axis)

Take-home message

- In theory, the price of fairness is small

- In practice, the situation is trickier – but *some* emphasis on fairness can be added without much drop in overall efficiency

- Present-day kidney exchange models and solvers are amenable to fairness criteria

Tutorial outline

- Introduction & preliminaries
- Optimization models & state of the practice
- Deep dive into dimensions of kidney exchange:
  - Short-term uncertainty
  - Fairness vs economic efficiency
  - Long-term uncertainty & dynamic optimization
  - Incorporating human expert judgment in better ways
  - Incentives & mechanism design
- Other organ exchanges
- Conclusion & open research problems

Dynamic kidney exchange

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

| Vertex Removal	Edge Removal	Vertex/Edge Add
Transplant, this exchange	Matched, positive crossmatch	Normal entrance
Transplant, deceased donor waitlist	Matched, candidate refuses donor	
Transplant, other exchange ("sniped")	Matched, donor refuses candidate	
Death or illness	Pregnancy, sickness changes HLA	
Altruist runs out of patience		
Bridge donor reneges		

Initial theoretical work

- Ünver [RES 2010] studies minimizing avg wait time in kidney exchange
  - 2-cycles or no cycle cap; no chains
  - No tissue type incompatibility
  - No pairs expire
  - Poisson arrivals
  - Proves that simple dispatch rules are optimal

- Zenios [Mgmt Sci 2002] studies maximizing avg quality-adjusted life years
  - 2-cycles only (no longer cycles, no chains)
  - Only two types of patient-donor pairs
  - Models exchange as a divisible birth and death process
  - No matching aspects of the problem
  - No patients expire, but long wait penalized by a fixed cost
  - Optimal policy is analytically derived
    - Limits the number of patients that can take part in exchange. Patients not admitted queue for altruistic donors (wait time here assumed zero)

No good prior-free online algorithm (even without chains)

- Proposition. No deterministic prior-free algorithm can achieve competitive ratio better than L/2
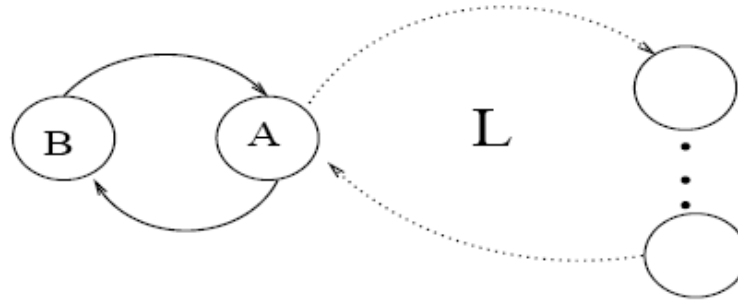
- Proposition. No prior-free algorithm can achieve competitive ratio better than 2 − (2/L)

- => Have the algorithm use distributional information

- But full stochastic optimization totally unscalable here

Family 1 of approaches [Awasthi and Sandholm IJCAI-09]

- At each step
  - Draw sample trajectories
  - Leverage our offline algorithm to pick an action, i.e., combination of cycles and chains (not policy)

Algorithm 1

Adaptation of REGRETS/CONSENSUS [Bent & van Hentenryck 04a,b, Mercier & van Hentenryck 07]
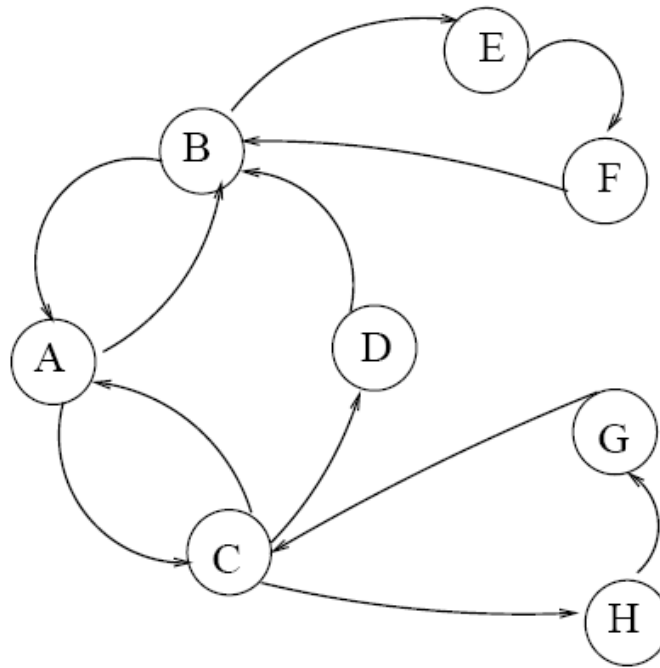
- For each cycle c in G_t, score(c)=0

- Generate scenarios s_1, …, s_m

- For each scenario s_i
  - S = solution to offline problem on $\{G_t, s_i\}$
  - For each cycle c in G_t,
    - If c is in S, score(c)=score(c)+value(S)
    - else score(c)=score(c) - δ

- Using the batch ILP, find a set of vertex-disjoint cycles with maximum score, and return it

Interior value of δ was best

δ	Average number of lives saved	Standard deviation
0	244.3	3.36
8	**248.0**	**3.32**
15	245.7	3.45
20	242.3	4.87
50	243.6	3.12
100	242.8	4.12
500	208.3	3.79

Algorithm 1 is not optimal

ABCD exist. In step 2, A disappears, and either EF or GH appear.

Optimal solution is cycle ACDB, but that is not optimal on any trajectory!

Algorithm 2

See also [Chang et al. 00]

- IDEA: Optimize the scenarios for each action separately instead of each scenario separately

- For each cycle c in G_t, score(c)=0

- Generate scenarios s_1, ..., s_m

- For each cycle c in G_t
    - For each scenario s_i
        - S = solution to offline problem on $\{G_t$-c, $s_i\}$
        - score(c)=score(c)+value(S)+value(c)

- Using the batch ILP, find a set of vertex-disjoint cycles with maximum score, and return it

Algorithm 3: Adaptation of AMSAA
[Mercier & van Hentenryck 08]

- *Global anticipatory gap (GAG)* ~ no action good across scenarios
  - This problem likely to have large GAG
  - AMSAA designed for problems with large GAG; optimal in limit

- Generate scenarios s_1, ..., s_m
- For each state σ // Construct an approximate MPD
  - if σ is a final state, then $v(\sigma)$ is offline solution in σ
  - else $v(\sigma)$ = avg value of offline solution over scenarios $\{s_1$, ..., $s_m\}$, assuming no vertex dies
- Solve the MDP using tree search starting at state G_t
- For each cycle c in G_t, score(c)=$Q(G_t,c)$
- Using the batch ILP, find a set of vertex-disjoint cycles with maximum score, and return it

Experimental setup for online tests

- Real data set: 158 pairs, 11 altruistic donors, 4086 edges, highly sensitized

- Artificial data set using [Saidman et al.] generator: 510 pairs, 25 altruistic donors, 15,400 edges

- Death rate set so 12% survive 10 years

- Dummy action (=inaction) allowed

Parameter tuning to scale to the large: Experiments uncovered interesting tradeoffs

- **Number of sample trajectories**

- **Lookahead depth (number of steps)**
  - For a given number of sample trajectories, interior optimum

- **Batch size**
  - Large => can look deep into the future (for given lookahead depth)
  - Small => finer-grained decision making

- We also tuned batch size for offline benchmark

Experimental results on trajectory-based algorithms

- Dummy action helps (not needed in Algorithm 1)
- Algorithm 2 outperforms Algorithms 1 and 3
  - Also, 3 doesn't scale
- Outperforms batch approach
- Scales to 500-600 pairs

Performance of our online algorithms
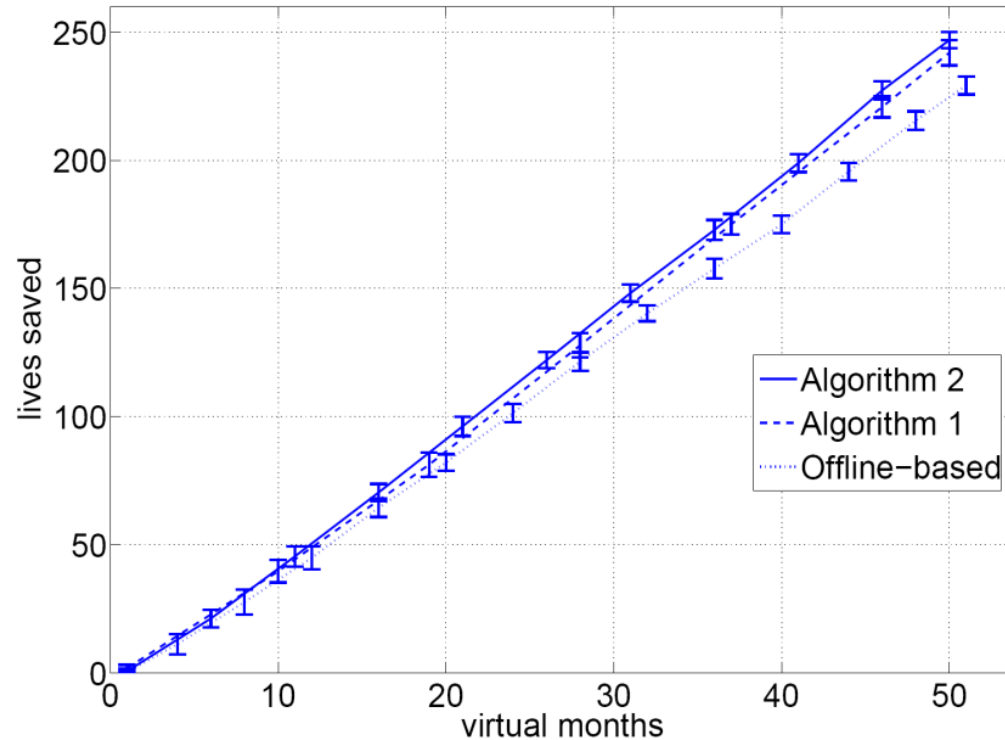
- On the real data, **outperformed batch approach by 6.5%** (std dev 1.7%)
- On the large generated data, in steady state, **outperformed batch approach by 13.0%** (std dev 2.2%)
- Pool less depleted

Family 2 of our approaches:
New approach to dynamic problems
[Dickerson, Procaccia, Sandholm *AAAI-12*]

- Idea: Learn the *potential* of each type of graph element (e.g., vertex type, edge type, cycle type, or graph type)

- Adjust ILP objective by subtracting potentials of the elements the solution uses up

- Theory on how much associating potentials to larger elements can help

Experiment with vertex potentials

Used instance generator by Saidman et al. [2006]

Expiration for pairs & altruists: 12% survive 10 years

- 4 altruistic donor ABO types + 16 patient-donor pair ABO types

- Learn vertex potentials using ParamILS (later, SMAC)
  [Hutter, Hoos, Leyton-Brown & Stützle JAIR-09]
  - Each training instance had 95 pairs and 5 altruists arriving over 25 months

Results on test set

#altruists = 0.05 * |Candidates|
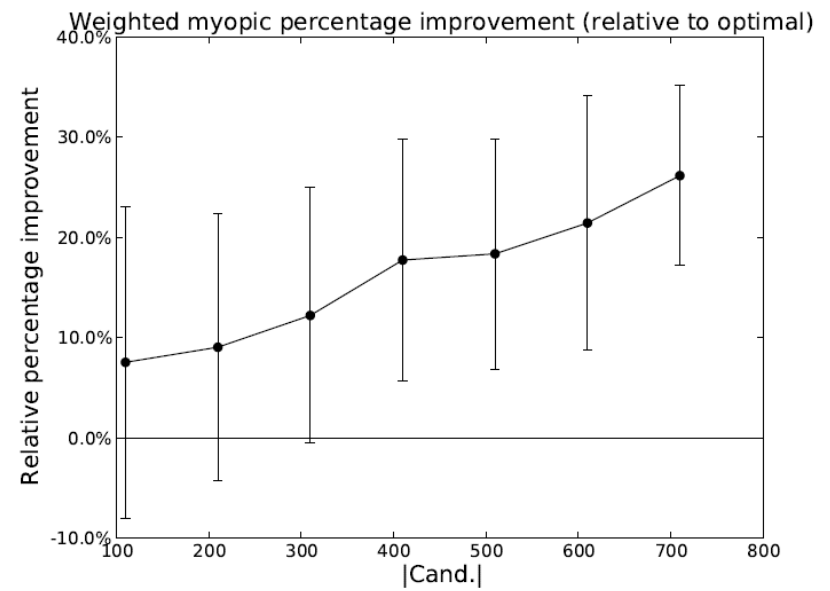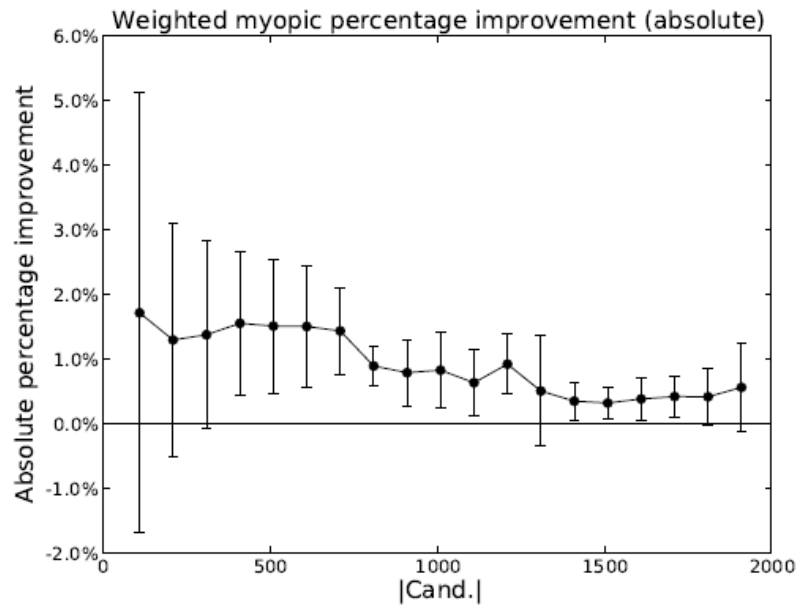
Could improve further by conditioning potentials on additional patient and donor attributes?

Tutorial outline

- Introduction & preliminaries
- Optimization models & state of the practice
- Deep dive into dimensions of kidney exchange:
  - Short-term uncertainty
  - Fairness vs economic efficiency
  - Long-term uncertainty & dynamic optimization
  - Incorporating human expert judgment in better ways
  - Incentives & mechanism design
- Other organ exchanges
- Conclusion & open research problems

The Big Problem

- What is "best"?
  - Maximize matches right now or over time?
  - Maximize transplants or matches?
  - Prioritization schemes (with fairness)?
  - …

> Want expert humans in the loop to express value judgment, but not guessing at priority points or impact of policy changes on matching results
>
> **Human-AI hybrid**

# FutureMatch: Learning to match in dynamic environments

[Dickerson & Sandholm, *AAAI-15*]
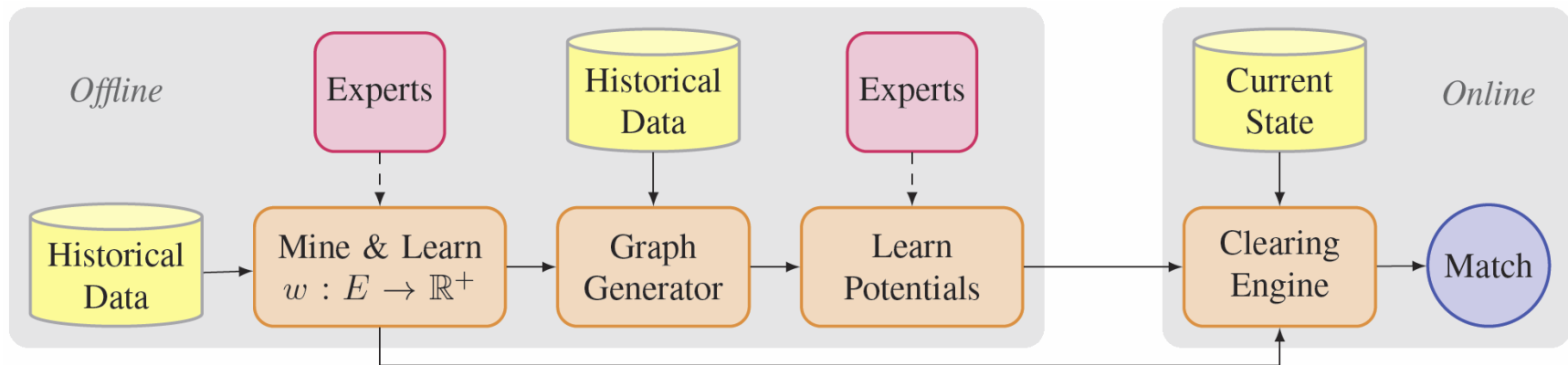
Offline (run once or periodically)

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*
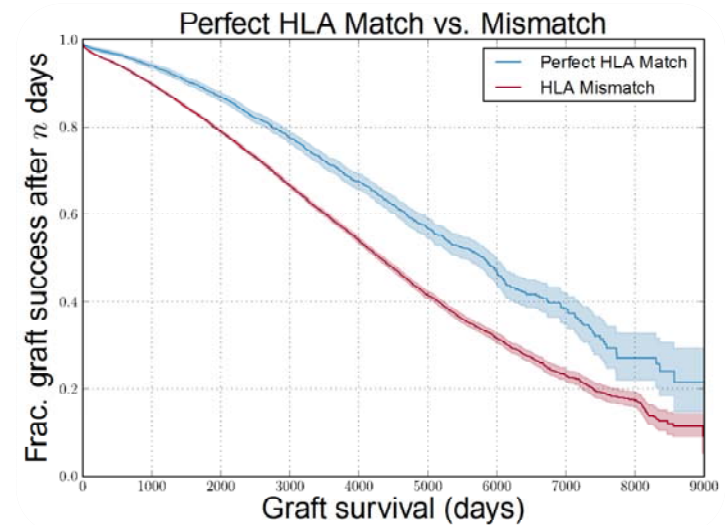4. Using this generator and a realistic exchange simulator, learn potentials for graph elements as a function of the exchange dynamics
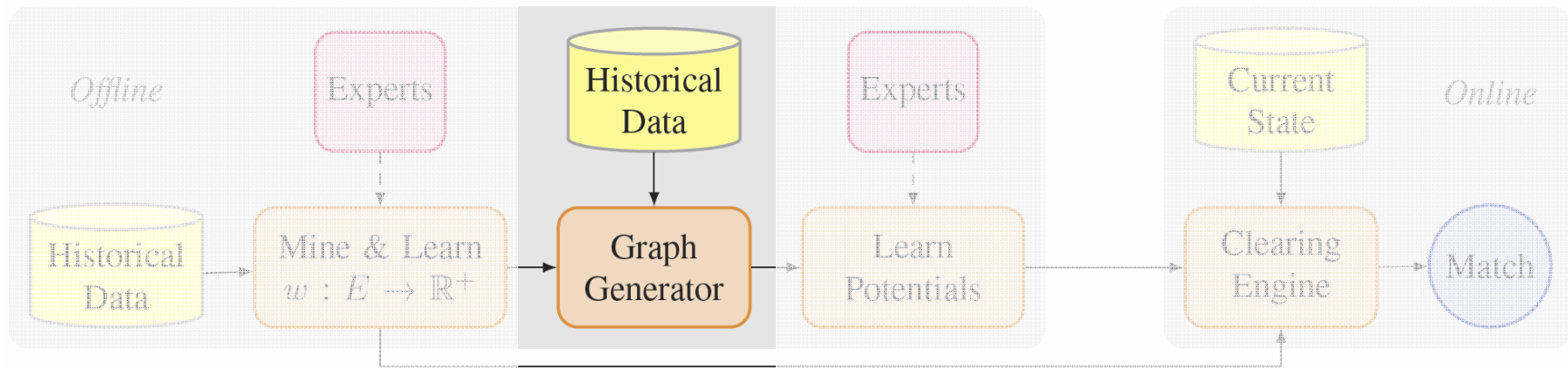
Online (run every match)

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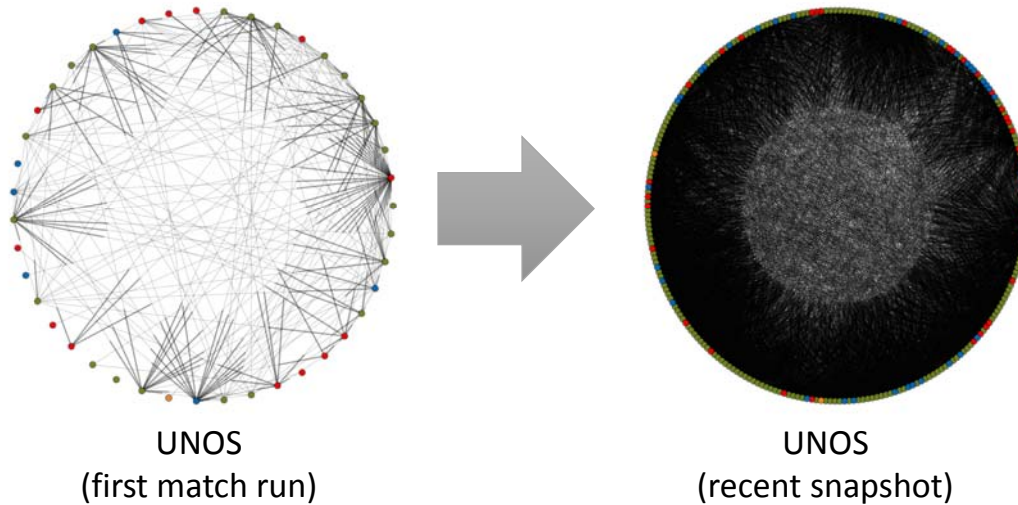2. Solve maximum weighted matching and return match
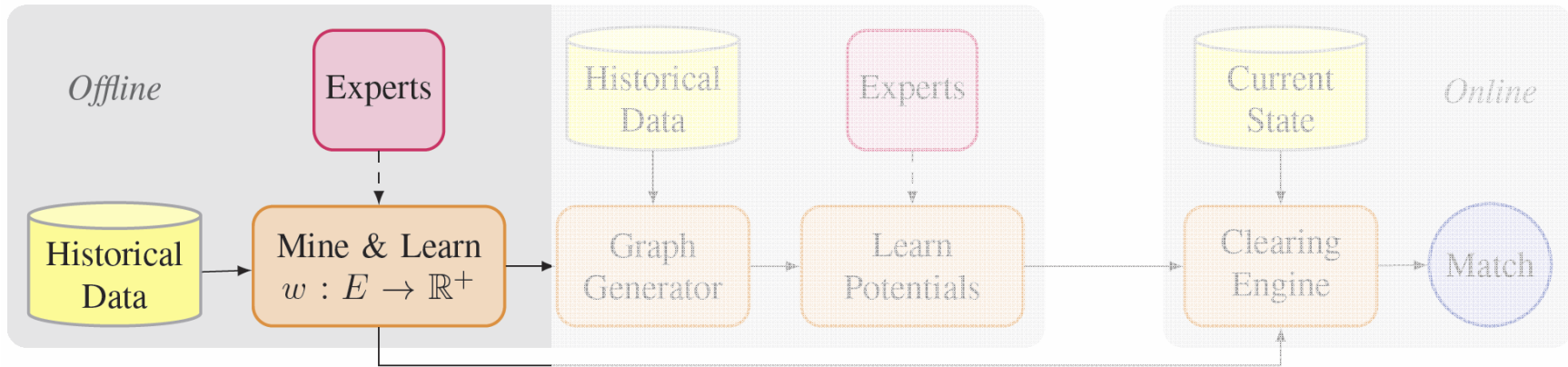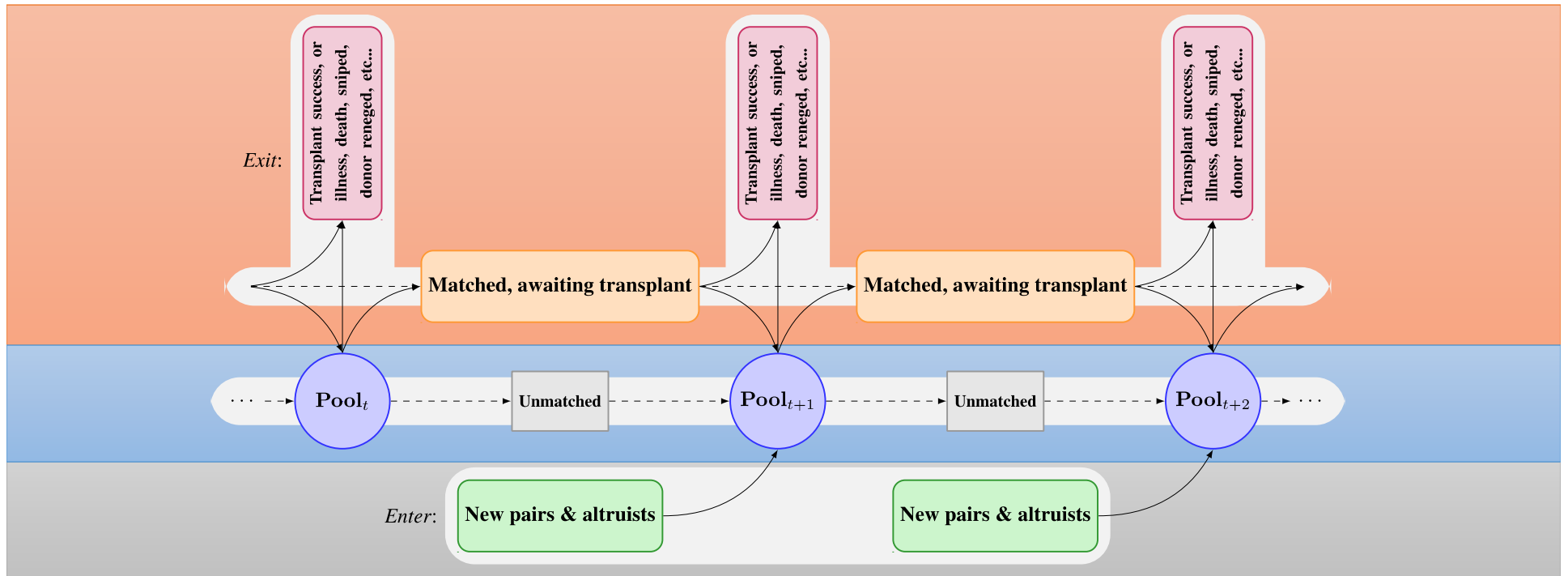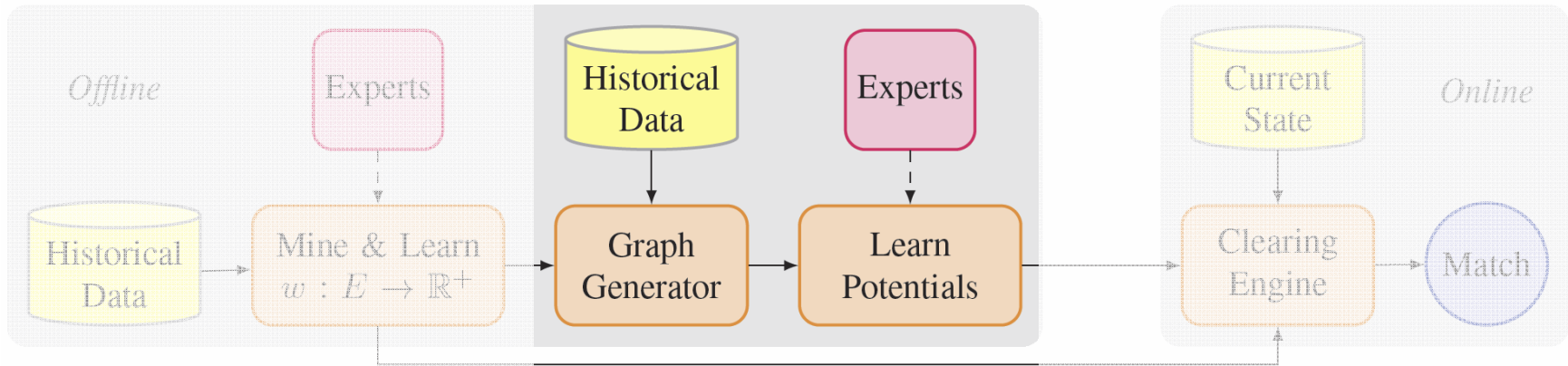
Example objective: MaxLife

- Maximize aggregate length of time donor organs last in patients …
  - … possibly subject to prioritization schemes, fairness, etc …

- Learn survival rates from all living donations since 1987
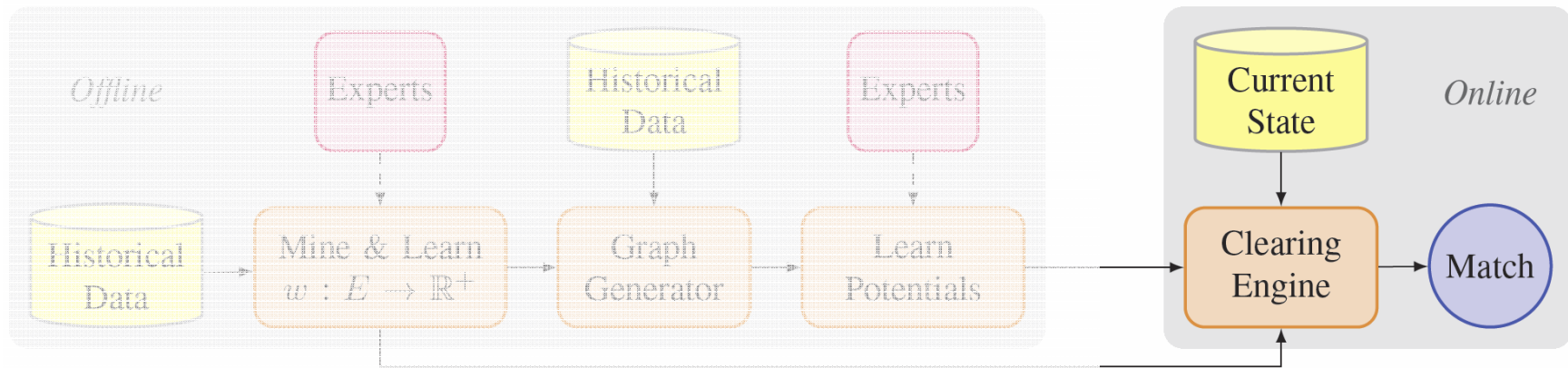 - ~75,000 transplants
- Translate to edge weight

Perfect HLA Match vs. Mismatch

- 300+ match runs with real UNOS data
- Important to use realistic distribution

UNOS
(first match run)

UNOS
(recent snapshot)

Offline

Experts

Historical
Data

Mine & Learn
$w : E \rightarrow \mathbb{R}^+$

Historical
Data

Experts

Graph
Generator

Learn
Potentials

Current
State

Online

Clearing
Engine

Match

Exit:

Transplant success, or illness, death, sniped, donor reneged, etc...

Transplant success, or illness, death, sniped, donor reneged, etc...

Transplant success, or illness, death, sniped, donor reneged, etc...

Matched, awaiting transplant

Matched, awaiting transplant

Pool_t

Unmatched

Pool_{t+1}

Unmatched

Pool_{t+2}

Enter:

New pairs & altruists

New pairs & altruists

158

Online:

- Adjust solver to take potentials into account at runtime

- E.g., $P_{\bullet\text{-}O} = 2.1$ and $P_{O\text{-}AB} = 0.1$

- 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

- Also take into account learned weight function w

**Edge weights preprocessed →
no runtime hit!**

Experimental results

- 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

- Sweet spot depends on distribution:
  - Luckily, we can generate – and learn from – realistic families of graphs!

Tutorial outline

- Introduction & preliminaries
- Optimization models & state of the practice
- Deep dive into dimensions of kidney exchange:
  - Short-term uncertainty
  - Fairness vs economic efficiency
  - Long-term uncertainty & dynamic optimization
  - Incorporating human expert judgment in better ways
  - Incentives & mechanism design
- Other organ exchanges
- Conclusion & open research problems

168

Transplant centers hide pairs and NDDs from exchange(s)

- Why do centers do this?
  - Logistical benefit
  - Money

- What fraction of locally matchable pairs/NDDs do centers hide?
  - A: 100% [Stewart, Leishman, Sleeman, Monstello, Lunsford, Maghirang, Sandholm, Gentry, Formica, Friedewald, Andreoni. 2013. American Transplant Congress]

- No mechanism design solution possible in static setting
  [Roth, Sönmez, Ünver (2007a); Ashlagi, Fischer, Kash, Procaccia, GEB-13; Ashlagi & Roth (2014)]

- Incentive-compatible, efficient, long-term-IR credit mechanism
  [Hajaj, Dickerson, Hassidim, Sandholm, Sarne, AAAI-15]
  - Matching favors centers that reveal more than their expected number of pairs/NDDs, and disfavors those who reveal fewer than that
  - Supports chains and long cycles
  - Assumes pairs and NDDs last for only one matching period

Mechanism Desiderata

(Through the lens of kidney exchange)

Individual rationality (IR)

> Will I be better off participating in the mechanism than I would be otherwise?

- Long-term IR:
  - In the long-run, a center will receive at least the same number of matches by participating

- Short-term IR:
  - At each time period, a center receives at least the same number of matches by participating

Strategy proofness

Do I have any reason to lie to the mechanism?

- In *any* state of the world …
    - time period, past performance, competitors' strategies, current private type, etc

- … a center is not worse off reporting its full private set of pairs than reporting *any* other subset

→ No reason to strategize

Efficiency

> Does the mechanism result in the absolute best possible solution?

- Efficiency:
  - Produces a maximum (i.e. max global social welfare) matching given *all* pairs, regardless of revelation

- IR-Efficiency:
  - Produces a maximum matching constrained by short-term individual rationality

The basic kidney exchange game

- Set of n transplant centers $T_n = \{t_1 \ldots t_n\}$, each with a set of incompatible pairs V_h

- Union of these individual sets is V, which induces the underlying compatibility graph

- **Want**: all centers participate, submit full set of pairs

- An allocation M is **k-maximal** if there is no allocation M' that matches all the vertices in M and also more

  - Note: k-efficient $\rightarrow$ k-maximal, but not vice versa

Individually rational? [Ashlagi & Roth 2014, and earlier]

- Vertices a_1, a_2 belong to center a,
    b_1, b_2 belong to center b
- Center a could match 2 internally
- By participating, matches only 1 of its own
- Entire exchange matches 3 (otherwise only 2)

It can get much worse

Theorem: For k>2, there exists *G* s.t. no IR *k*-maximal mechanism matches more than 1/(*k*-1)-fraction of those matched by *k*-efficient allocation

- Bound is tight

- All but one of *a*'s vertices is part of another length *k* exchange (from different agents)

- *k*-maximal and IR if *a* matches his *k* vertices (but then nobody else matches, so *k* total)

- *k*-efficient to match (k-1)*k
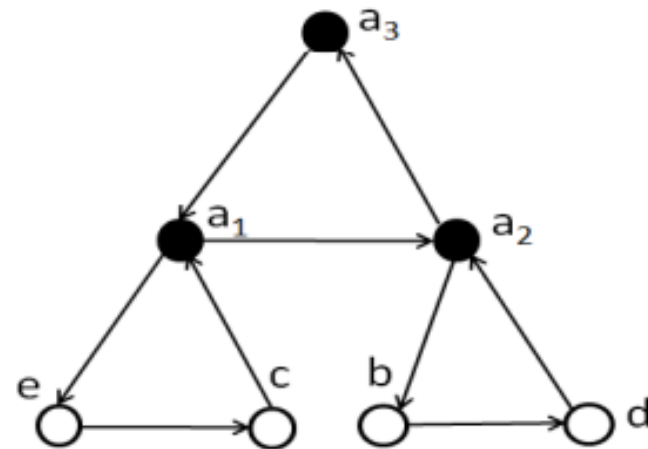
Example: k=3

Restriction #1

Theorem: For all k and all compatibility graphs, there exists an IR k-maximal allocation

- Proof sketch: construct k-efficient allocation for each specific hospital's pool V_h
- Repeatedly search for larger cardinality matching in an entire pool that keeps all already-matched vertices matched (using augmenting matching algorithm from Edmonds)
- Once exhausted, done

Restriction #2

Theorem: For k=2, there exists an IR 2-efficient allocation in every compatibility graph

- Idea: Every 2-maximal allocation is also 2-efficient
  - collection of sets of matched vertices form a matroid
    - special independence cases with k=2, also this is a PTIME problem with the $|V|^3$ bipartite augmenting paths matching algorithm
- By Restriction #1, 2-maximal IR always exists → this 2-efficient IR always exists
-

A Dynamic, Credit-Based Mechanism

(That is strategy proof and efficient, if some assumptions hold.)

Dynamic, Credit-Based Mechanism

[Hajaj et al. *AAAI-2015*]

- Repeated game

- Centers are risk neutral, self interested

- Transplant centers have (private) sets of pairs:
  - Maximum capacity of $2k_i$
  - General arrival distribution, mean rate is k_i
  - Exist for one time period

- Centers reveal subset of their pairs at each time period, can match others internally

Credits

- Clearinghouse maintains a **credit balance** c_i for each transplant center over time

- High level idea:
  - **REDUCE** c_i: center i reveals fewer than expected
  - **INCREASE** c_i: center i reveals more than expected

  - **REDUCE** c_i: mechanism tiebreaks in center i's favor
  - **INCREASE** c_i: mechanism tiebreaks against center I

  Also remove centers who misbehave "too much."

Credits now → matches in the future

The dynamic mechanism

1. Initial credit update
   - Centers reveal pairs
   - Mechanism updates credits according to k_i

2. Compute maximum global matching
   - Gives the utility U_g of *a* max matching

3. Selection of a final matching
   - Constrained to those matchings of utility U_g
   - Take c_i into account to (dis)favor utility given by matching to a specific center *i*
   - Update c_i based on this round's (dis)favoring

4. Removal phase if center is negative for "too long"

Theoretical guarantees

Theorem: No mechanism that supports cycles and chains can be both long-term IR and efficient

Theorem: Under reasonable assumptions, the prior mechanism is both long-term IR and efficient

Experiments on Real Data

(Uses data from UNOS exchange, first ~100 match runs)

UNOS Generator, Efficient

UNOS Generator, IR-Efficient
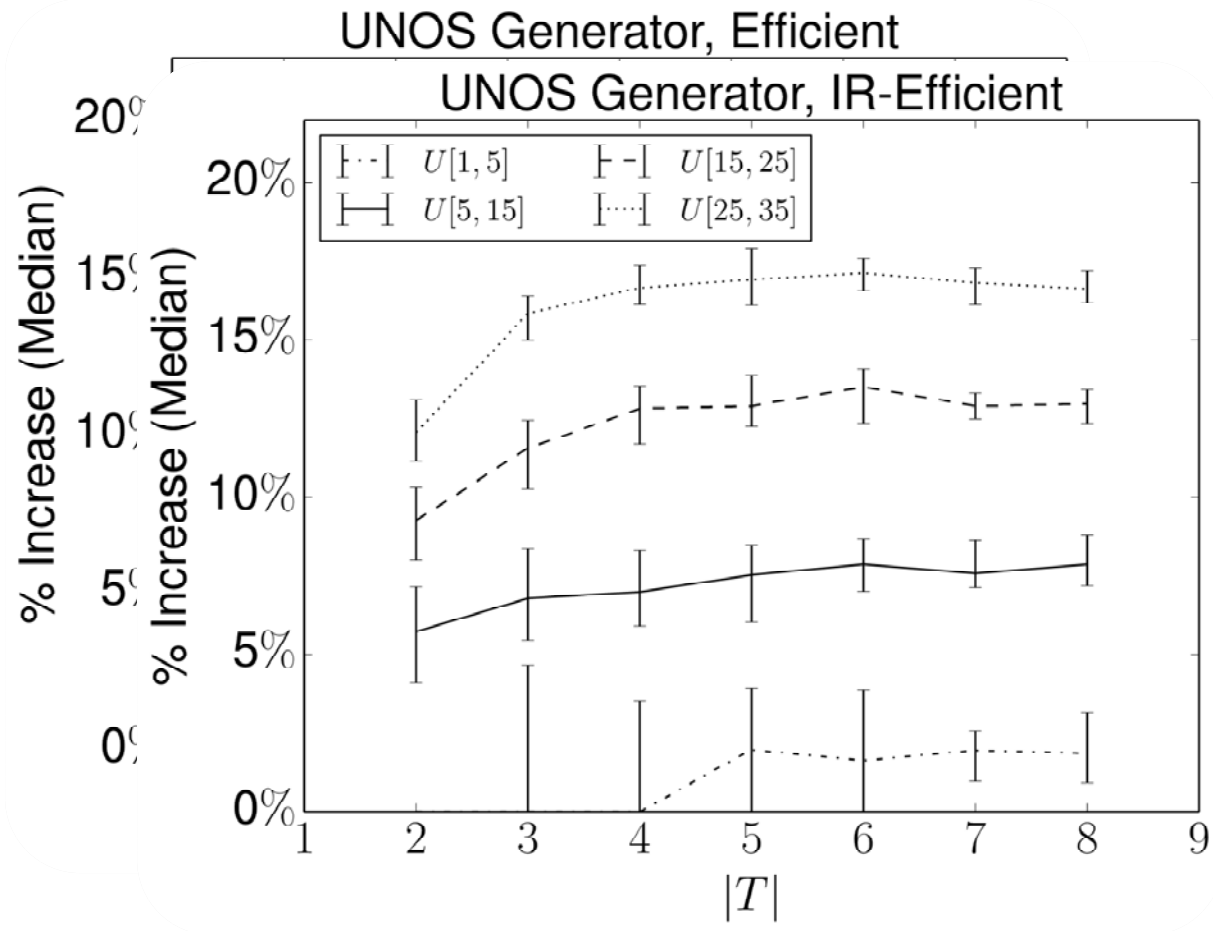
Experiments on Simulated Data *(Uses data generated according to [Saidman et al. 2006])*
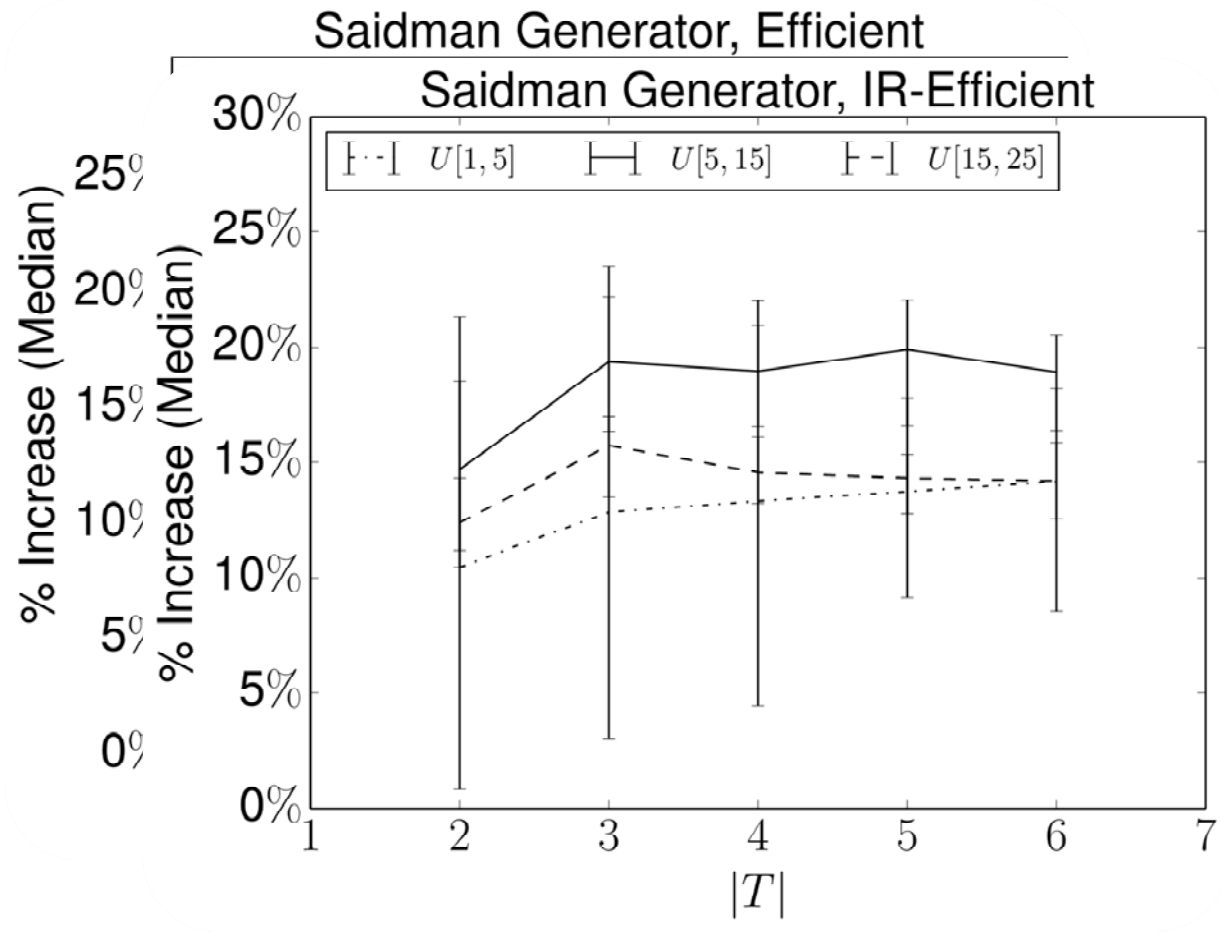
This is still a very open problem!

- All models are still limited:
  - Agency attributed only to transplant centers, not
    - Patient-donor pairs (generally)
    - Central bodies like a government
  - Money?
  - Most results in the static setting
  - Determinism
  - Unreasonable knowledge of agents (e.g., avg. entry)
- Clear policy implications

Competing dynamic matching markets

[Das et al. *AMMA-15*]

- Dynamic matching markets are typically modeled in isolation, with each agent entering a single market.

- Real-world applications (e.g kidney exchange) often involve multiple matching platforms drawing from overlapping pools. Questions:
  - Whether competing platforms increase **global loss** relative to a single centralized matching platform.
  - How does one platform's **matching policy** affect global loss in a multi-platform setting?
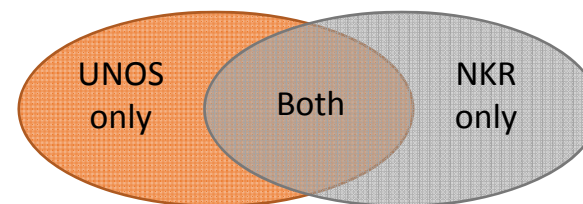
Kidney exchange is dynamic

- Patient-donor pairs (agents) arrive gradually over time
  - Stay in the market to find a compatible pair
  - May leave if the patient's condition deteriorates to the point where kidney transplants become infeasible
- Only considering **undirected 2-matching** so far

Planner / Clearinghouse platform

- Minimizes the number of agents who *perish* (leave the exchange without finding a match)

- Knows agent's expiration time

- Has only probabilistic knowledge about future incoming agents

- Selects a subset of acceptable transactions at any point in time
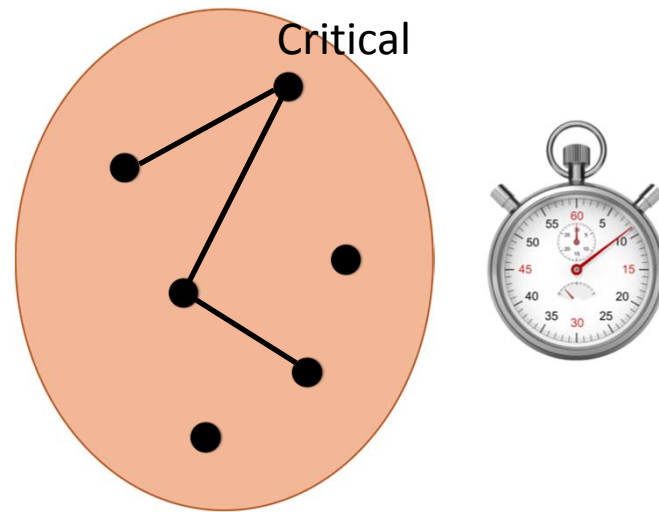
Greedy and patient exchanges
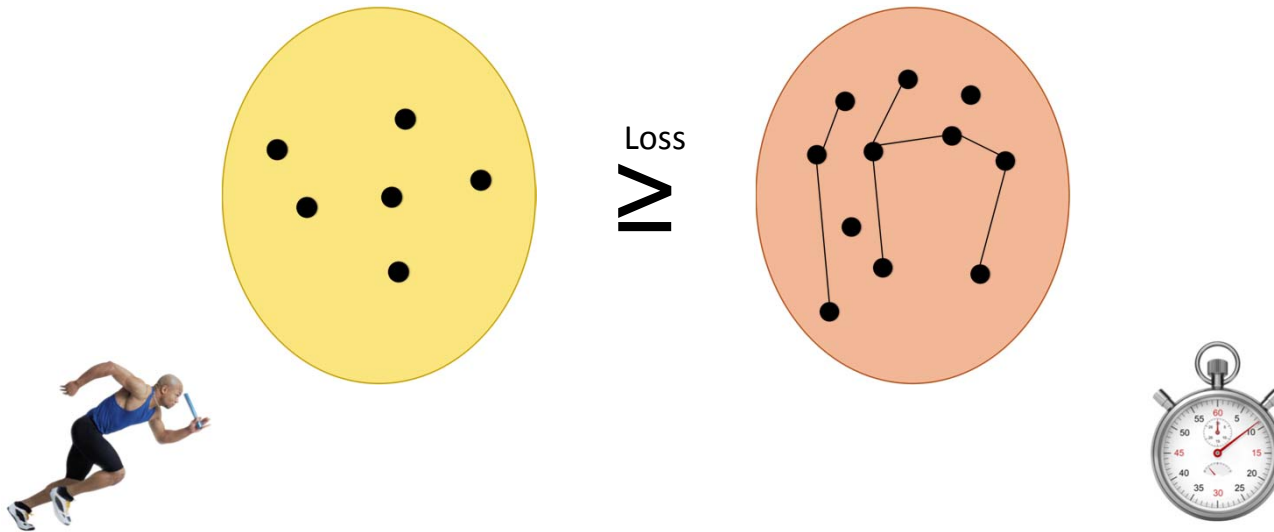
[Akbarpour et al., *EC-14*]
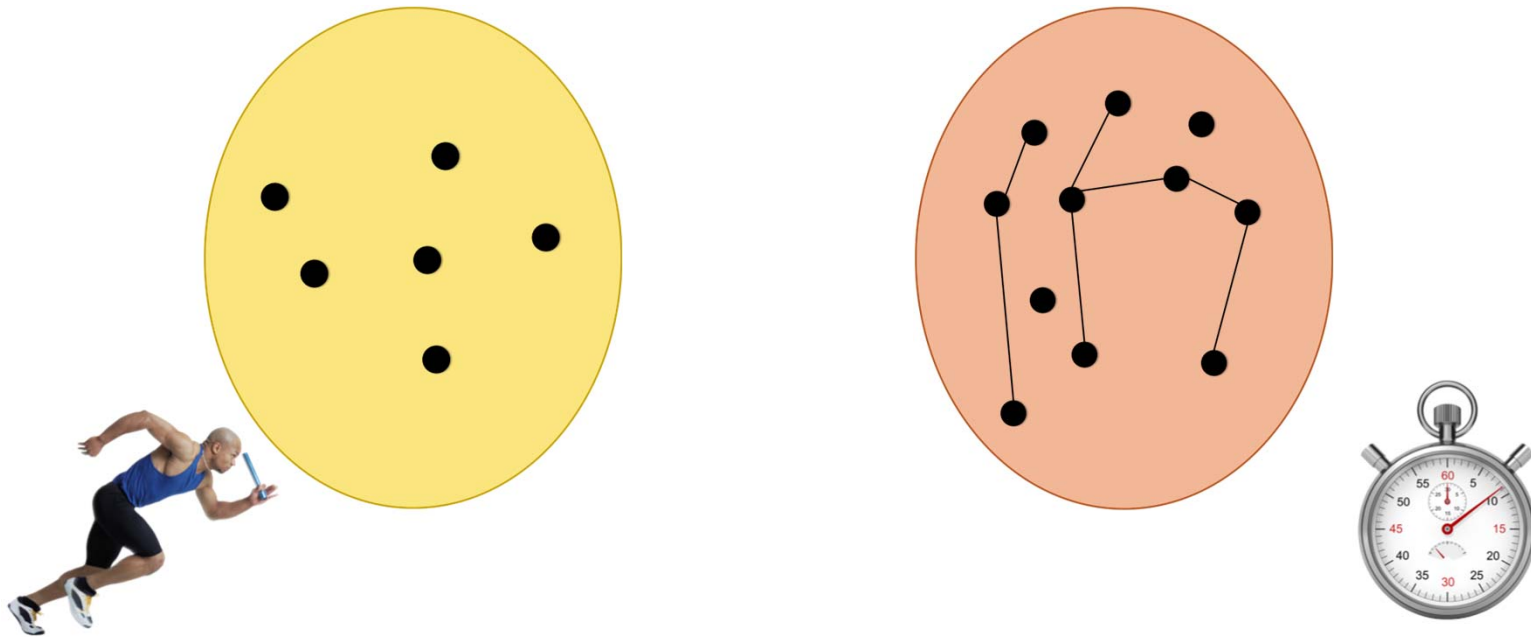
Greedy exchange

Patient exchange

Critical
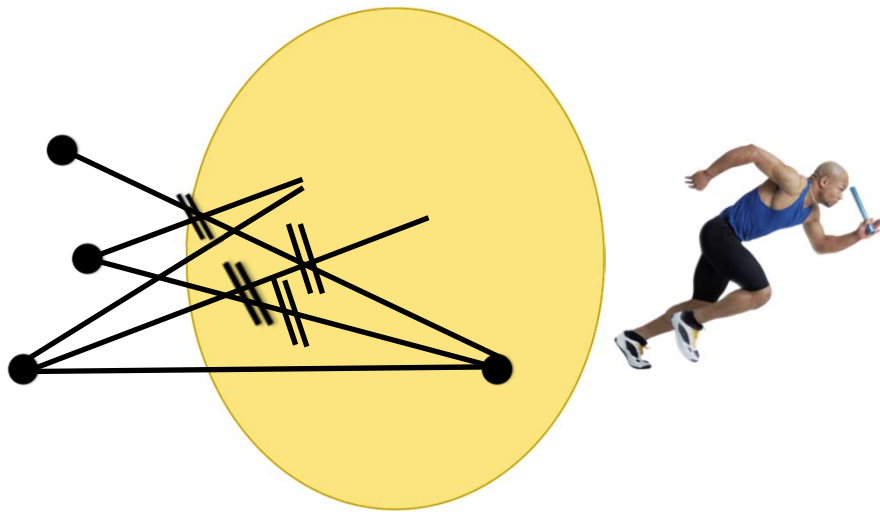
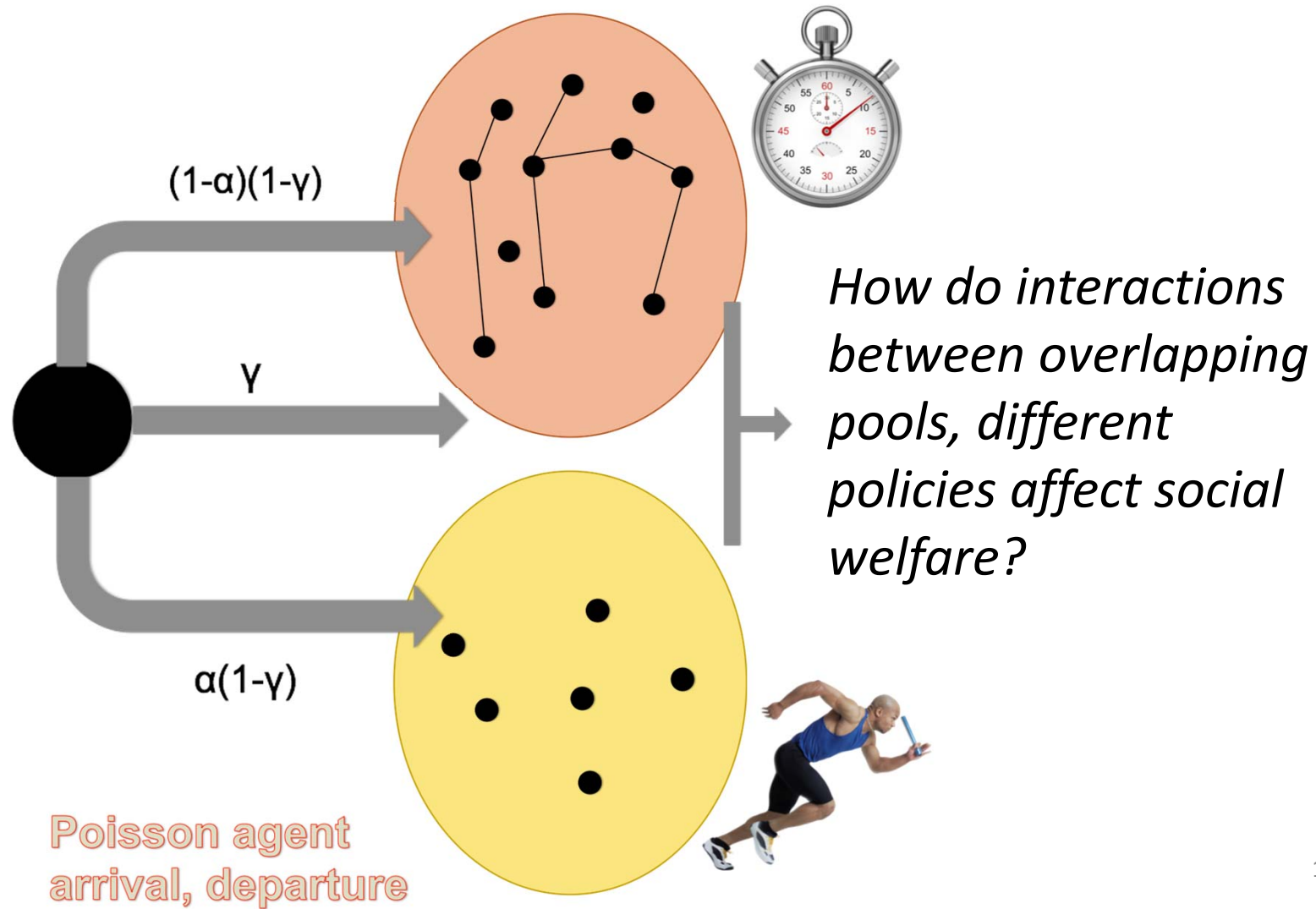# Greedy and patient exchanges

$$\text{Loss} = \frac{\text{Expected } \# \text{ of perished agents}}{\text{Expected } \# \text{ of total agents entering market}}$$

Loss

$\geq$

[Akbarpour et al., *EC-14*]

195

Competing exchanges

[Das et al. *AMMA-15*]

$(1-\alpha)(1-\gamma)$

γ

$\alpha(1-\gamma)$

How do interactions between overlapping pools, different policies affect social welfare?

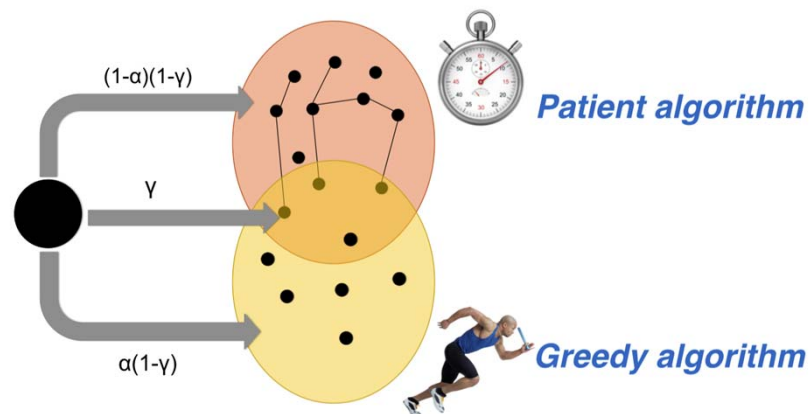Poisson agent arrival, departure

196
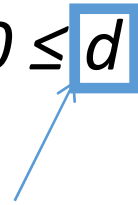
Model details

- Agents (patient-donor pairs) arrive at the market according to a Poisson process, with rate parameter $m \geq 1$

- The sojourn of an agent is drawn from an exponential distribution, with rate parameter $\lambda = 1$

- Pr(acceptable transaction) = $d/m,\ 0 \leq \boxed{d} \leq m$

Average degree of node in graph

Model details

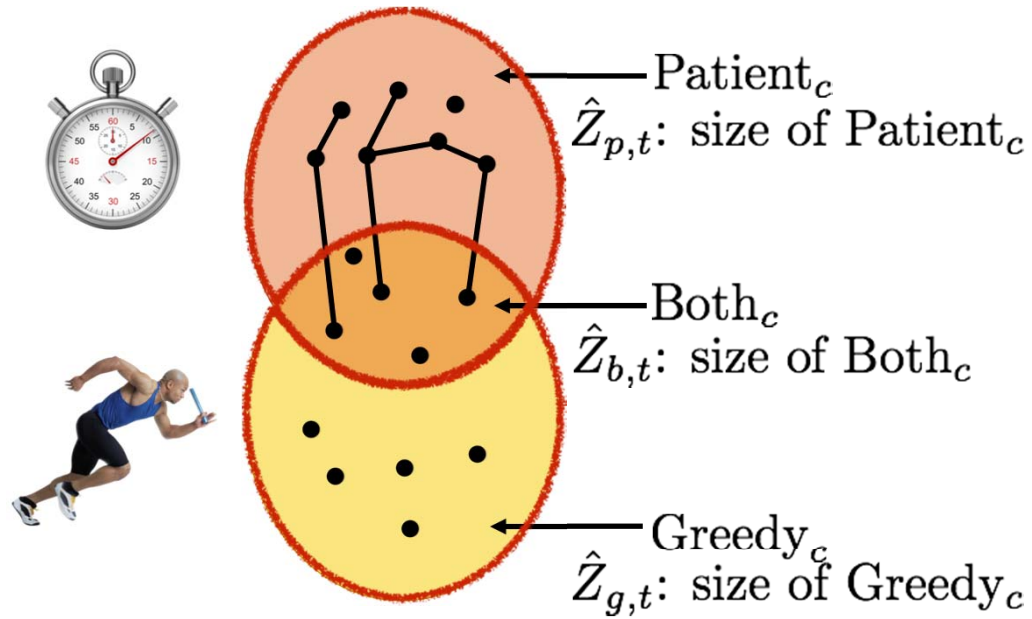

$\Pr(Both \text{ exchanges}) = \gamma$

$\Pr(\text{Greedy exchange alone}) = (1 - \gamma)\alpha$

$\Pr(\text{Patient exchange alone}) = (1 - \gamma)(1 - \alpha)$

Three pools

Proof sketch - bound on overall loss

Part of parameter space:

$$\text{Greedy}_c \geq \text{Both}_c \Leftrightarrow (1-\gamma)\alpha \geq \gamma$$

If $T, m \to \infty, m > 10d$, we can get

$$\mathbf{L}(\text{Greedy}_c) \geq \frac{\alpha(1-\gamma)}{1+3d}$$

$$\mathbf{L}(\text{Competing}) \geq \mathbf{L}(\text{Greedy}_c) \geq \frac{\alpha(1-\gamma)}{1+3d}$$
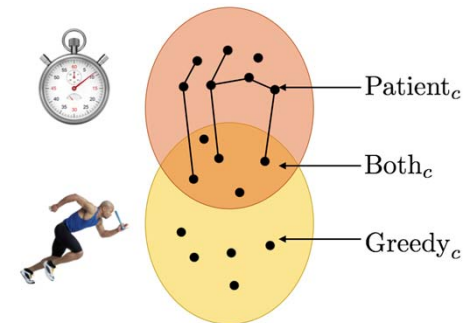
Arrival: $m \geq 1$
Sojourn: $\lambda = 1$
$\Pr(\text{edge}) = d/m$

$\Pr(Both) = \gamma$
$\Pr(\text{Greedy}_c) = (1-\gamma)\alpha$
$\Pr(\text{Patient}_c) = (1-\gamma)(1-\alpha)$

Patient$_c$
Both$_c$
Greedy$_c$

Proof sketch - bound on overall loss

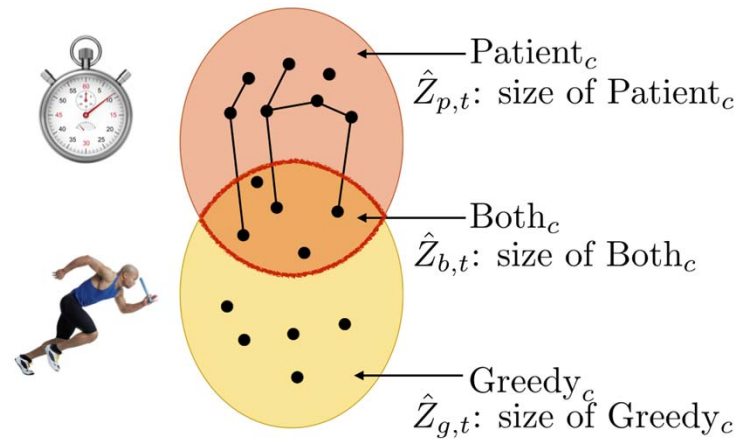Comparing to Akbarpour et al,

$$\mathbf{L}(Individual \text{ Patient}) < \tfrac{1}{2}e^{-d/2}$$

THEOREM 1. Assume $\gamma \leq 0.5$, $m > 10d$, and $\alpha(1 - \gamma) \geq \max\{\gamma, \tfrac{1}{2}e^{-d/2}(1 + 3d)\}$. Then, as $m \to \infty$ and $T \to \infty$, almost surely
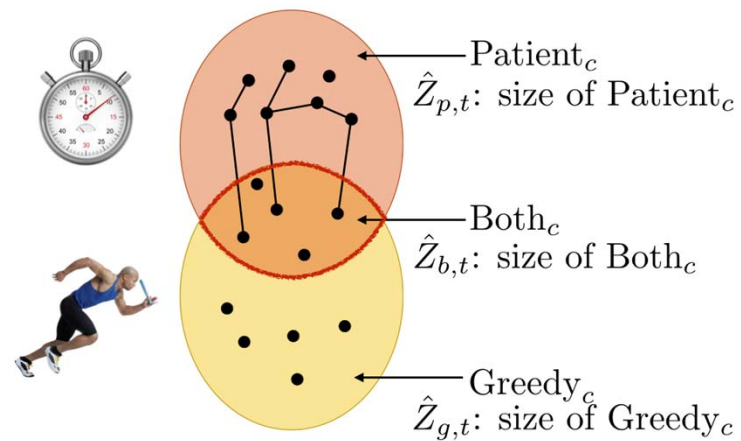
$$\mathbf{L}(\text{Competing}) > \mathbf{L}(Individual \text{ Patient})$$

Expression for overall loss

$$\mathbf{L}(\text{Competing}) = \frac{1}{mT} \int_0^T \mathbb{E}\Big[\hat{Z}_{p,t}(1 - d/m)^{\hat{Z}_{p,t} + \hat{Z}_{b,t} - 1}$$

$$+ \hat{Z}_{b,t}(1 - d/m)^{\hat{Z}_{p,t}} + \hat{Z}_{g,t}\Big]dt$$

Expression for overall loss

$Patient_c$
$\hat{Z}_{p,t}$: size of $Patient_c$

$Both_c$
$\hat{Z}_{b,t}$: size of $Both_c$

$Greedy_c$
$\hat{Z}_{g,t}$: size of $Greedy_c$

$$\hat{\xi}_g, \hat{\xi}_p, \hat{\xi}_b := \mathbb{E}_{\hat{Z}_. \sim \hat{\pi}_.}[\hat{Z}_.]$$

$$\mathbf{L}(\text{Competing}) \approx \frac{\hat{\xi}_g + \hat{\xi}_p(1-d/m)^{\hat{\xi}_p + \hat{\xi}_b - 1} + \hat{\xi}_b(1-d/m)^{\hat{\xi}_p}}{m}$$

Simulation setup

- Agents (patient-donor pairs) arrive at the market according to a Poisson process, with rate parameter *m = 100*

- The sojourn of an agent is drawn from an exponential distribution, with rate parameter λ = *1*

- Pr(acceptable transaction) = 0.02

- $\hat{\xi}_g, \hat{\xi}_p,$ and $\hat{\xi}_b$ can be approximated well using Monte Carlo simulations

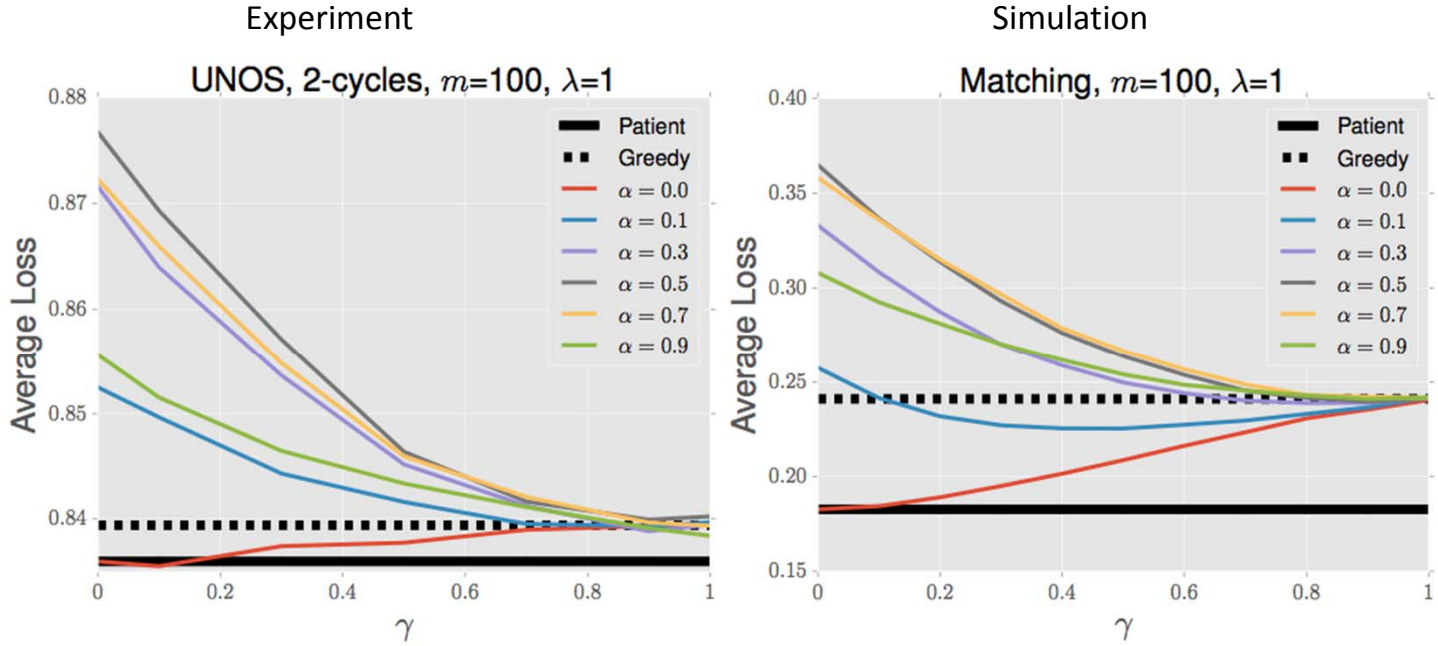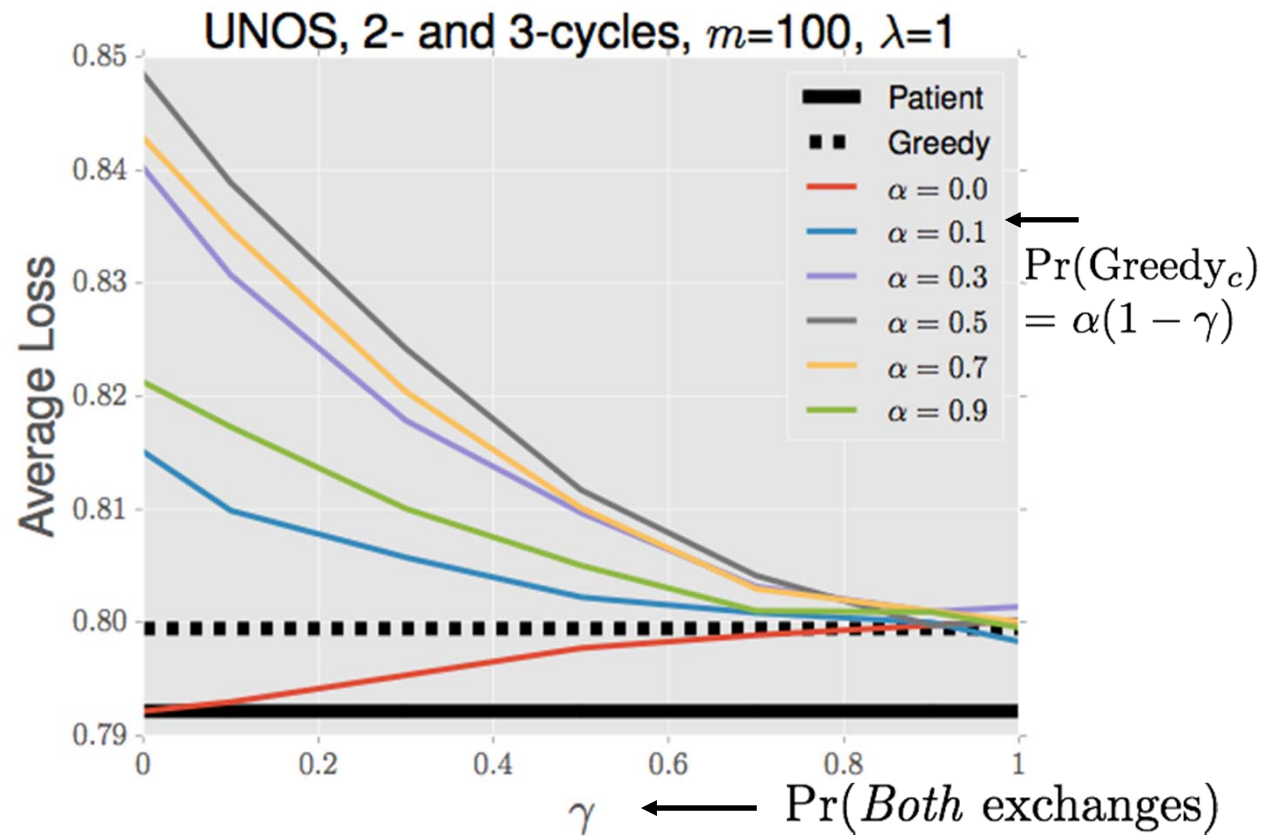# Simulation results
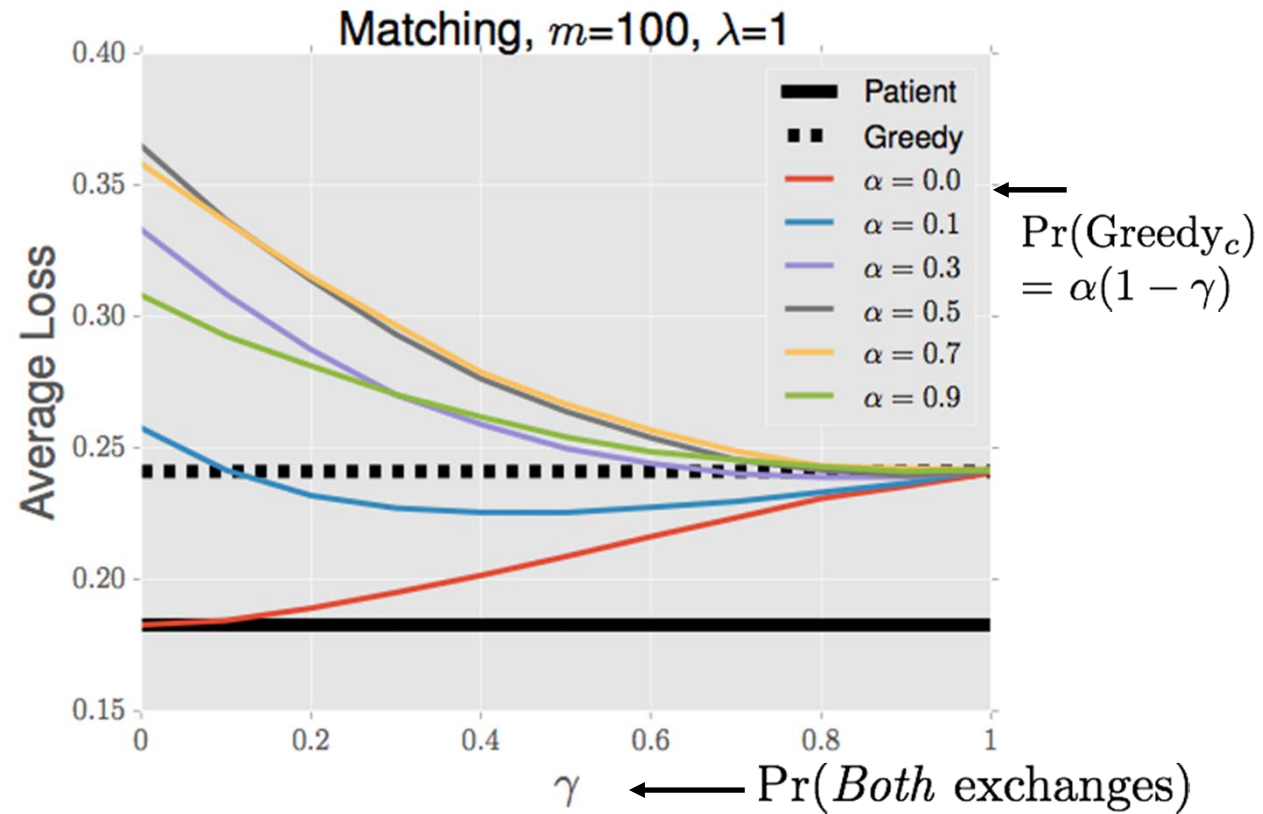
Kidney exchange experiments

- Edges are determined by medical characteristics (Pr(acceptable transaction) ≠ d/m)
  - UNOS (Real patient profiles)
  - SAIDMAN (Simulated patient profiles)
- Can incorporate longer cycles, chains, etc.
  - We consider 2- and 3-cycles

Experimental results

Experimental results

Problem still very open, very relevant!

- Relax assumptions:
  - agents select subset of platforms to enter
  - platforms select matching policy
  - 2- vs 3-cycles, chains?

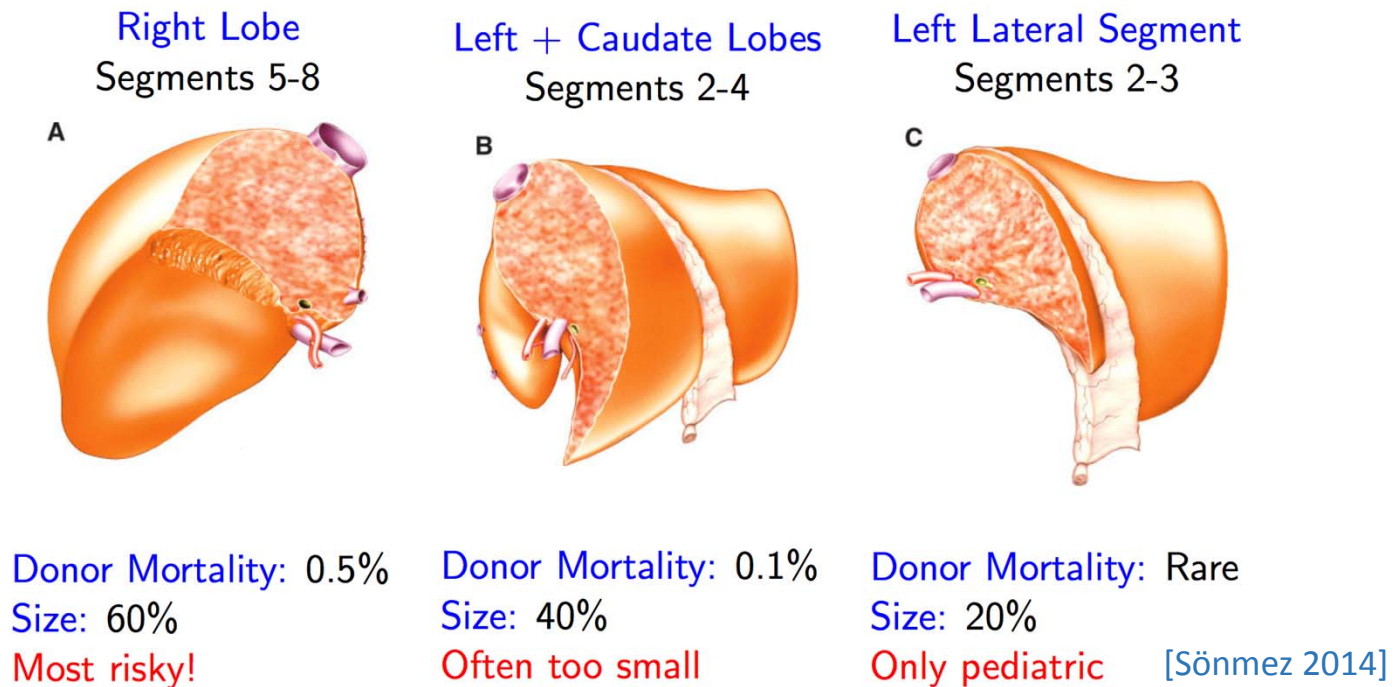- Bounds on loss: generalize, tighten, $n > 2$ platforms

Tutorial outline

- Introduction & preliminaries
- Optimization models & state of the practice
- Deep dive into dimensions of kidney exchange:
  - Short-term uncertainty
  - Fairness vs economic efficiency
  - Long-term uncertainty & dynamic optimization
  - Incorporating human expert judgment in better ways
  - Incentives & mechanism design
- Other organ exchanges
- Conclusion & open research problems

Moving beyond kidneys: Livers

[Ergin, Sönmez, Ünver *w.p. 2015*]

- Similar matching problem (mathematically)

| Right Lobe	Left + Caudate Lobes	Left Lateral Segment
Segments 5-8	Segments 2-4	Segments 2-3
A	B	C
Donor Mortality: 0.5%	Donor Mortality: 0.1%	Donor Mortality: Rare
Size: 60%	Size: 40%	Size: 20%
Most risky!	Often too small	Only pediatric [Sönmez 2014]

- Right lobe is biggest but riskiest; exchange may reduce right lobe usage and increase transplants
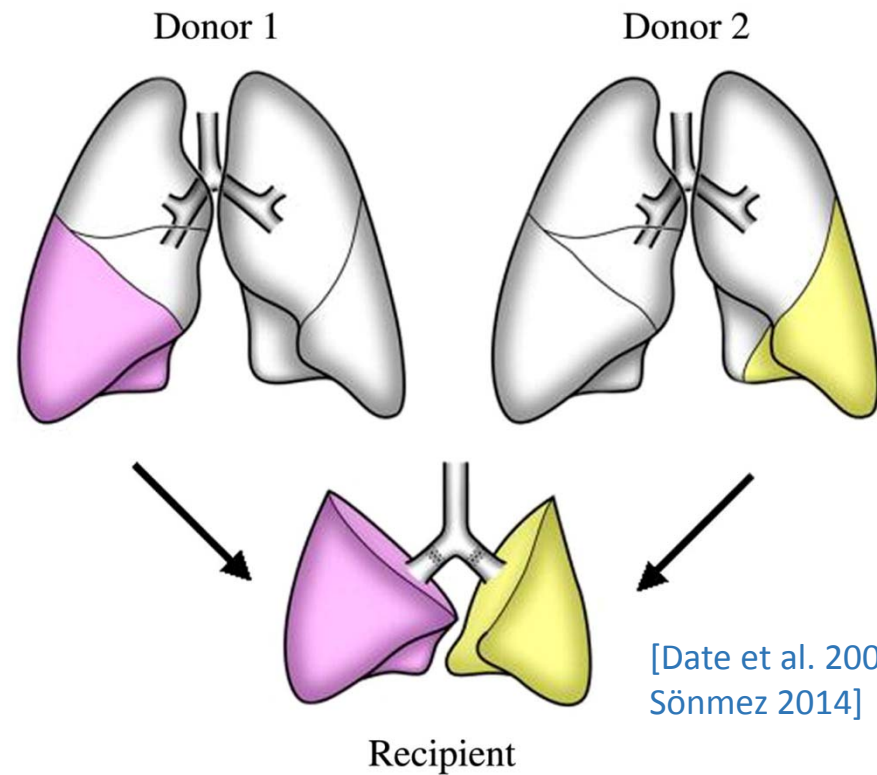
Moving beyond kidneys: Lungs

[Ergin, Sönmez, Ünver *w.p. 2014*]

- Fundamentally different matching problem
  - **Two** donors needed
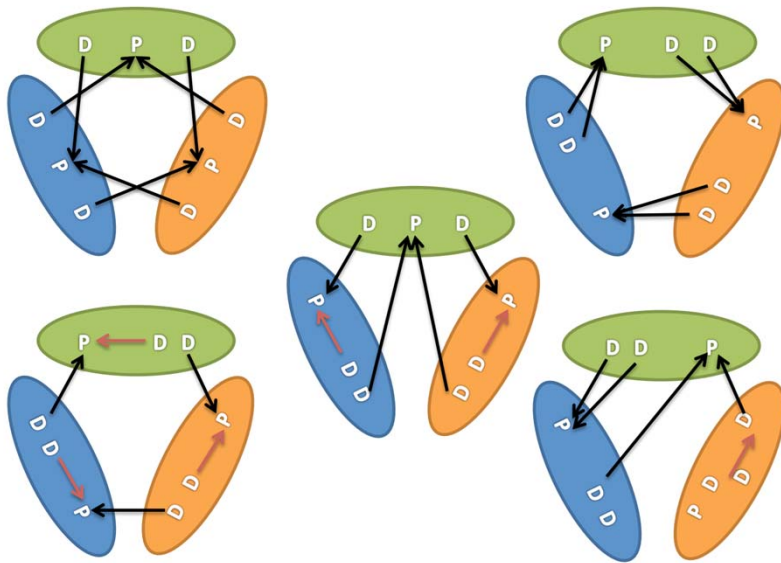
3-way lung exchange configurations

(Compare to the single configuration for a "3-cycle" in kidney exchange.)

Donor 1   Donor 2

Recipient

[Date et al. 2005; Sönmez 2014]

Mechanism design: Lung exchange [Luo & Tang *IJCAI-15*]

Theorem: Even the problem of finding a non-empty feasible swap is NP-hard
(Reduction from 3D-Matching)

They give a Pareto-efficient, IR, IC mechanism for the static setting
(where the agents are patient-donor-donor triples, not the transplant centers)

Moving beyond a single organ

[Dickerson Sandholm *AAAI-14, JAIR-2016*]

- 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
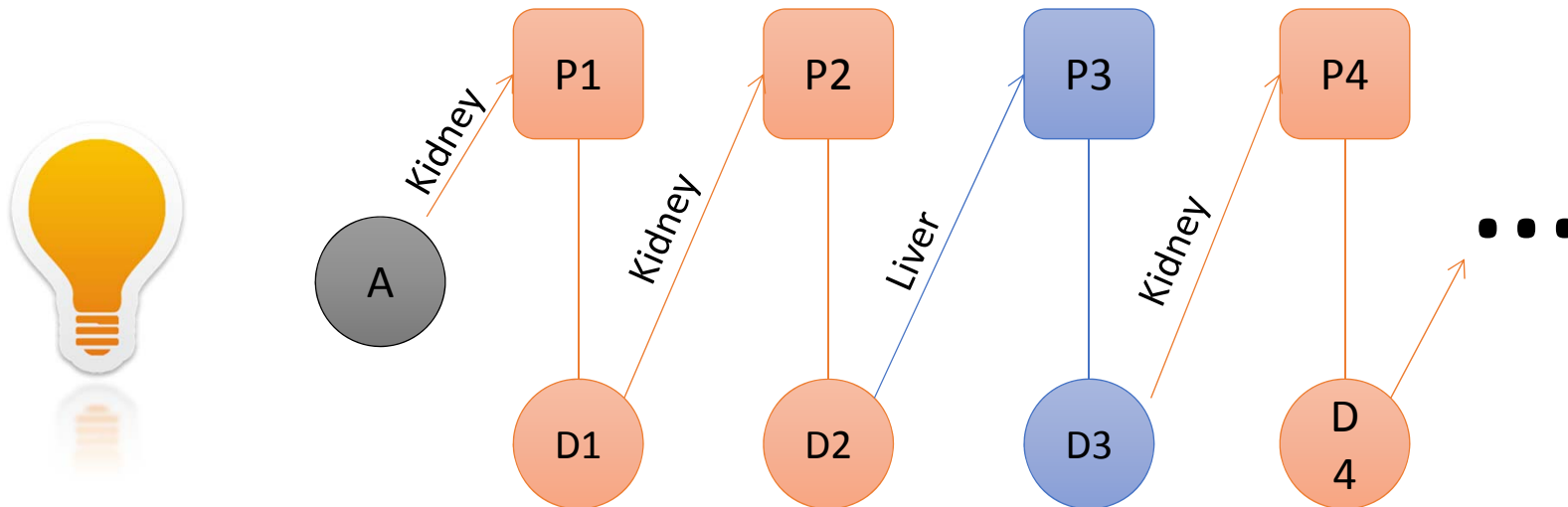
Would this help?

- Theory: adapted Erdős-Rényi models

- Dense model [Saidman et al. 2006]
  - Constant probability of edge existing
  - Less useful in practice [Ashlagi et al. 2012, Ashlagi Jaillet Manshadi 2013]
    [Dickerson Procaccia Sandholm 2013, 2014]

- Sparse model [Ashlagi et al. 2012]
  - $1-\lambda$ fraction is *highly-sensitized* ($p_H = c/n$)
  - λ fraction is *lowly-sensitized* ($p_L > 0$, constant)

- Not all kidney donors want to give livers
  - Constant probability $p_{K \to L} > 0$

Sparse graph, many altruists

- n_K kidney pairs in graph D_K

- $n_L = \gamma n_K$ liver pairs in graph D_L

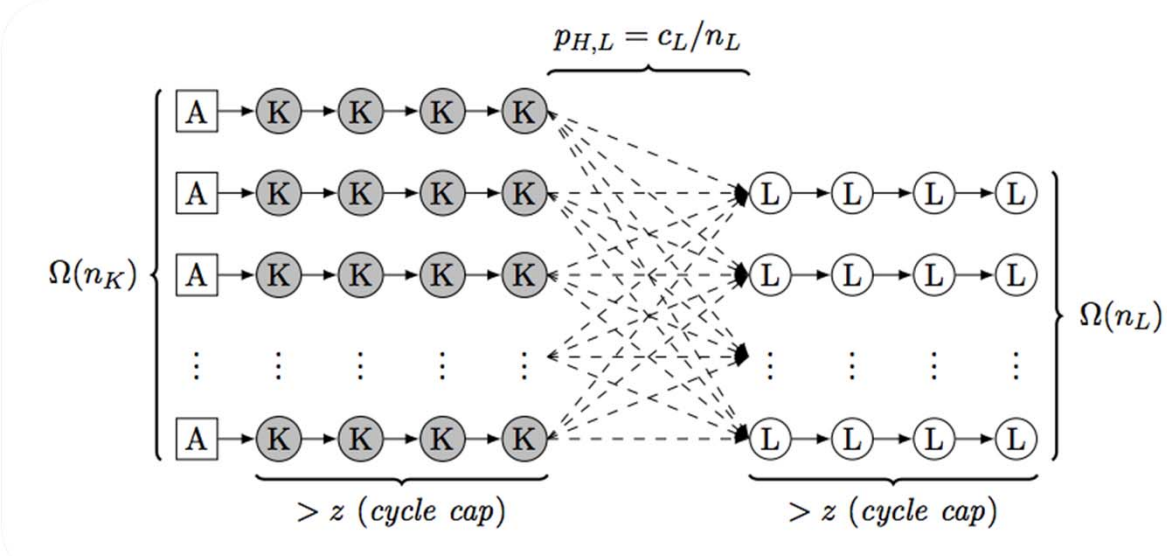- Number of altruists $t(n_K)$

- 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
  - $\to$ linear number of D_K chains extended into D_L
  - $\to$ linear number of previously unmatched D_L vertices matched

217

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!
- λ 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 = 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 λ (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
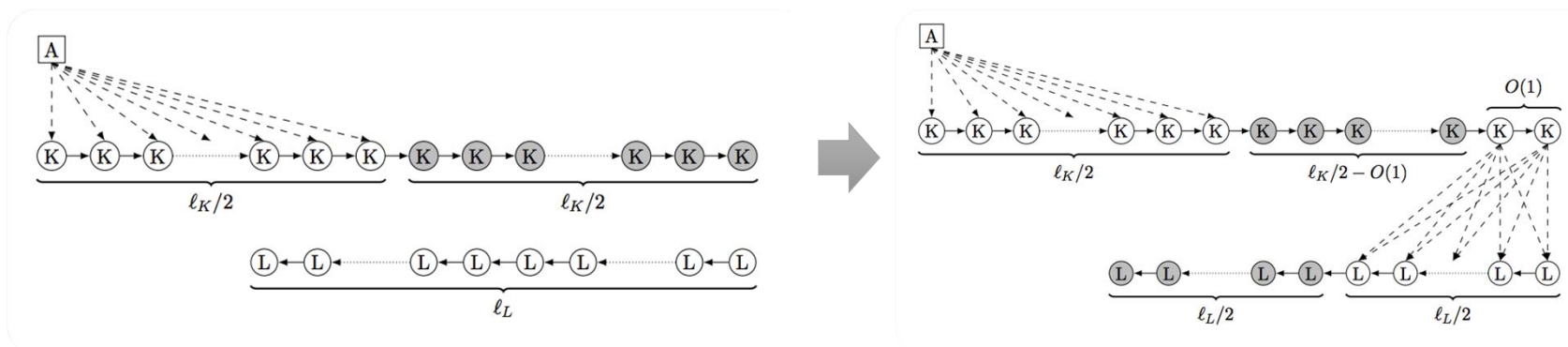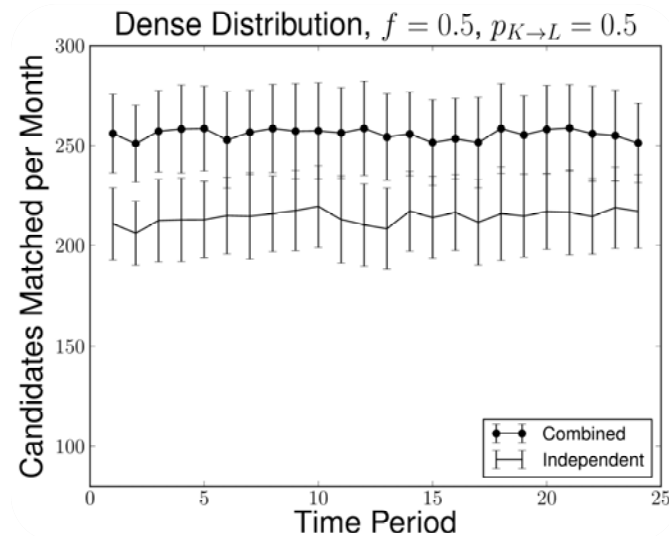
Dense graph, many altruists
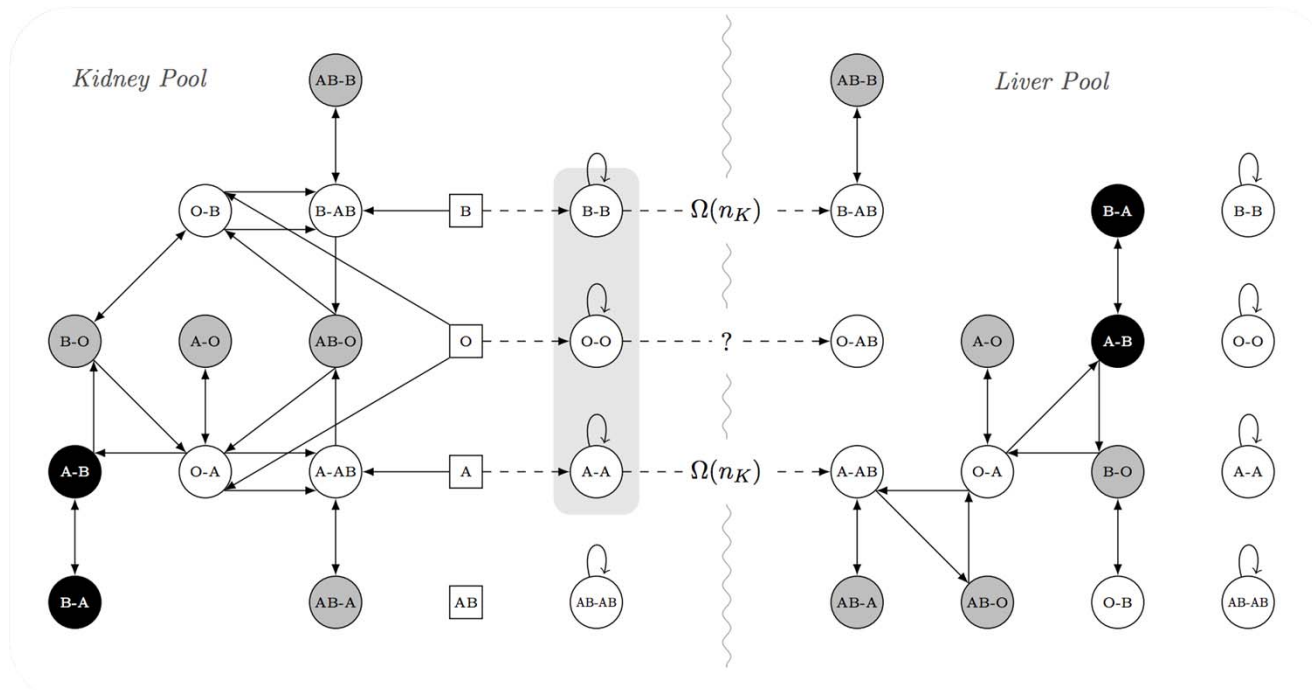
Take efficient matching on the kidney exchange graph alone, extend a linear number of chains into the leftover pairs after an efficient matching in the liver exchange alone.

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?

Dense Distribution, $f = 0.5$, $p_{K \to L} = 0.5$

Candidates Matched per Month

Time Period

Combined
Independent

Tutorial outline

- Introduction & preliminaries
- Optimization models & state of the practice
- Deep dive into dimensions of kidney exchange:
  - Short-term uncertainty
  - Fairness vs economic efficiency
  - Long-term uncertainty & dynamic optimization
  - Incorporating human expert judgment in better ways
  - Incentives & mechanism design
- Other organ exchanges
- Conclusion & open research problems

Conclusions

- Kidney exchanges are a broadly fielded success for AI

- Powered by scalable, flexible batch solvers

- Open chains are powerful

- Real kidney exchanges have cycles, chains, weights, arrivals/departures, edge failures, fairness considerations, human value judgments, …

- Dynamic problem
  - General-purpose trajectory-based online algorithms
  - General purpose idea of how to use "potentials" to capture the future into batch optimization
  - Both leverage distributional information and our offline algorithm
  - Both outperform batch-based approach

- Failure-aware probabilistic matching

- Fairness

- Futurematch: learning to do generalized matching in complex settings

- Centers hide pairs; impossibility results for static settings, but "credit" mechanisms work

- Liver lobes, multi-organ, lung parts, …

Future research

- Work on real problem: 2- and 3-cycles, chains, dynamics, edge failures
- We recently open-sourced the most realistic organ-exchange simulator
- Still lots to be done
  - Even faster algorithms, esp. with chains
  - Better dynamic algorithms that handle arrivals and departures; can one improve more than 10% over batch approach?
    - E.g., potentials based on more donor and patient features [we're working on this]
  - Better failure-aware algorithms
  - Better edge testing policies [we're working on this]
  - Matching cadence: Race to bottom among exchanges
    [Das, Dickerson, Li, Sandholm, AMMA-15]
  - Better incentive schemes
    - Credit scheme [Hajaj, Dickerson, Hassidim, Sandholm, Sarne, AAAI-15]
  - Multi-donor kidney exchange [we're working on this]
  - Other organs
    - Liver & cross-organ exchange [Dickerson & Sandholm, GREEN-COPLAS-13, AAAI-14]
    - Lung "components" [Ergin, Sönmez, Ünver, draft 2014-15; Tang et al. 2015]

Future work regarding fielding

- Getting dynamic and failure-aware approach fielded
- Better crossmatch prediction
- Better edge testing policies
- Getting credit schemes fielded
- Better donor pre-select tools
- Insurance to pay for testing, etc.
- International exchange
- Exchanges beyond kidneys
- In the US: Shutting down sniping manual private exchanges, and having one system (as there is for deceased donors)