Kidney Exchange

With an emphasis on computation & work from CMU

John P. Dickerson

(in lieu of Ariel Procaccia)

Carnegie Mellon University

Today's lecture: kidney exchange

Hmm … Hmm …

Al Roth

Tayfun Sönmez

Utku Ünver

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

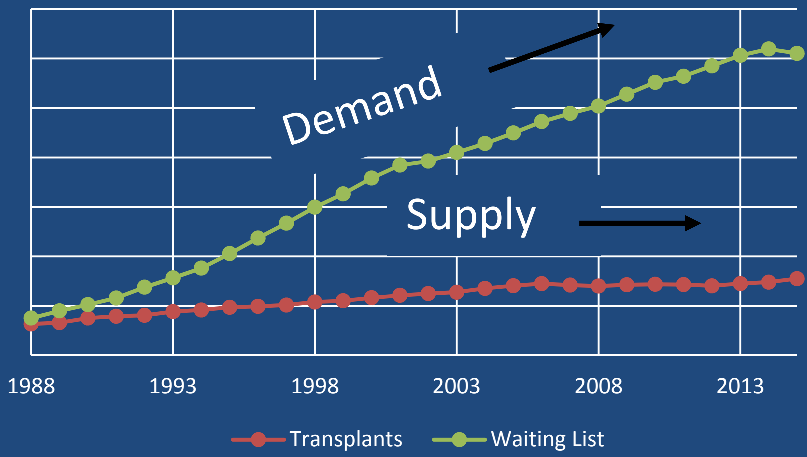
This is a fairly CMU-centric lecture because some of it is on my thesis work, but I am happy to talk about anything related to kidney exchange!

High-Level Motivation

~~Organ Failure~~

Kidney Failure

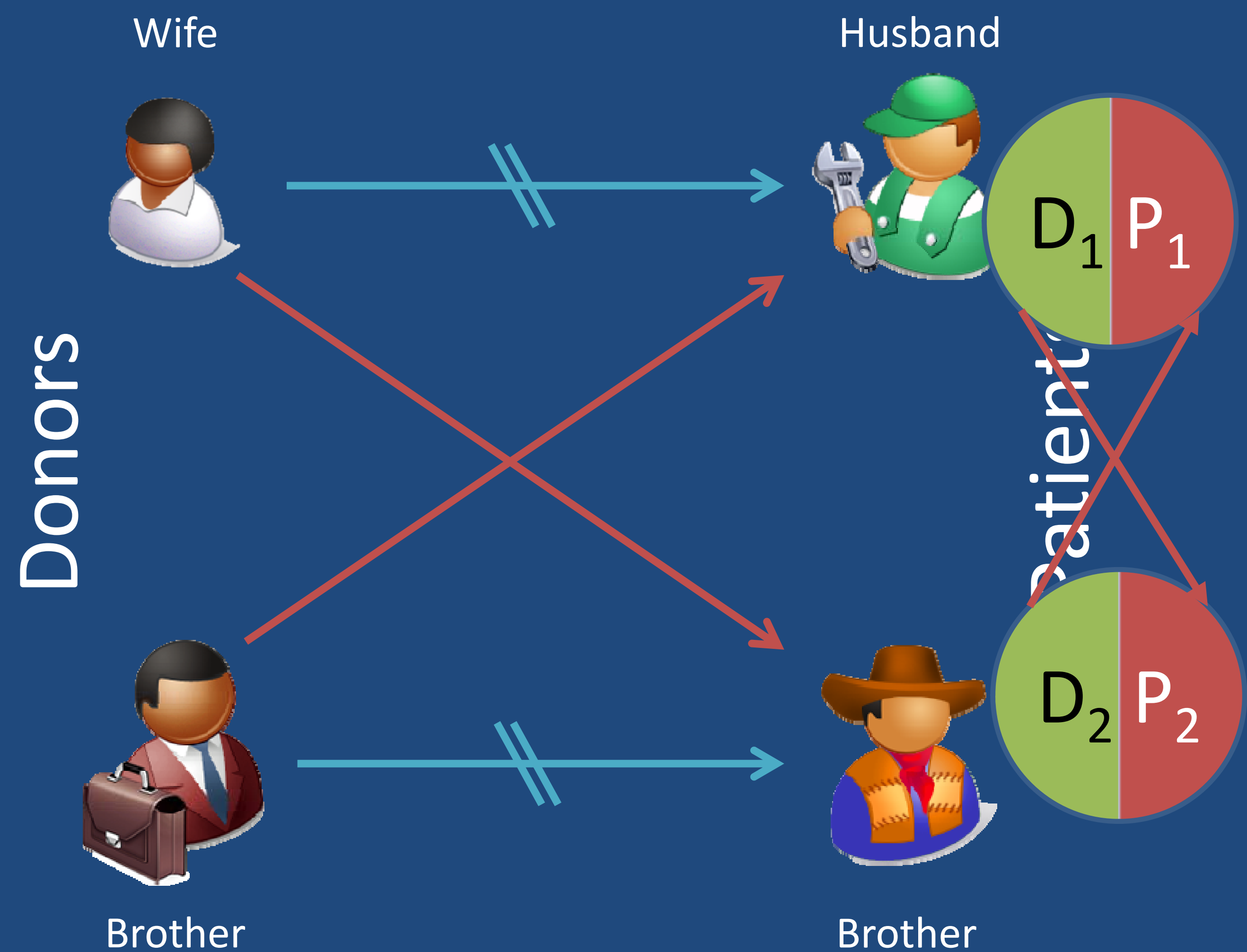
Kidney transplantation



- 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

UNOS DONATE LIFE

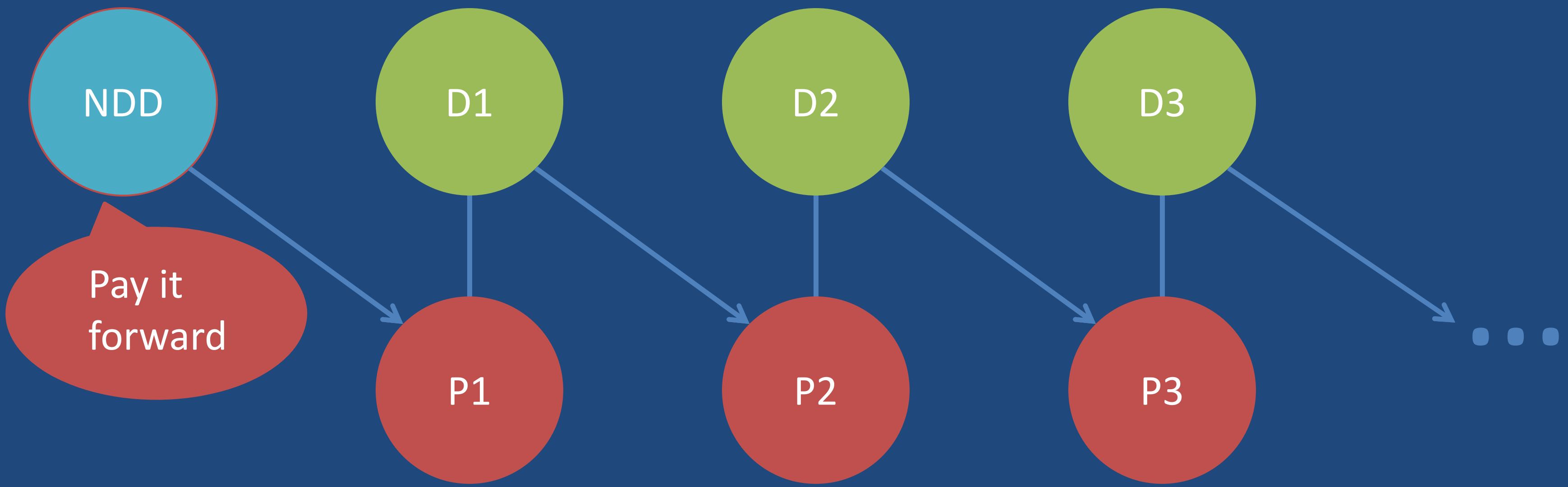
Kidney exchange



(2- and 3-cycles, all surgeries performed simultaneously)

Non-directed donors & chains

[Rees et al. 2009]



- *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

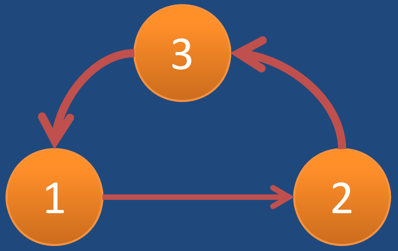
- NEPKE (started 2003/2004, now closed)
- United Network for Organ Sharing (UNOS)
  - US-wide, 140+ transplant centers
  - Went live Oct. 2010, conducts biweekly matches
- Alliance for Paired Donation
- Paired Donation Network (now closed)
- National Kidney Registry (NKR)
- San Antonio
- Canada
- Netherlands
- 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, $2 \le L \le 5$ 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

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

Flow constraint

Subject to

$\Sigma_j\, x_{ij} = \Sigma_j\, x_{ji}$ for each vertex *i*

$\Sigma_j\, x_{ij} \leq 1$ for each vertex *i*

$\Sigma_{1\leq k\leq L}\, x_{i(k)i(k+1)} \leq L-1$ for paths i(1)…i(L+1)

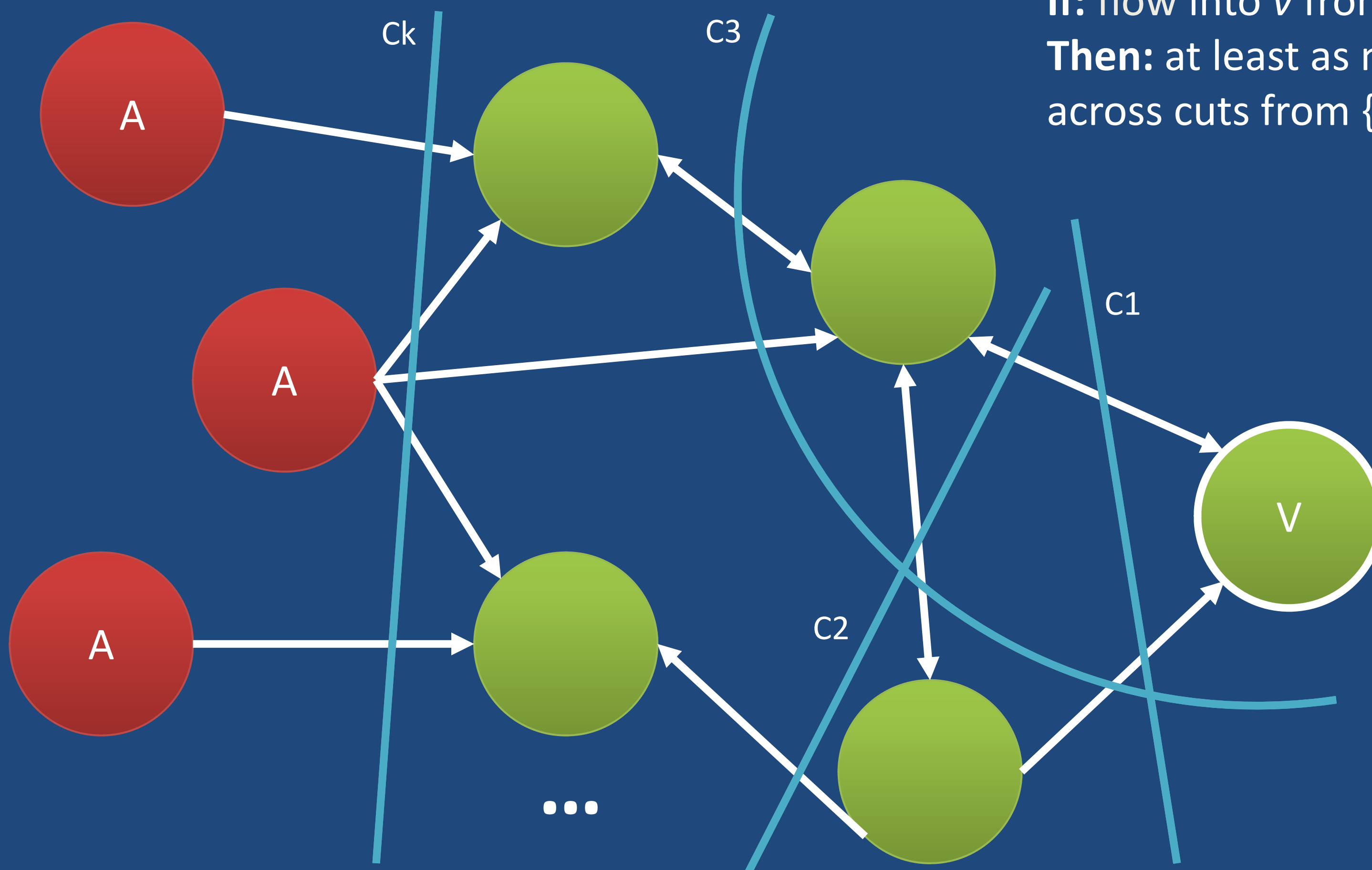
(no path of length L that doesn't end where it started – cycle cap)

Best Edge Formulation

"Best" = maximum cardinality

[Anderson et al. 15]

If: flow into *v* from a chain
Then: at least as much flow across cuts from {A}



Basic IP formulation #2

"Best" = maximum cardinality

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

Maximize

$$\Sigma\ |c|\, x_c$$

Subject to

$$\Sigma_{c\,:\,i \text{ in } c}\, x_c \leq 1 \qquad \text{for each vertex } i$$

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 → constraint in full model violated
  - No positive price variables → $OPT_{reduced} = OPT_{full}$
- 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

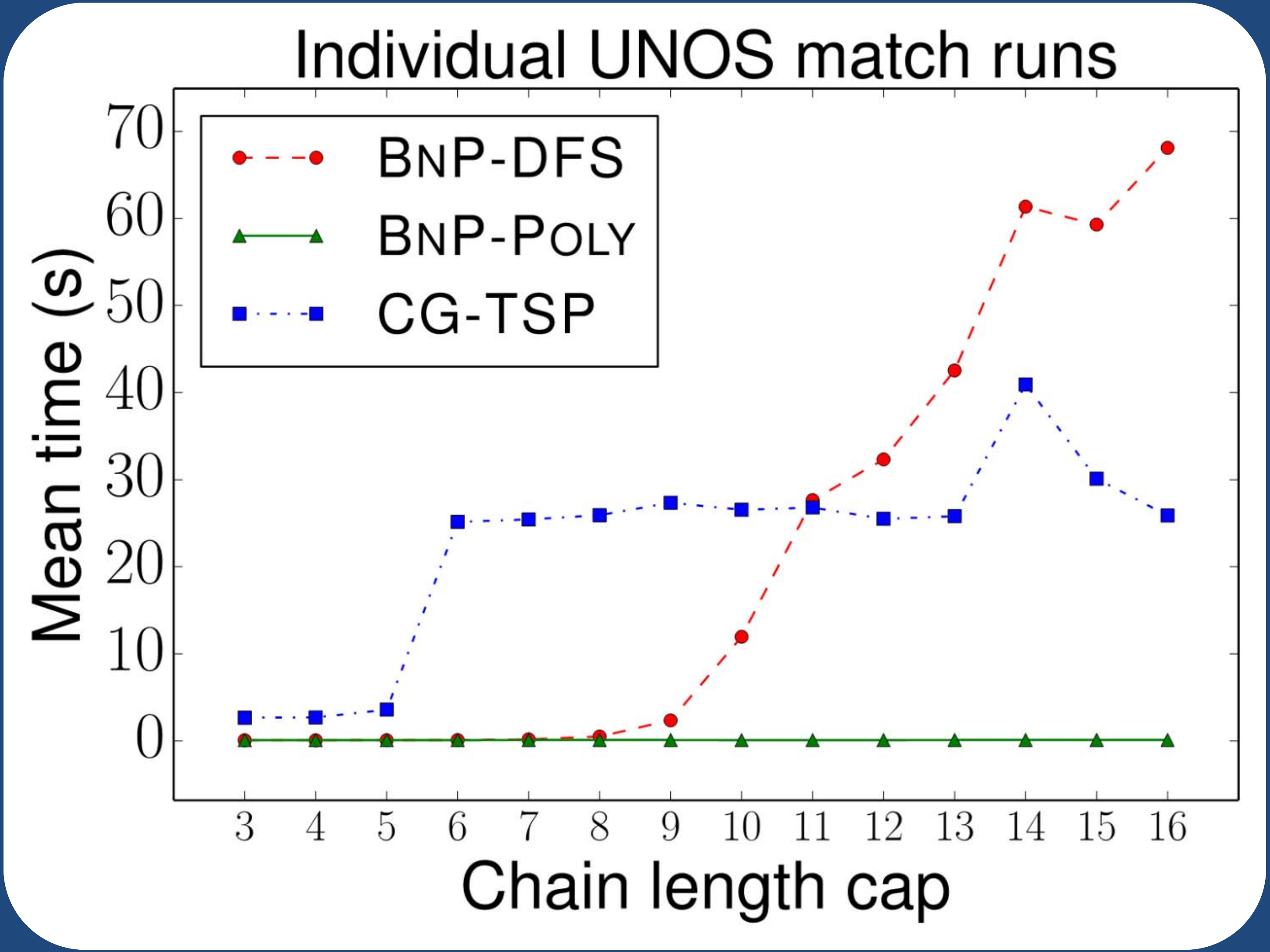
The Right Idea

- Idea: solve structured optimization problem that implicitly prices variables

- Price: $w_c - \Sigma_{v \text{ in } c}\, \delta_v \;=\; \Sigma_{e \text{ in } c}\, w_e - \Sigma_{v \text{ in } c}\, \delta_v \;=\; \Sigma_{(u,v) \text{ in } c}\, [w_{(u,v)} - \delta_v]$
- 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 $1-|V|$ → visits each vertex exactly once
    - We only need *some* short path (or proof that no negative cycle exists)
  - Now pricing runs in time $O(|V||E|\text{cap}^2)$

Experimental results



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 $(L+1)/2$ – ½ 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

$$u(c) = \underbrace{\sum w_e}_{\text{Value of successful cycle}} \bullet \underbrace{\prod q_e}_{\text{Probability of success}}$$

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

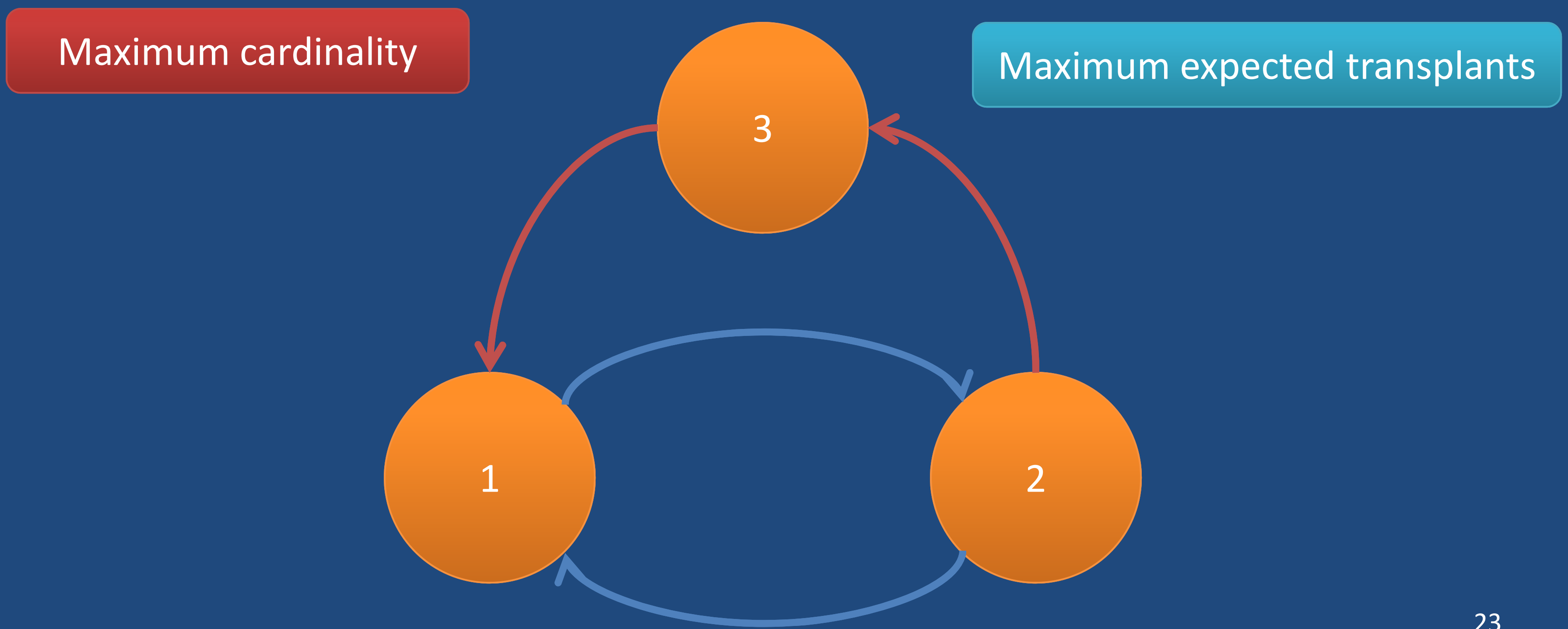
Chain executes in entirety

- 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

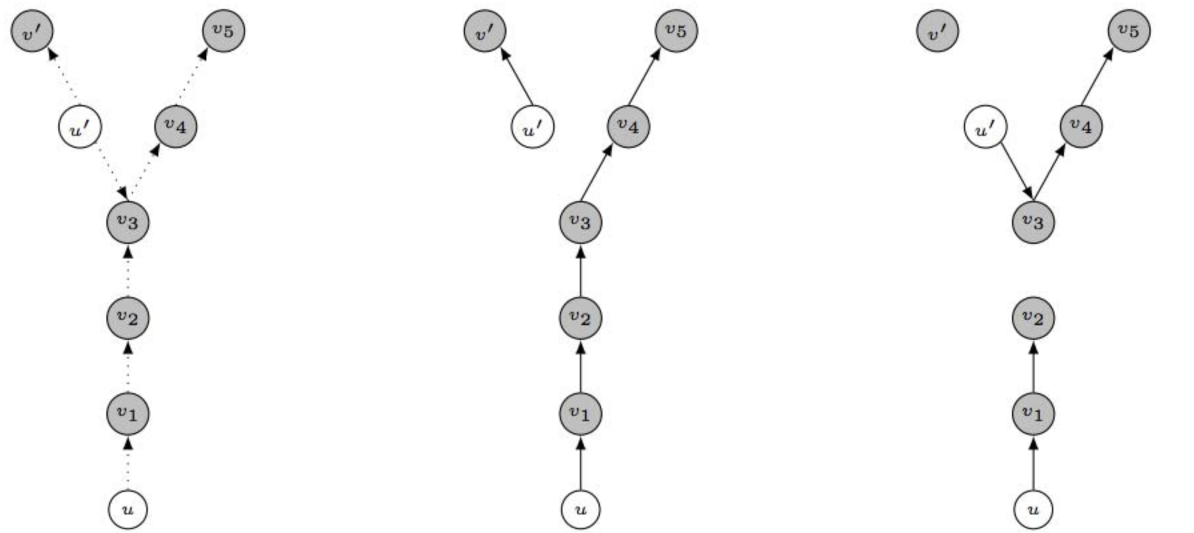
- $G(n, t(n), p)$: random graph with
  - n patient-donor pairs
  - $t(n)$ altruistic donors
  - Probability $\Theta(1/n)$ of incoming edges
- Constant transplant success probability q

Theorem

For all $q \in (0,1)$ and $\alpha, \beta > 0$, 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,

$$u_q(M') \geq u_q(M) + \Omega(n)$$

Brief intuition: Counting Y-gadgets

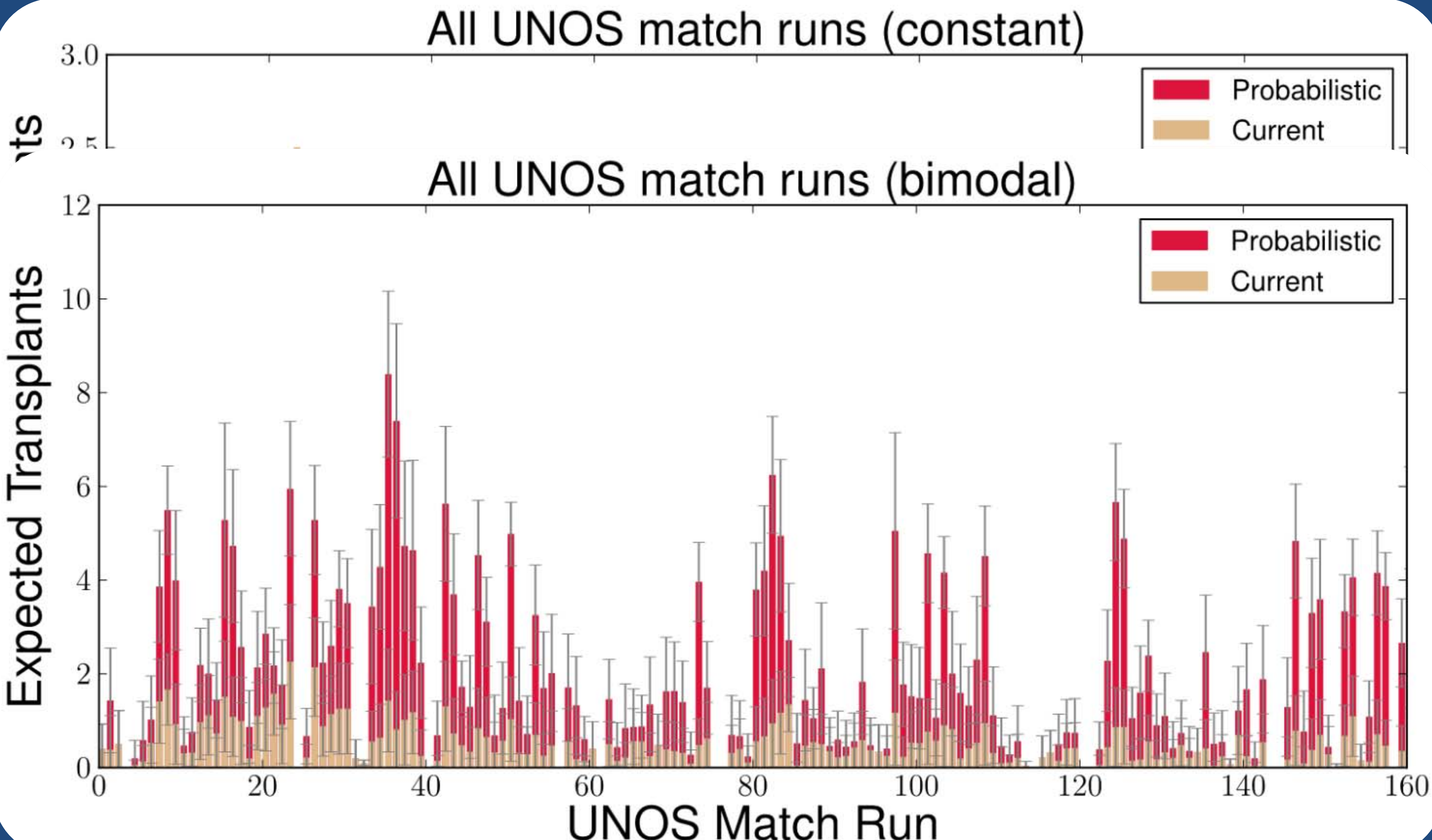


(a) A Y gadget. (b) The maximum cardinality matching M_Y. (c) The matching M'_Y.

- For every structure *X* of constant size, w.h.p. can find *Ω*(*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 × *Ω*(*n*) = *Ω*(*n*)
- Direct the edges: flip 50/50 coins, constant fraction are entirely directed correctly → constant × *Ω*(*n*) = *Ω*(*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
All UNOS match runs (bimodal)
Expected Transplants
12
10
8
6
4
2
0
0
20
40
60
80
100
120
140
160
UNOS Match Run
Probabilistic
Current

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

- 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 → resp. constraint in full model violated
  - No positive price cycles → optimality at this node

Considering only "good" chains

Theorem:
Given a chain *c*, any extension *c'* will not be needed in an optimal solution if the infinite extension has non-positive value.

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

Optimistic future value of infinite extension

Donation to waitlist

Discounted utility of current chain

Pessimistic sum of LP dual values in model

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.

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

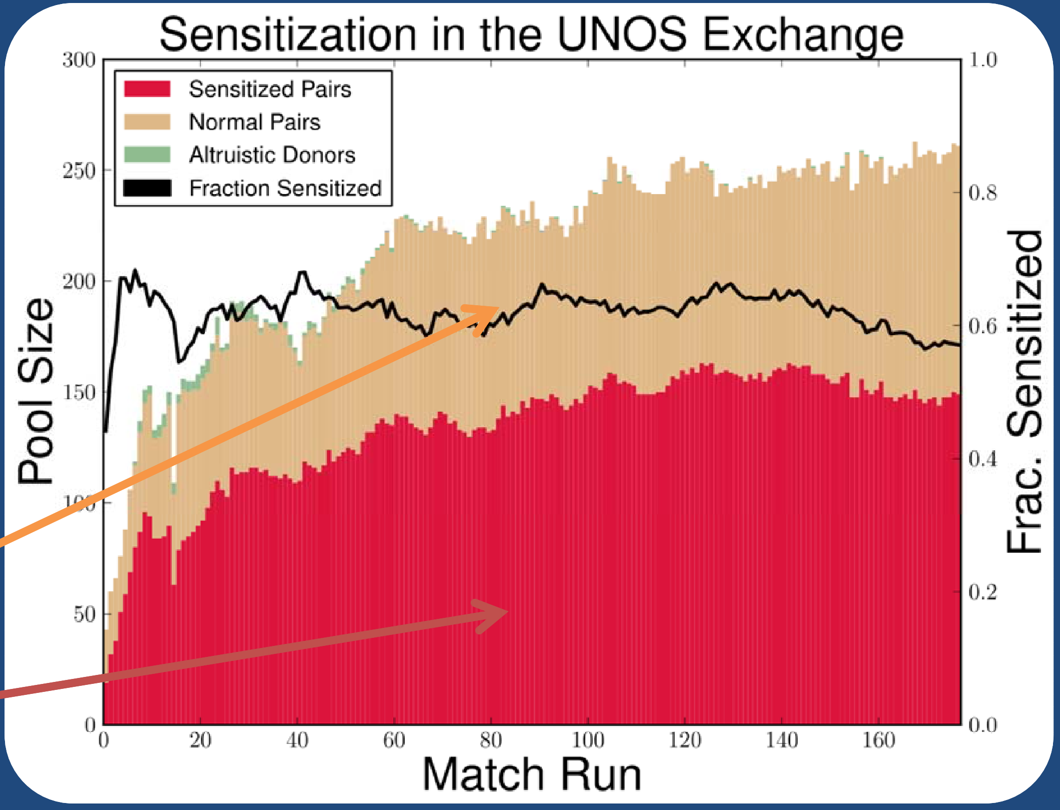
Sensitization at UNOS

- 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 $u : \mathcal{M} \rightarrow \mathbb{R}$

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

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

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

Theoretical result #2

Under the "most stringent" fairness rule:

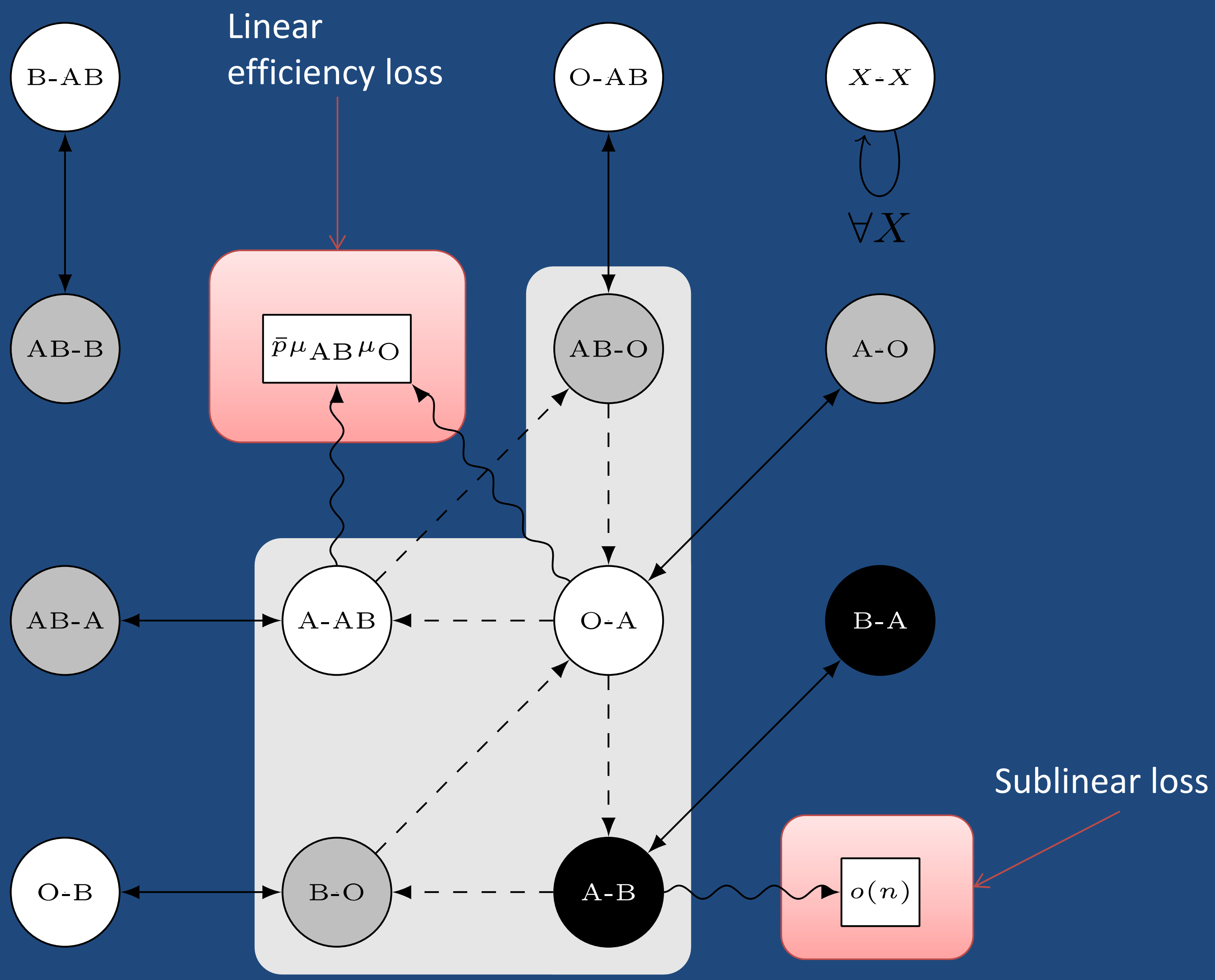
$$u_{H \succ L}(M) = \begin{cases} u(M) & \text{if } |M_H| = \max_{M' \in \mathcal{M}} |M'_H| \\ 0 & \text{otherwise} \end{cases}$$

Theorem

Assume "reasonable" level of sensitization and "reasonable" distribution of blood types. Then, almost surely as *n* → ∞,

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

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



Linear
efficiency loss
B-AB
O-AB
X-X
$\forall X$
AB-B
$\bar{p}\mu_{\mathrm{AB}}\mu_{\mathrm{O}}$
AB-O
A-O
AB-A
A-AB
O-A
B-A
Sublinear loss
O-B
B-O
A-B
$o(n)$

From theory to practice

- Price of fairness is **low** in theory
- Fairness criterion: *extremely* strict.
- Theoretical assumptions (standard):
  - Big graphs ("$n \rightarrow \infty$")
  - 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 α 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 → 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

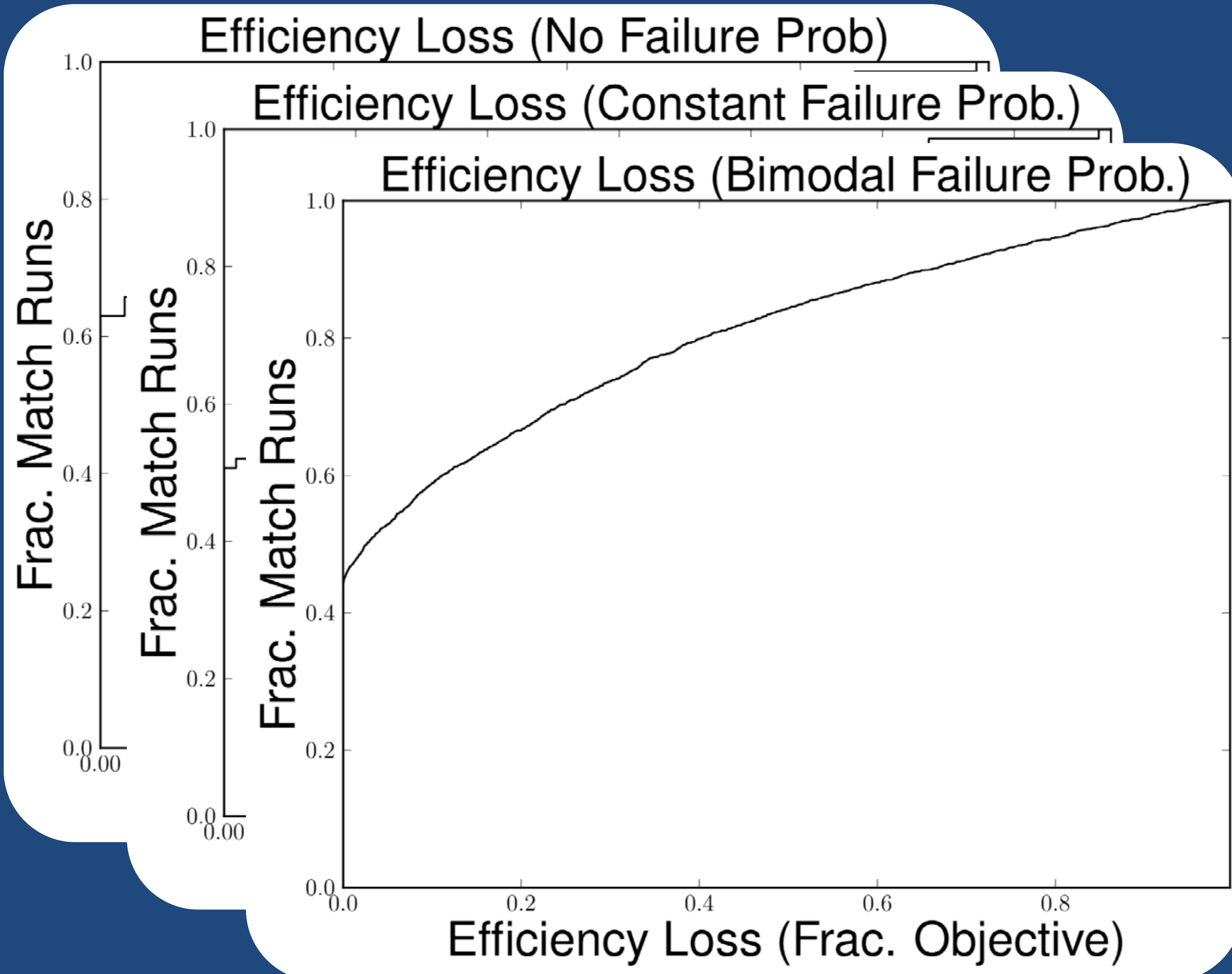
Price of fairness: Generated data

Size	Saidman (US)	Saidman (UNOS)	Heterogeneous
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%)

- 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

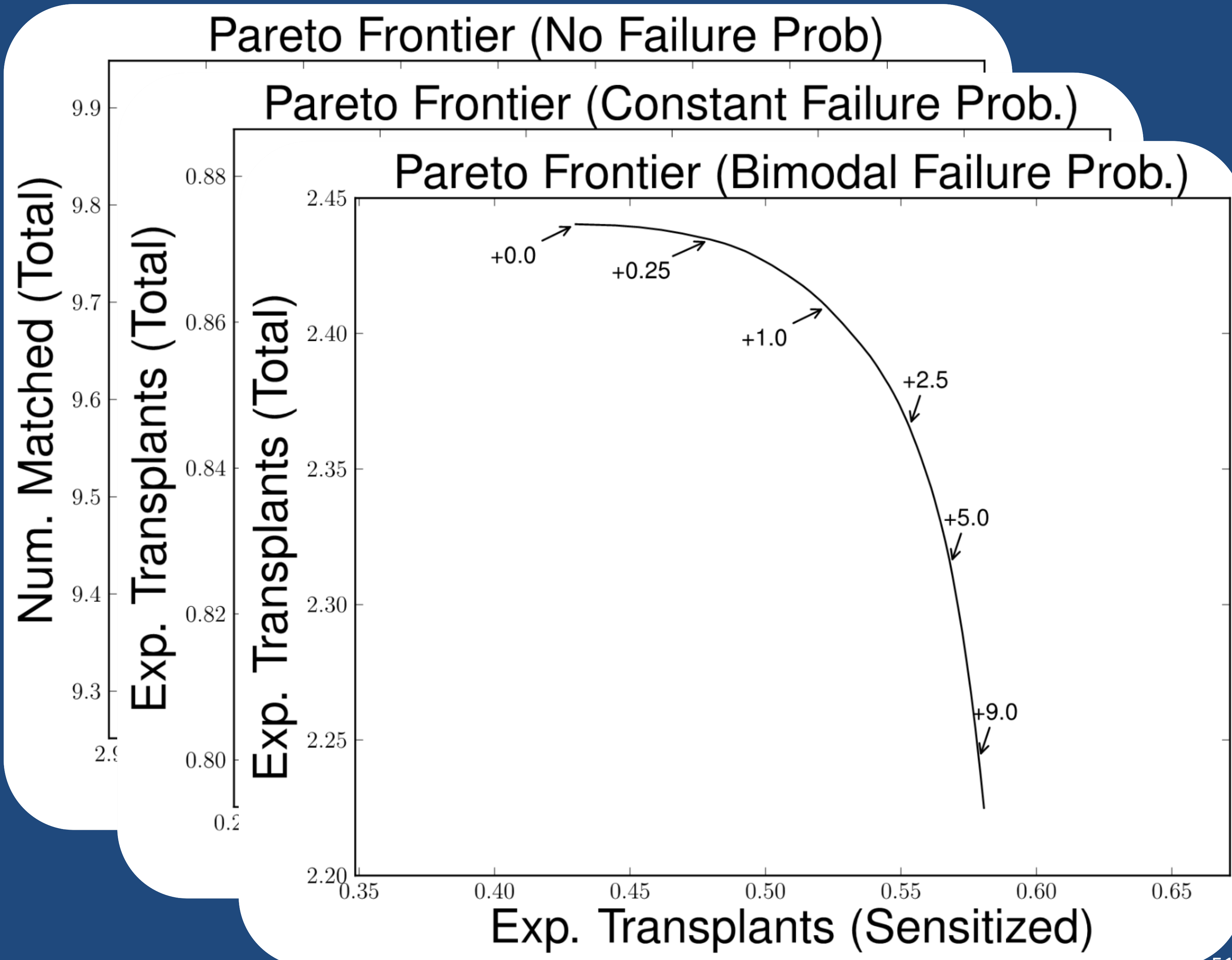
Real UNOS runs

Lexicographic fairness, varying failure rates


Efficiency Loss (No Failure Prob)
Efficiency Loss (Constant Failure Prob.)
Efficiency Loss (Bimodal Failure Prob.)
Frac. Match Runs
Frac. Match Runs
Frac. Match Runs
1.0
0.8
0.6
0.4
0.2
0.0
0.00
0.0
0.2
0.4
0.6
0.8
Efficiency Loss (Frac. Objective)

Real UNOS runs

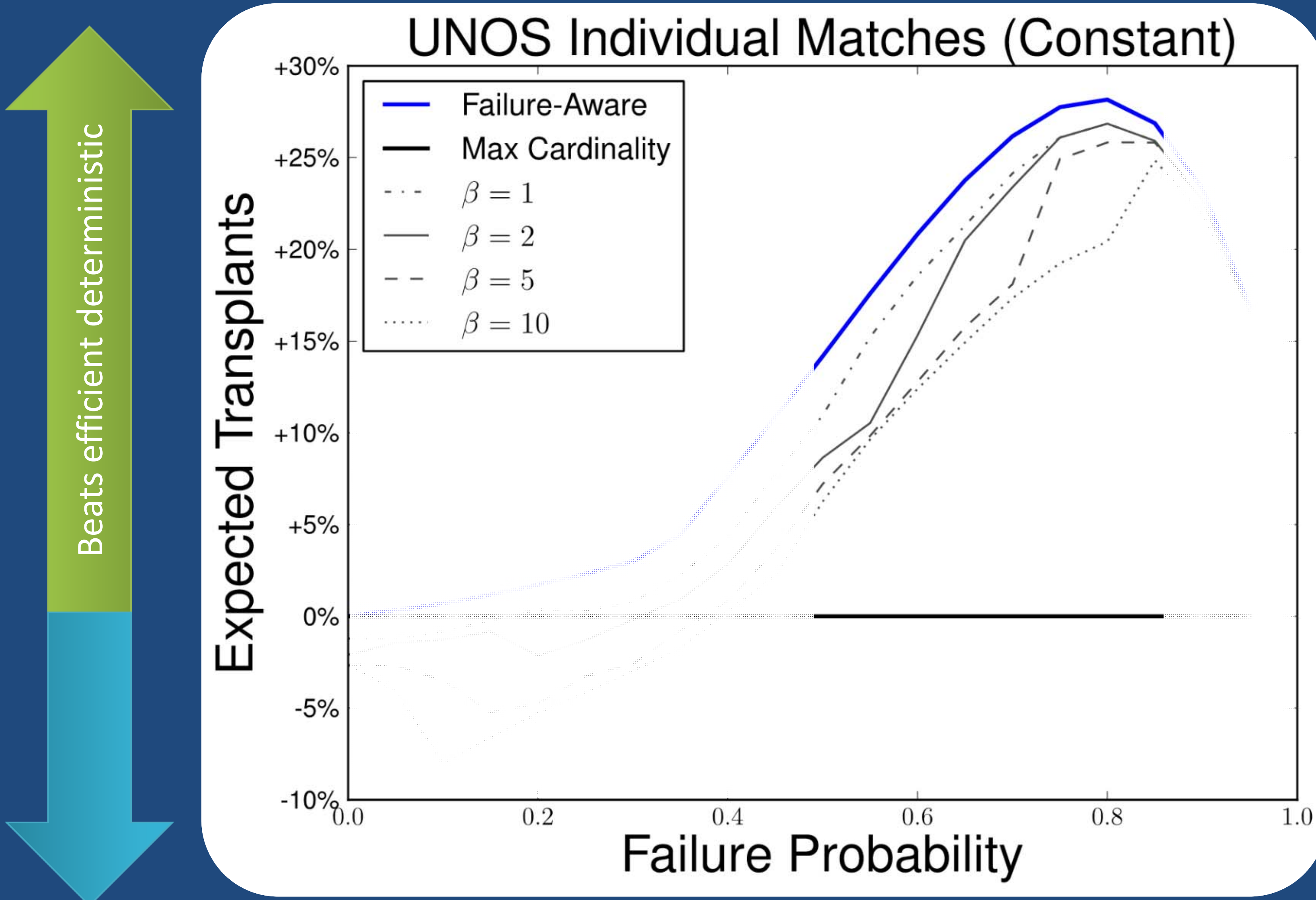
Weighted fairness, varying failure rates



Pareto Frontier (No Failure Prob)
Num. Matched (Total)
9.9
9.8
9.7
9.6
9.5
9.4
9.3
Pareto Frontier (Constant Failure Prob.)
Exp. Transplants (Total)
0.88
0.86
0.84
0.82
0.80
Pareto Frontier (Bimodal Failure Prob.)
Exp. Transplants (Total)
2.45
2.40
2.35
2.30
2.25
2.20
+0.0
+0.25
+1.0
+2.5
+5.0
+9.0
0.35
0.40
0.45
0.50
0.55
0.60
0.65
Exp. Transplants (Sensitized)

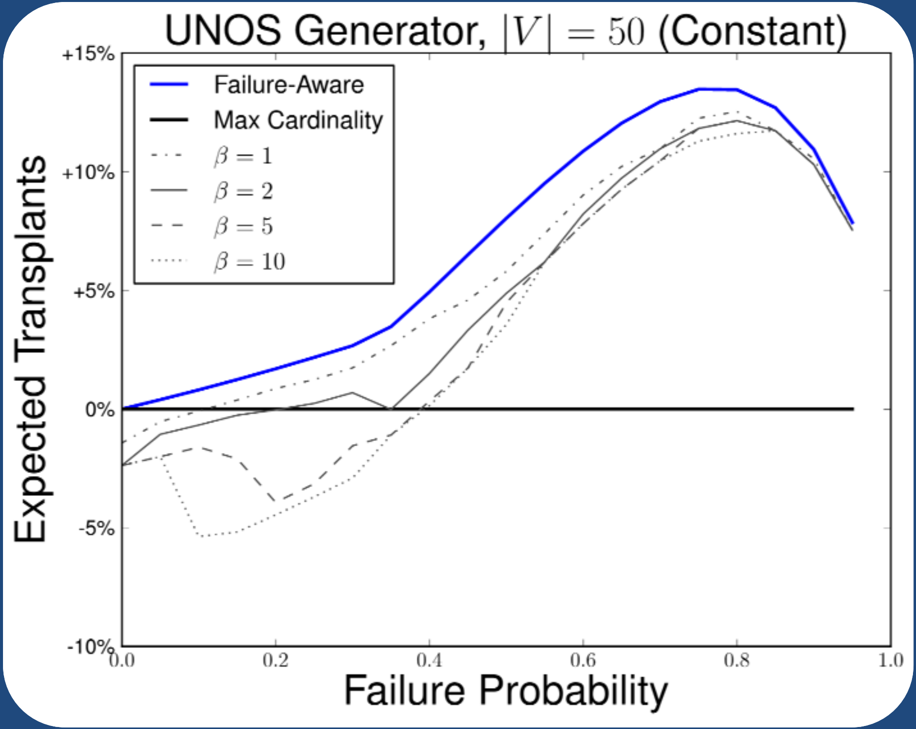
Contradictory goals

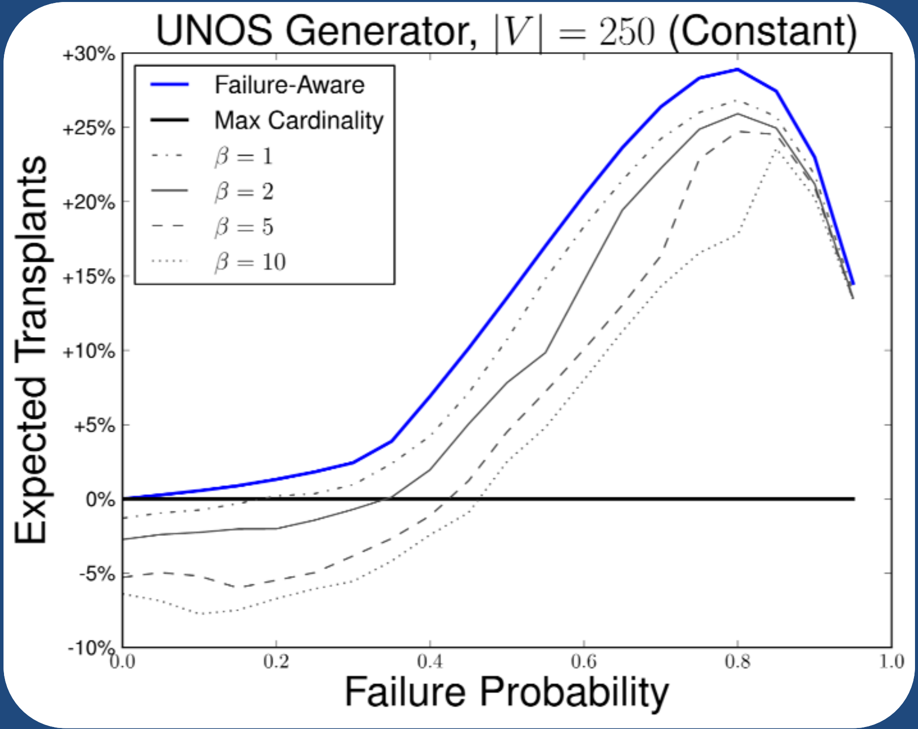
- Earlier, we saw **failure-aware** matching results in tremendous gains in #expected transplants
- Gain comes at a price – may further marginalize hard-to-match patients because:
  - Highly-sensitized patients tend to be matched in chains
  - Highly-sensitized patients may have higher failure rates (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)

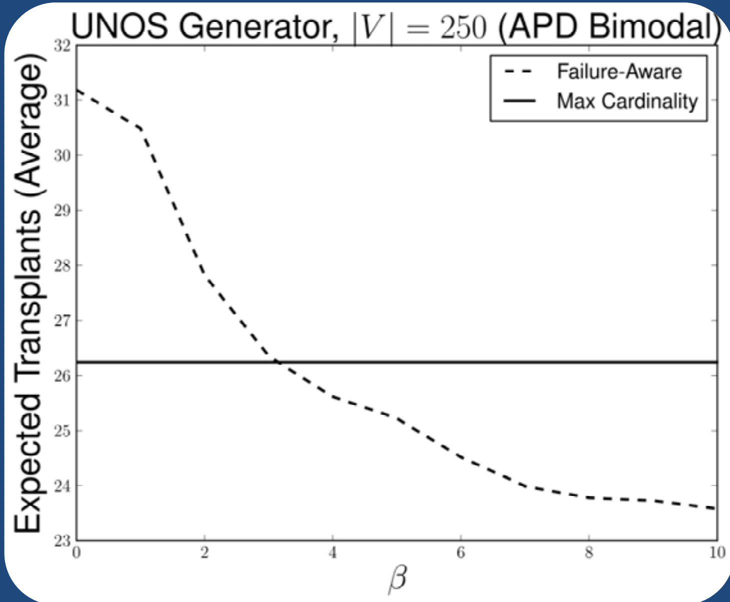
UNOS distributional generator

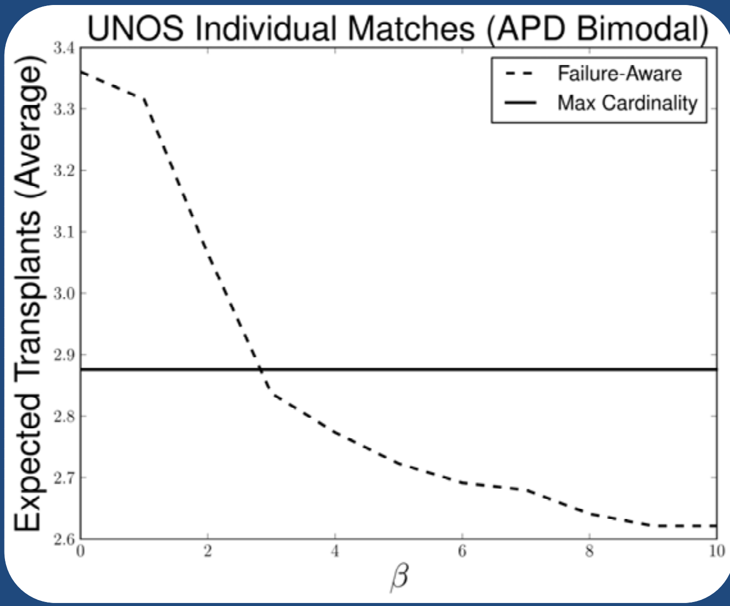




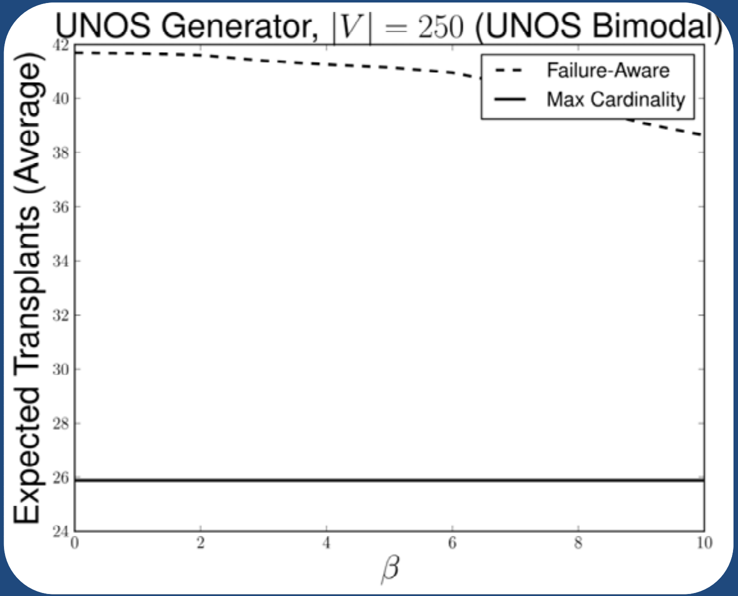
Generated UNOS runs, weighted fairness, constant probability of failure (x-axis), increase in expected transplants over deterministic matching (y-axis)

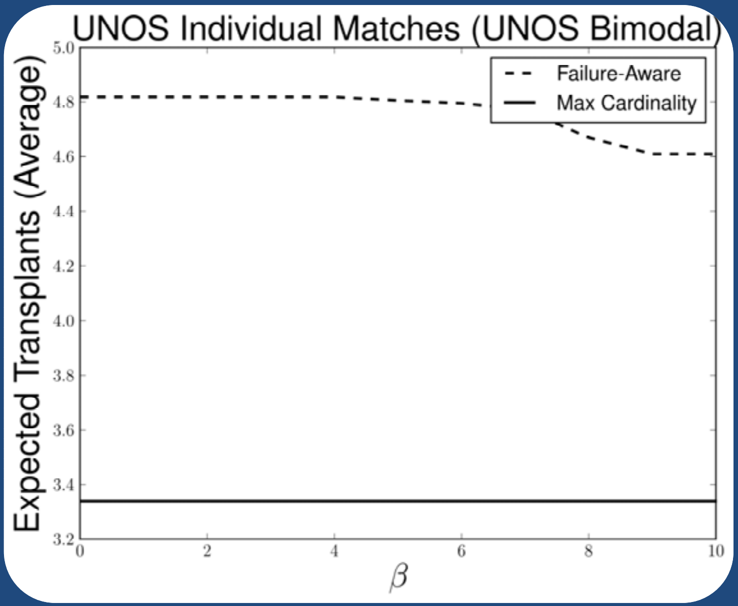
APD failure rate





UNOS failure rate





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*?

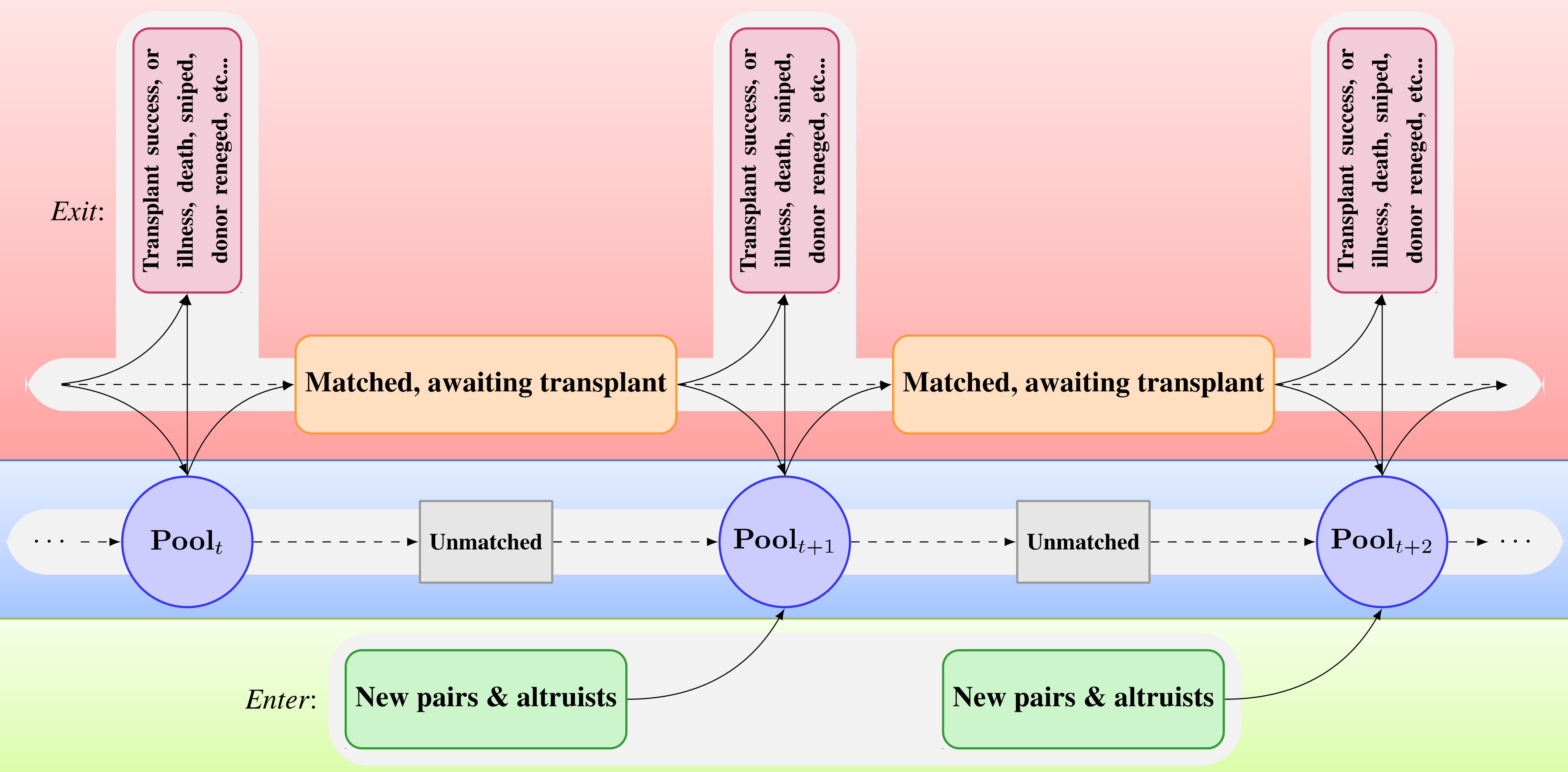
Dimension #3: Dynamism

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

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		

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?

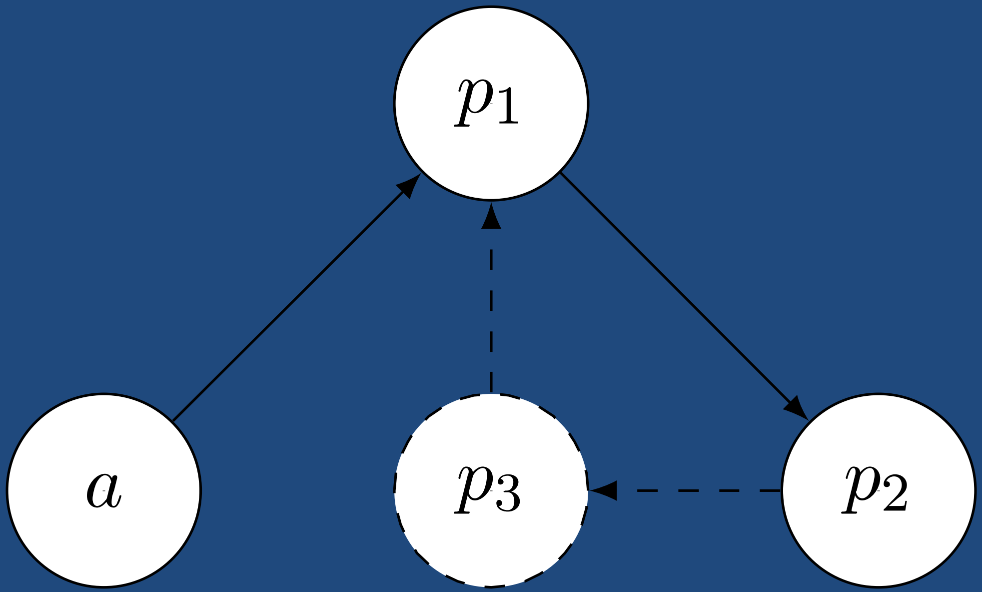
- Given a set of features *Θ* representing structural elements (e.g., vertex, edge, subgraph type) of a problem:
  - The potential P_ϑ for a type ϑ quantifies the *future usefulness* of that element
- E.g., let *Θ* = {O-O, O-A, …, AB-AB, •-O, …, •-AB}
  - 16 patient-donor types, 4 altruist types
  - O-donors better than A-donors, so: $P_{\bullet\text{-}O} > P_{\bullet\text{-}A}$

Using potentials to inform myopia

- Using heavy one-time computation to learn potential of each type ϑ
- Adjust solver to take them 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

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

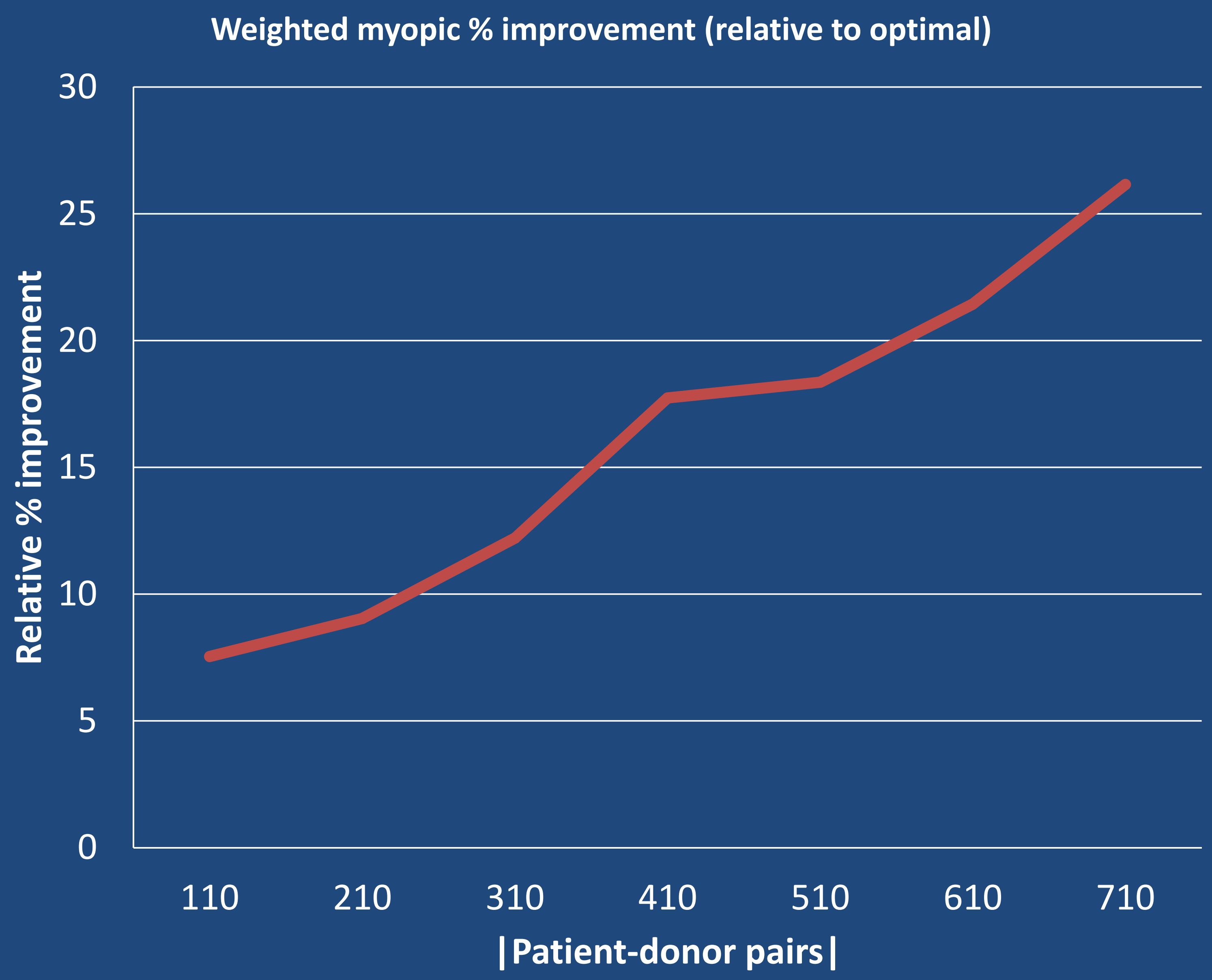
Expressiveness Theory

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)
Relative % improvement
30
25
20
15
10
5
0
110
210
310
410
510
610
710
|Patient-donor pairs|

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

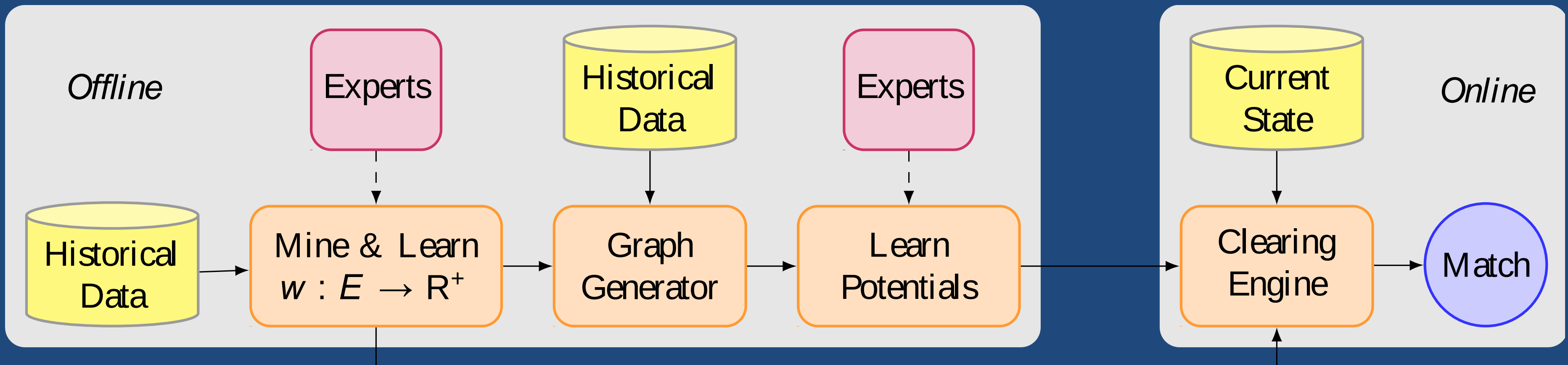
[Dickerson Sandholm AAAI-2015]

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)

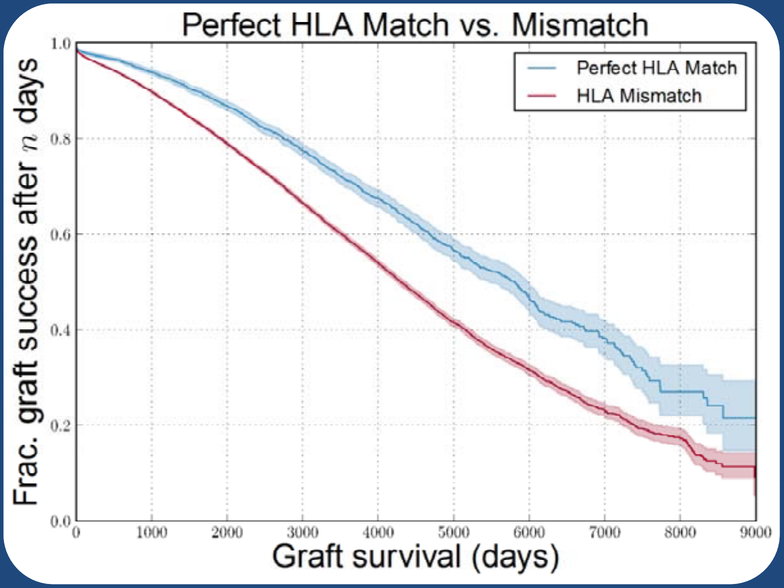
1. Domain expert describes overall goal
2. Take historical data and policy input to learn a weight function *w* for match quality
3. Take historical data and create a graph generator with edge weights set by *w*
4. Using this generator and a realistic exchange simulator, learn potentials for graph elements 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

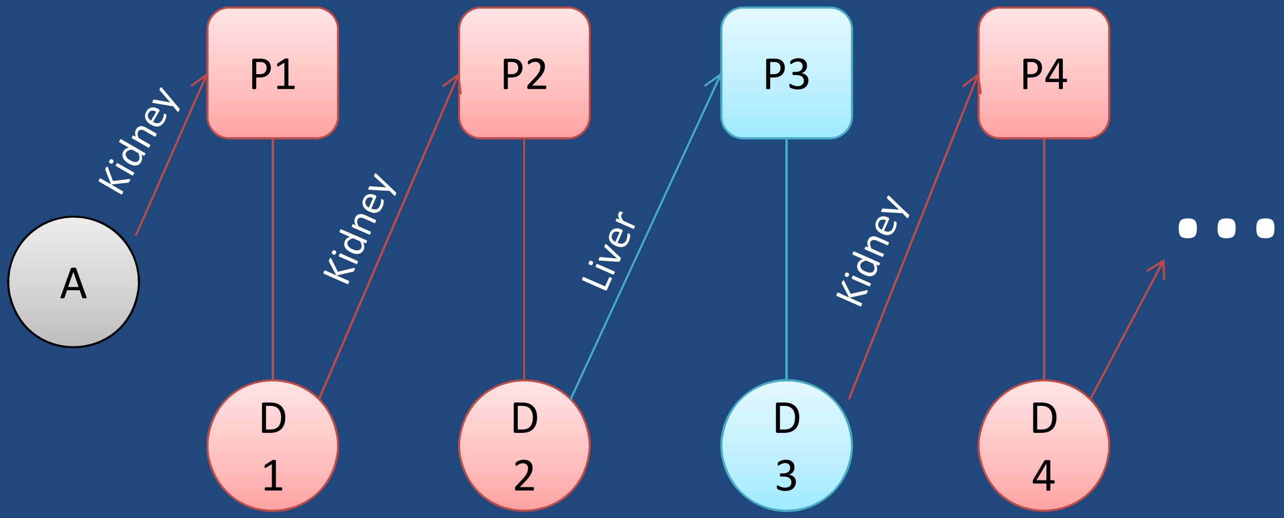
Lots left to do!

- 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

[Dickerson Sandholm AAAI-2014]

- 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-λ 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 $p_{K \rightarrow L} > 0$

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

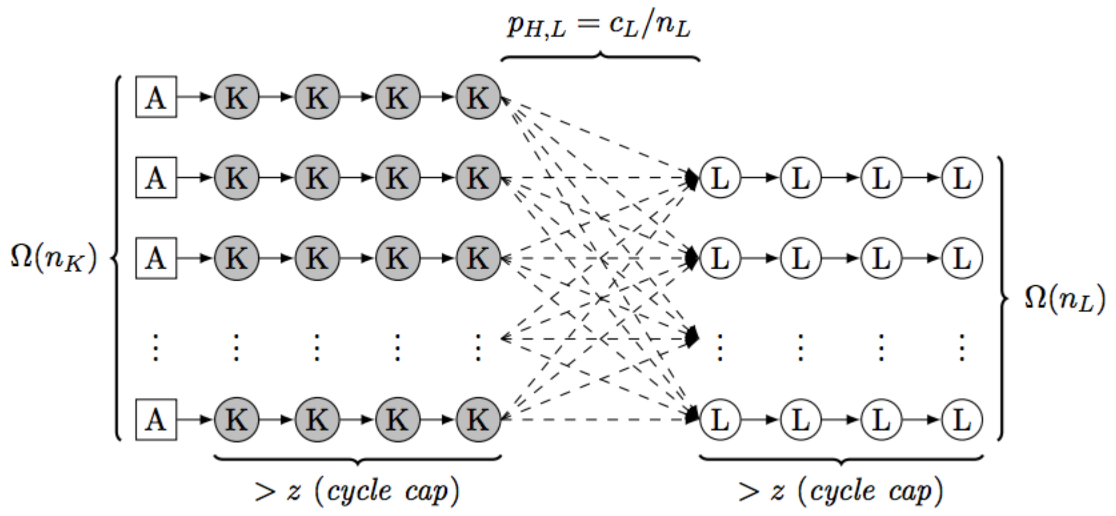
Assume $t(n_K) = \beta n_K$ for some constant $\beta>0$. Then, with probability 1 as $n_K \rightarrow \infty$,

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.

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
  - → linear number of D_K chains extended into D_L
  - → 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!
- λ is frac. lowly-sensitized
- Constant cycle cap z

Theorem

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

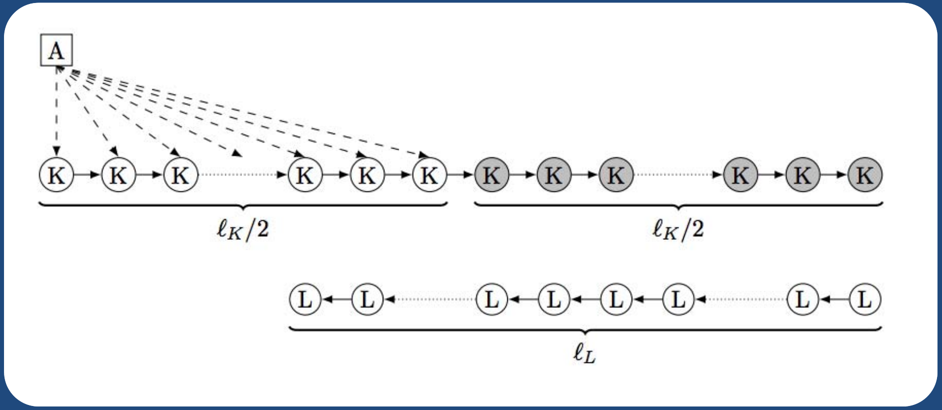
Any efficient matching on $D = \text{join}(D_K, D_L)$ matches $\Omega(n_K)$ more pairs than the aggregate of efficient matchings on D_K and D_L.

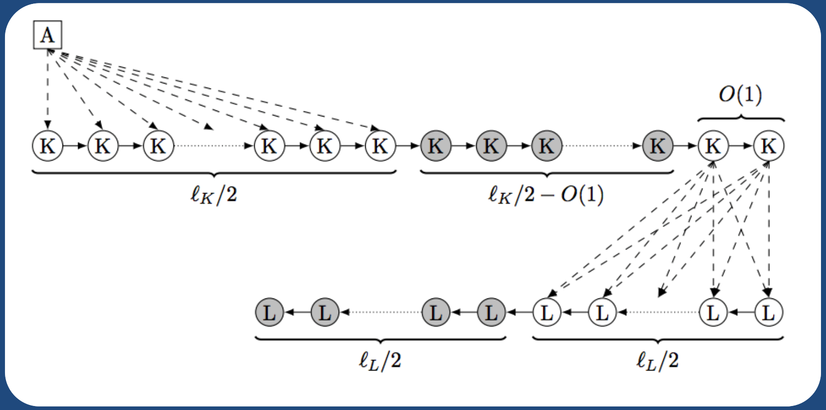
With constant positive probability.

Building on [Ashlagi et al. 2012]

Intuition

- For large enough 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

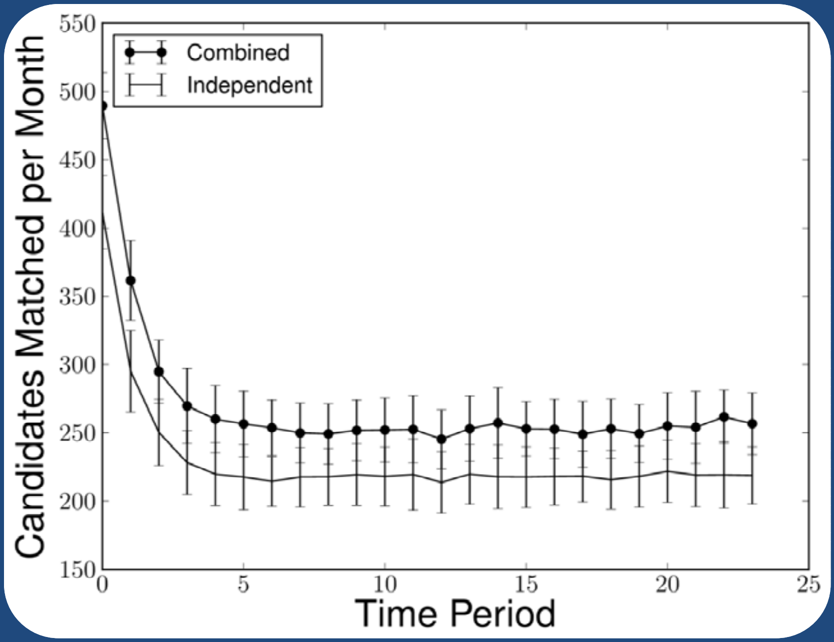




FutureMatch + multi-organ exchange?

- Combination results in
  - Linear gain in theory
  - Big gains in simulation
- **Equity problems**
  - Kidneys ≠ livers
  - Hard to quantify cross-organ risk vs. reward

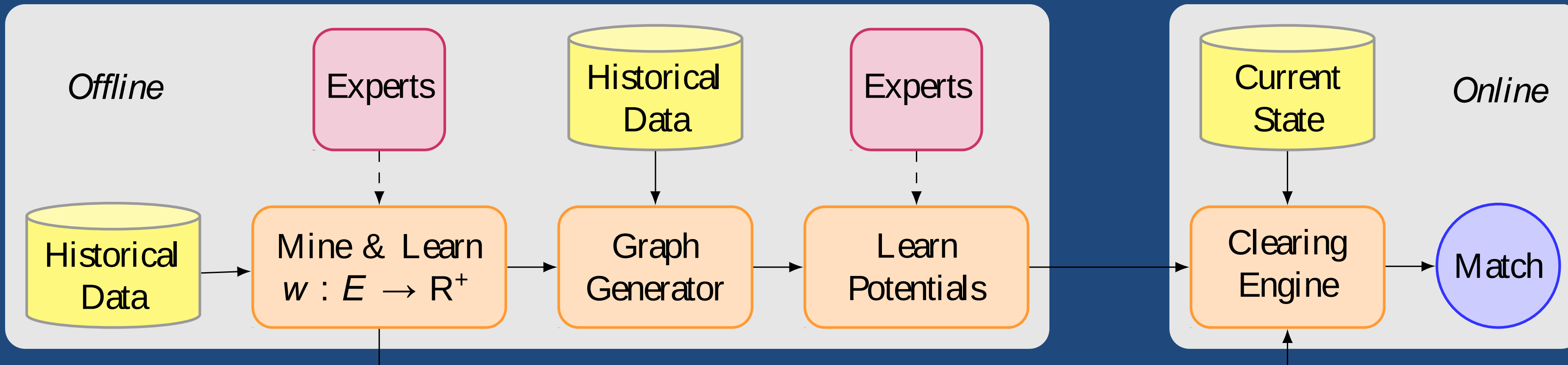
Let FutureMatch sort it out?



- 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

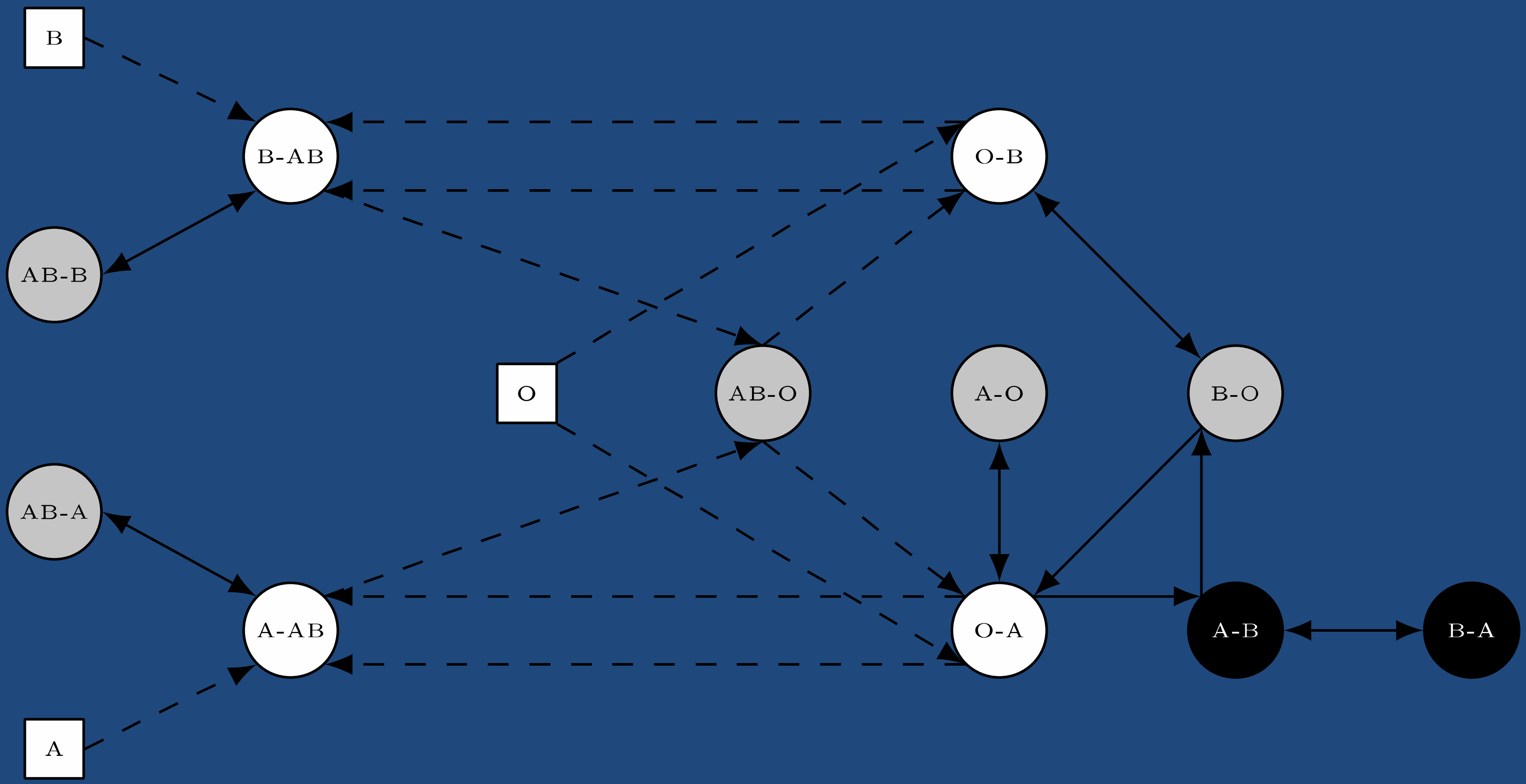
Very incomplete list of CMU folks working on kidney exchange/matching:

{ Avrim Blum, John Dickerson, Alan Frieze, Anupam Gupta, Nika Haghtalab, Jamie Morgenstern, Ariel Procaccia, R. Ravi, Tuomas Sandholm }

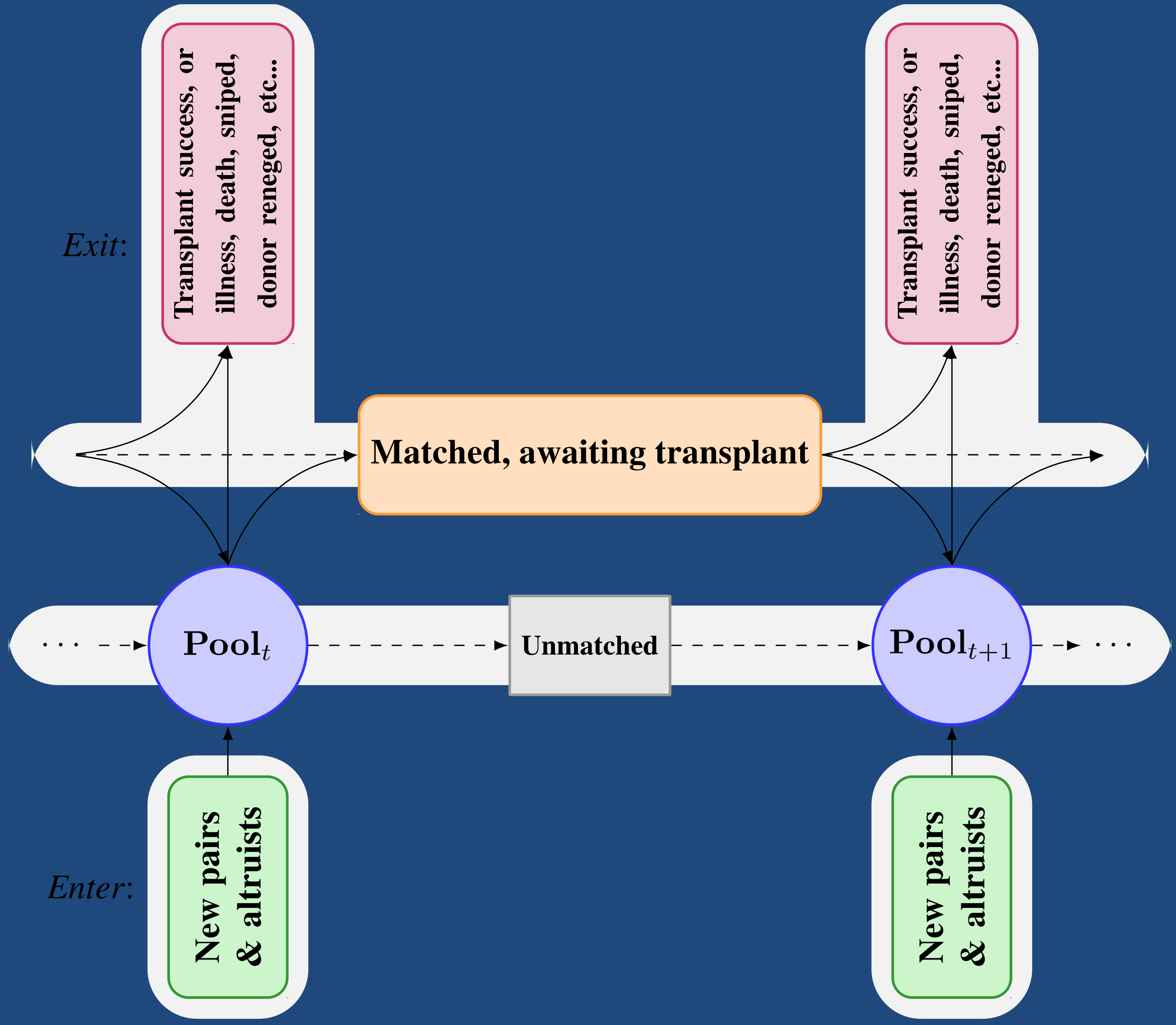
Thanks to:

Kidney Exchange

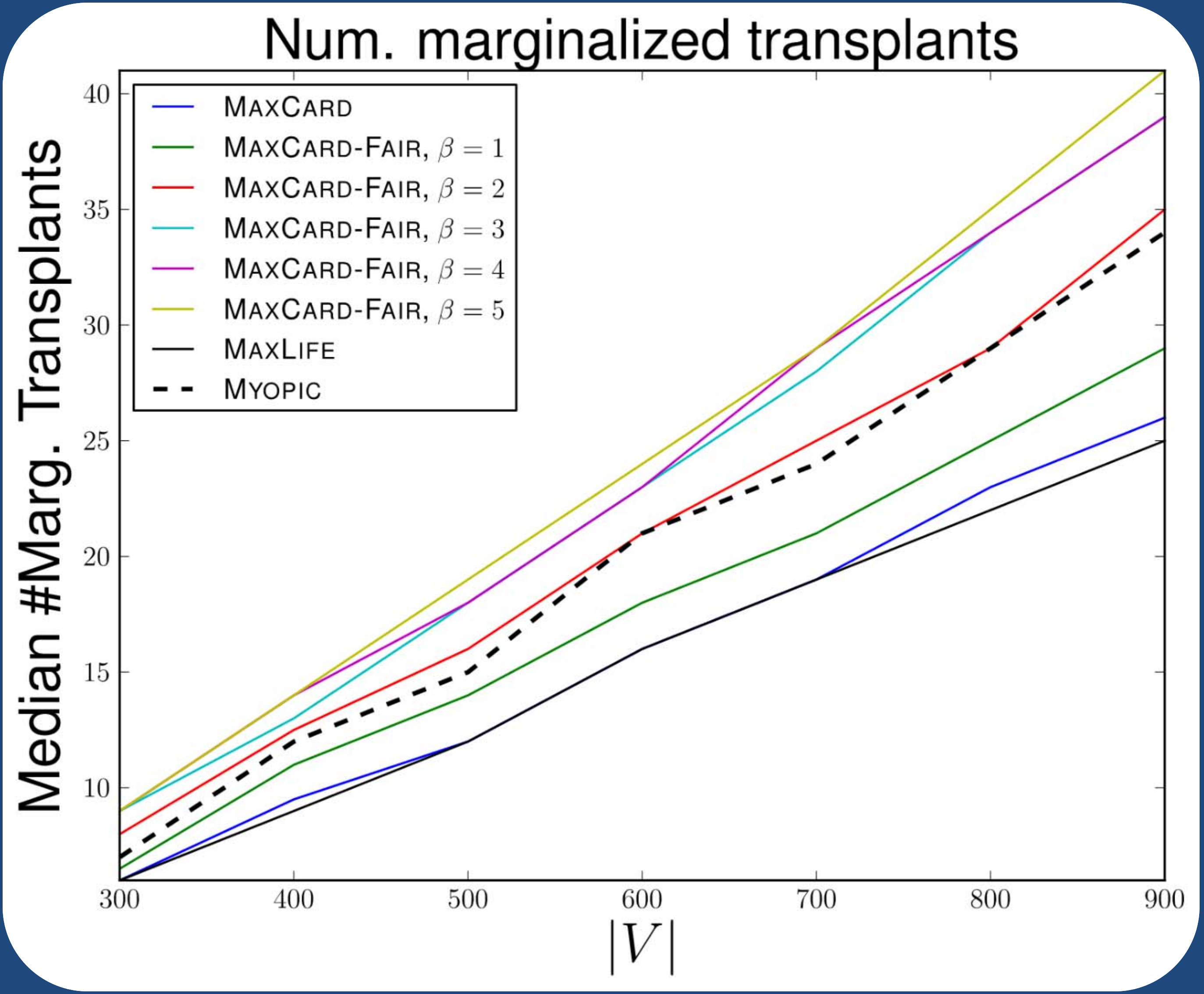
Backup Slides



- 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

Metric	Minimum	Average	Maximum	St. Dev.
Loss % (Objective)	0.00%	2.76%	19.04%	4.84%
Loss % (Cardinality)	0.00%	4.09%	33.33%	8.18%
Loss (Cardinality)	0	0.55	4	1.1

- 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)