EFFICIENT, ETHICAL, & FAIR MATCHING MARKET DESIGN VIA OPTIMIZATION

JOHN P DICKERSON

COMPUTER SCIENCE
UNIVERSITY OF MARYLAND

The Cornell, Maryland, Max Planck
Pre-doctoral Research School
August 7, 8, & 10, 2018

Markets come in many forms …

… some of which don't conform to conventional notions of markets …

… and some in which money may play little or no role.

– excerpt from *Who Gets What – and Why*

MATCHING MARKETS

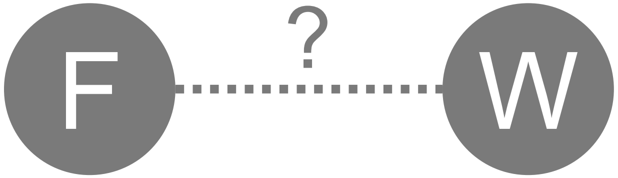
In matching problems, prices do not do all – or any – of the work

Agents are paired with other (groups of) agents, transactions, or contracts

- Workers to firms
- Children to schools
- Residents to hospitals
- Patients to donors
- Advertisements to viewers
- Riders to rideshare drivers

UNCERTAINTY

- Does a matched edge truly exist?
- How valuable is a match?
- Will a better match arrive in the future?



COMPETITION

Rival matching markets compete over the same agents

- **How does this affect global social welfare?**
- **How to differentiate?**

MATCH CADENCE

How quickly do new edges form?

How frequently does a market clear?

Is clearing centralized or decentralized?

Can agents reenter the market?

INTER-AGENT EFFECTS

Matching market literature focuses on maximizing the sum of the utility of individual matches (subject to constraints).

- Not always the right idea!

Say you are a firm hiring workers:
what is your goal?

Maximize the number of open positions filled …

… with "good" candidates …

… subject to fairness constraint(s) …

… and such that the entire hired cohort works well together!

Use data & optimization – alongside human domain expertise – to learn matching policies

Strong theoretical underpinnings provide design guidance & runtime guarantees

WARM UP

DIVERSE WEIGHTED BIPARTITE B-MATCHING

BIPARTITE *B*-MATCHING

In traditional matching, any vertex can be matched at most once

***b*-matching: given G = (V,E), and a length-|V| vector *b* of nonnegative integers …**

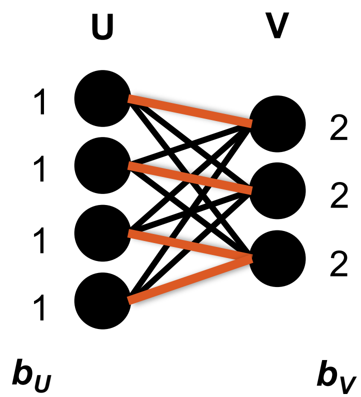
- Any vertex *i* can be matched at most *b*(*i*) times
- Generalizes traditional matching: ***b*** = **1**

Bipartite *b*-matching: given bipartite graph G = (U,V,E) ...

- PTIME for maximum cardinality/weight [Kleinschmidt 1995, & earlier]

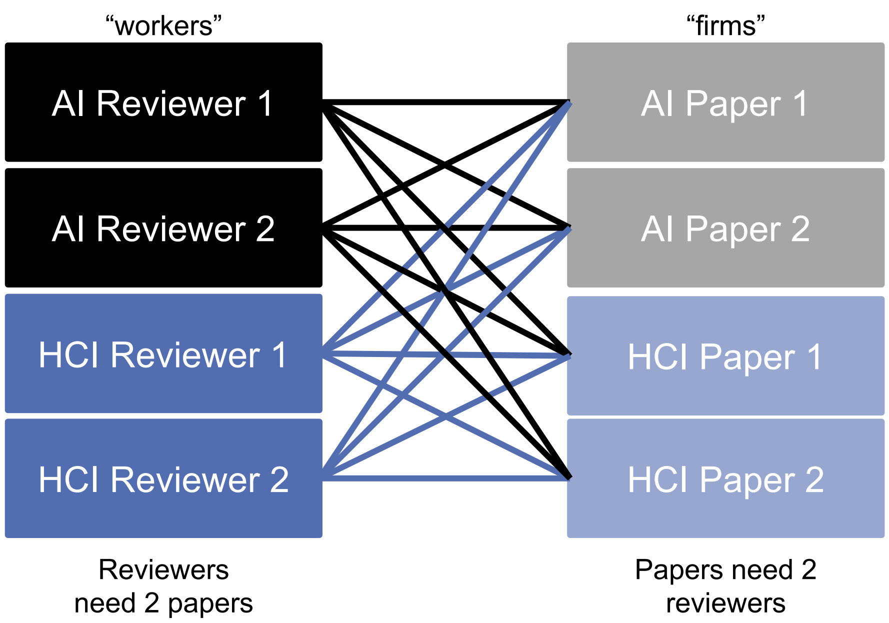
Further generalization: lower and upper bounds

- Vertex *i* must be matched at least $b_-(i)$, and at most $b_+(i)$, times
- NP-hard, even for existence



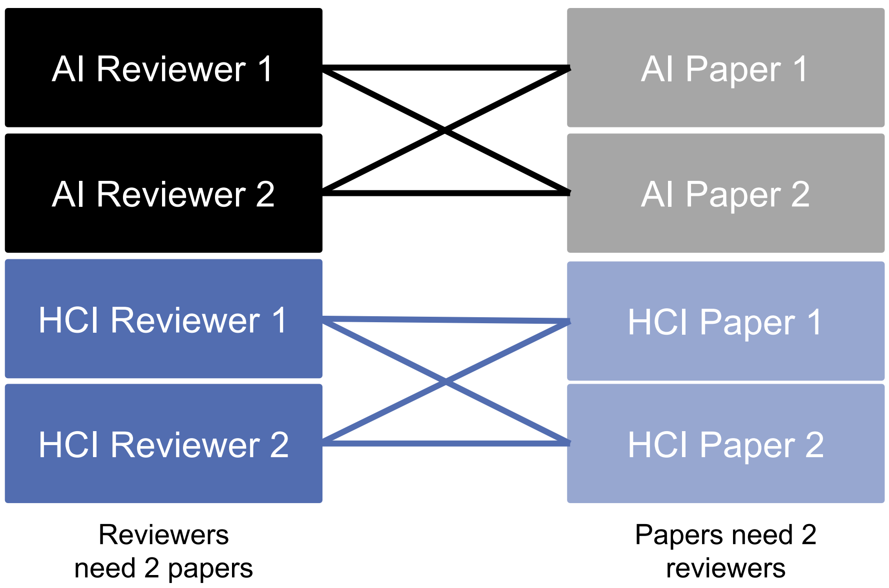
DIVERSITY IN MATCHING MARKETS

New goal: provide "good" coverage over different classes of items or agents



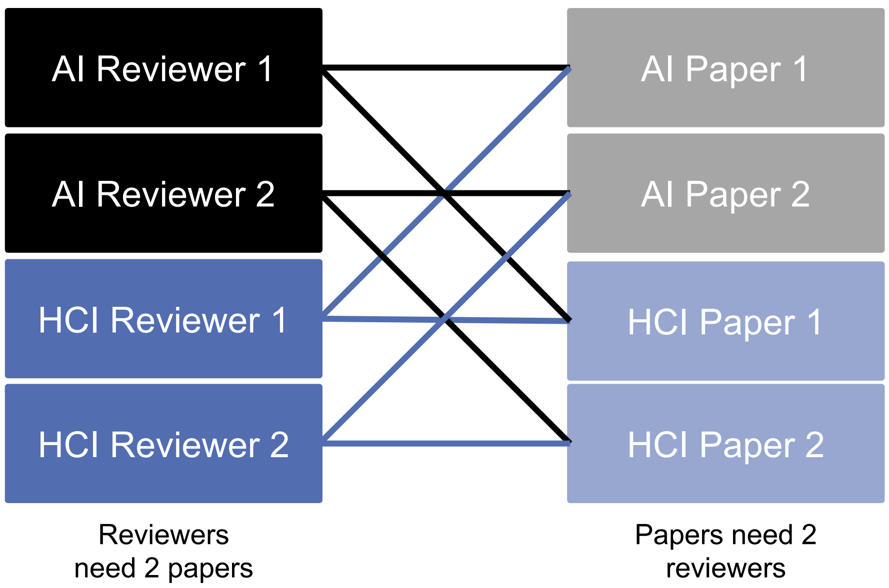
DIVERSITY IN MATCHING MARKETS

Maximum weighted matching will treat individual reviewer matchings as independent of the full review set for a paper



DIVERSITY IN MATCHING MARKETS

Maximum diverse weighted matching will balance individual quality with the diversity of opinion in the paper review set



HOW TO DEFINE DIVERSITY?

Given *K* classes on one side of the market …

- **{AI, HCI, Systems, Theory} paper classes → *K* = 4**
- **… want marginal gain of same-class matches to decrease.**

$$f_j(S_j) = \sum_{k=1}^{K} \sqrt{\sum_{\{i \mid i\in P_k \;\wedge\; (i,j)\in S_j\}} w_{i,j}}$$

Si e node *j*

r each cluster *k* on e opposing side of the market …

… the more times node *j* is matched to nodes in cluster *k*, the lower the gain

Full Objective

$$\max_{S\in\mathcal{S}} \sum_{i\in\mathrm{Left}} f_i(S_i) + \sum_{j\in\mathrm{Right}} f_j(S_j)$$

SOLVING THIS PROBLEM

Basic maximum weight bipartite matching: PTIME

Maximum weight bipartite b-matching (lower+upper): NP-hard [Chen et al. '16]

- Integer linear program (so, ~solvable)

Our problem: at least as hard ☹

- Mixed integer quadratic program (so, harder)
- (Also, the program is enormous)

One can show that an obvious PTIME greedy algorithm:

- Guarantees 1 – 1/e of optimality (for many cases)!
- Open question for the general case.

ENTROPY GAIN & THE PRICE OF DIVERSITY

We use entropy to measure the gain in diversity:

- Entropy is zero if all matches come from the same cluster
- Entropy is maximized if matches are "spread evenly" across clusters
- (Edge weights, aka individual match quality, affects this.)

Entropy gain: relative gain in entropy compared to max weight

Price of diversity: relative loss in efficiency when compared to a maximum weight (aka, efficient) matching

- Want: no price of diversity with high gain in entropy!
- One can show the price of diversity can be very bad in theory ☹.

BUT WHAT ABOUT IN PRACTICE?

MovieLens 1M dataset [Harper&Konstan '16]

- One million ratings of movies (we use a standard collaborative recommender system to fill in blanks)

SIGIR and KDD reviewer bidding [Karimzadehgan&Zhai '09, Sugiyama&Kan '10]

	Solve to optimality		Solve approximately	
Dataset	**PoD**	**EG**	**PoD**	**EG**
MovieLens	0.01	1.45	0.01	1.45
SIGIR	0.08	1.63	0.17	1.60
KDD	0.06	4.28	0.07	4.28

INITIAL TAKEAWAY

Assumes a well-defined objective …

We can greatly increase the diversity of a recommended matching at almost no cost to overall efficiency.

(Not in theory, but in practice, and in the **static case** …)

Really, no uncertainty!

No uncertainty!

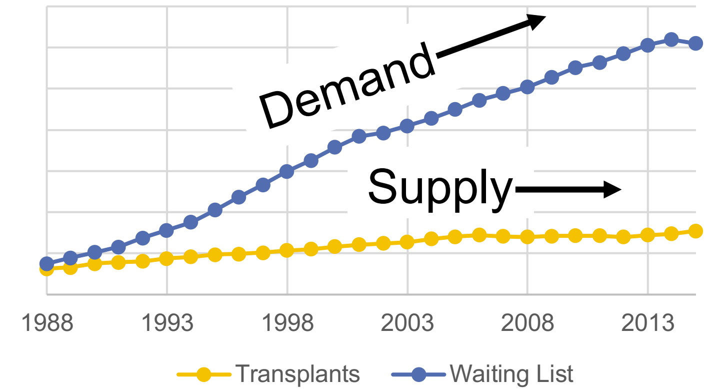
THESE THREE TALKS

- **Five dimensions of matching market design:**
  - Managing short-term uncertainty
  - Balancing equity & efficiency
  - Combining human input and optimization
  - Incentives & mechanism design
  - Non-linear objectives such as diversity
- ***(Each is supported by my work with a nationwide kidney exchange and in hiring.)***
- **Also, some open problems!**

Covers recent and ongoing work – talk to me for details!
Publications: jpdickerson.com/pubs.html

RUNNING EXAMPLE: ORGAN ALLOCATION

KIDNEY TRANSPLANTATION



- **US waitlist: about 100,000**
  - 35,587 added in 2017
- **4,044 people died while waiting**
- **14,022 people received a kidney from the deceased donor waitlist**
- **5,794 people received a kidney from a living donor**
  - Some through **kidney exchanges**! [Roth et al. 2004]
  - This talk: experience with UNOS national kidney exchange (and some data from the NHS NLDKSS)

UNOS DONATE LIFE

NHS

TRIED-AND-TRUE: DECEASED-DONOR ALLOCATION

Online bipartite matching problem:

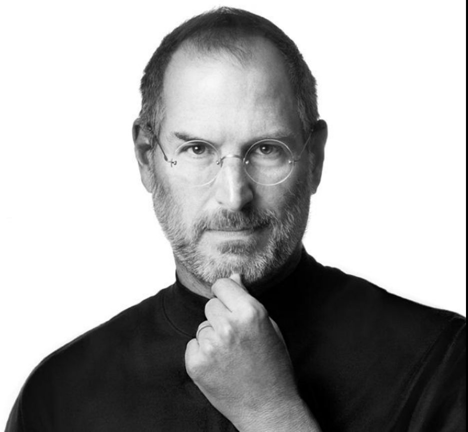
- Set of patients is known (roughly) in advance
- Organs arrive and must be dispatched quickly

Constraints:

- Locality: organs only stay good for 24 hours
- Blood type, tissue type, etc.

Who gets the organ? Prioritization based on:

- Age?
- QALY maximization?
- Quality of match?
- Time on the waiting list?



EFFICIENT, ETHICAL, & FAIR MATCHING MARKET DESIGN VIA OPTIMIZATION

JOHN P DICKERSON

Talk #2

COMPUTER SCIENCE
UNIVERSITY OF MARYLAND

The Cornell, Maryland, Max Planck
Pre-doctoral Research School
August 7, 8, & 10, 2018

QUICK ASIDE: MORE INFO

Many people asked: "How did you find out about this type of research?"

AAAI/ACM Conference on Artificial Intelligence, Ethics, & Society (AIES)

- http://www.aies-conference.com/

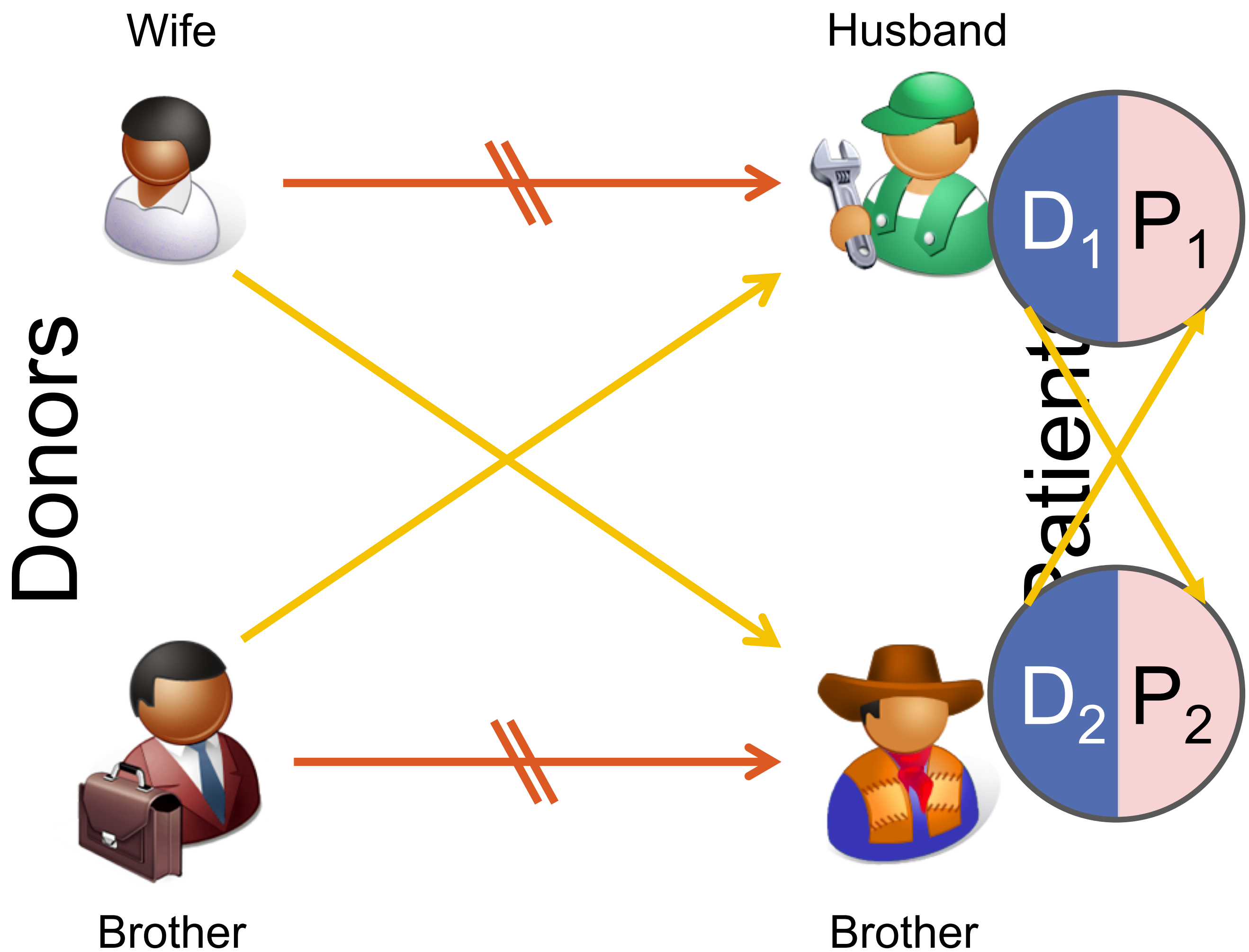
ACM Conference on Fairness, Accountability, and Transparency (FAT*)

- https://fatconference.org/

ACM Conference on Economics and Computation (EC)

- http://www.sigecom.org/ec18/

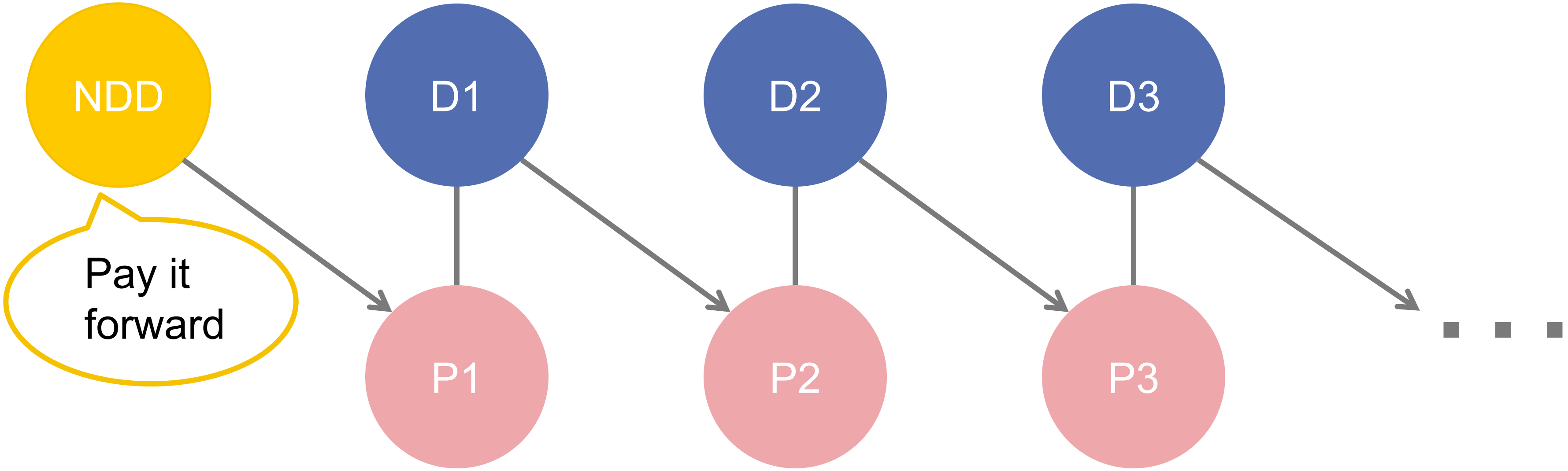
KIDNEY EXCHANGE



(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!**

REAL-WORLD IMPACT

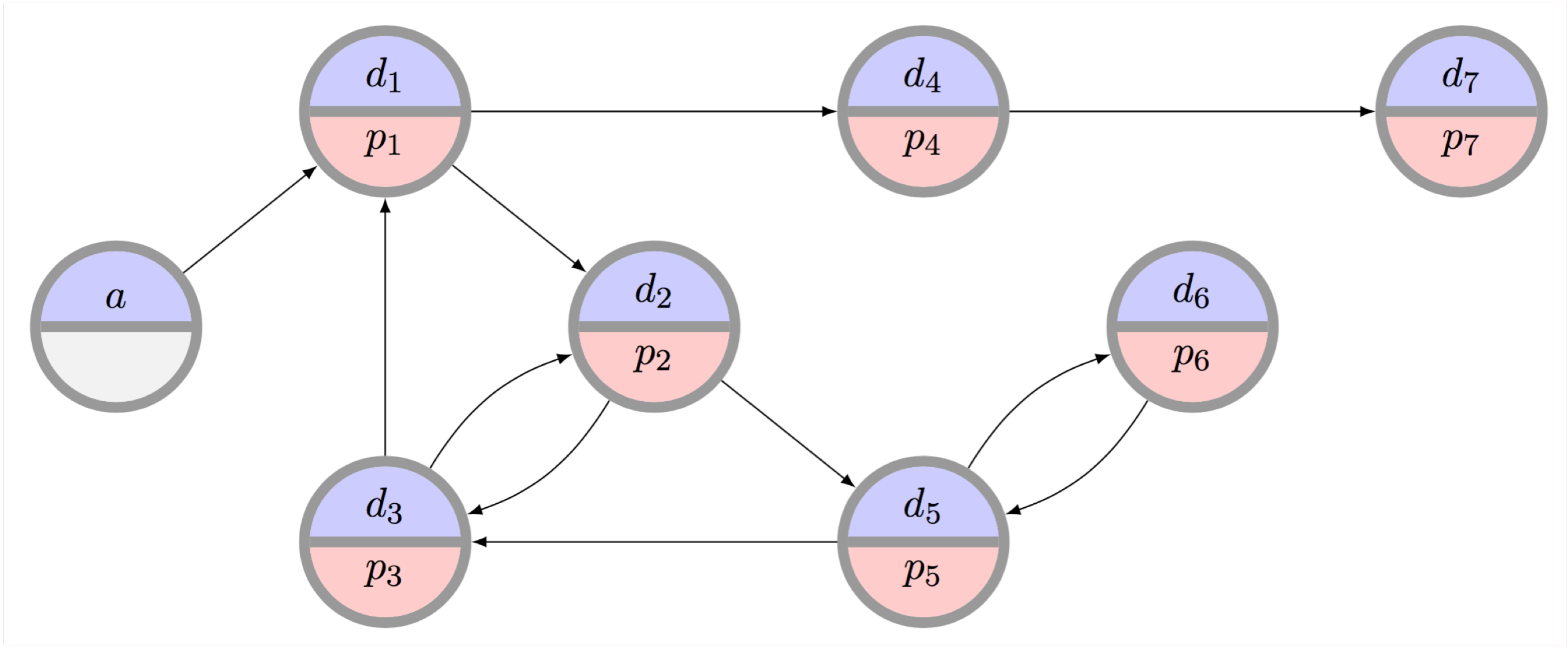
Kidney exchange is only a decade young, but already accounts for >10% of living donations in the United States

- Now a worldwide phenomenon (AU, CA, IL, PT, TR, UK, …)
- (Slowly) moving toward organized international exchange

Worked extensively with the United Network for Organ Sharing (UNOS) US nationwide kidney exchange!

- 153+ transplant centers (roughly 66% of the US)
- Completely autonomous biweekly match runs
- Only automated exchange in the US

THE CLEARING PROBLEM

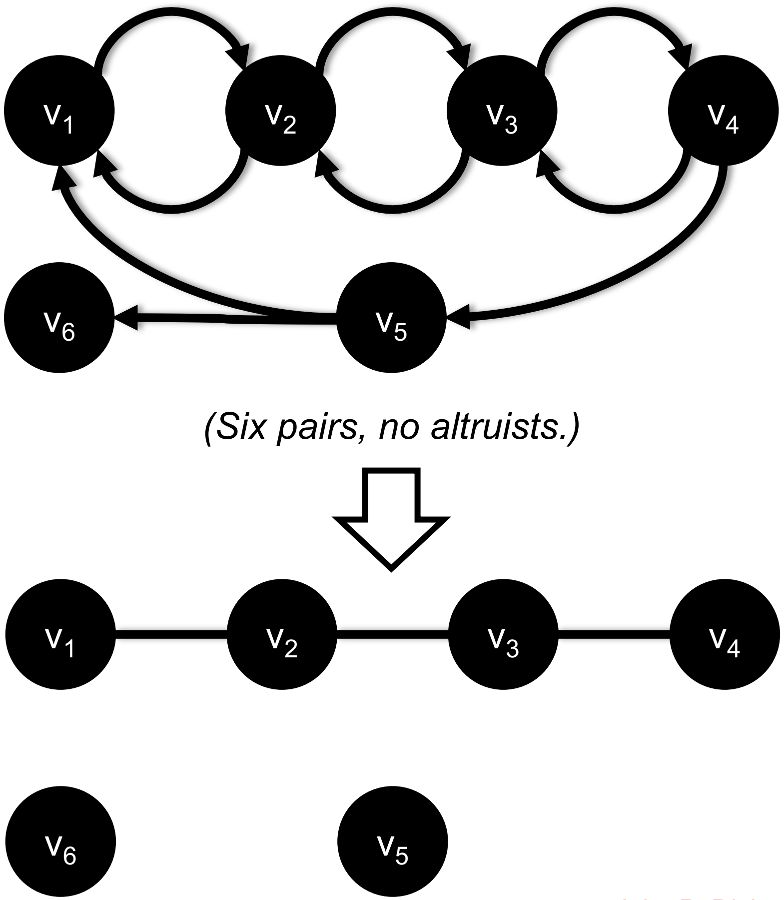


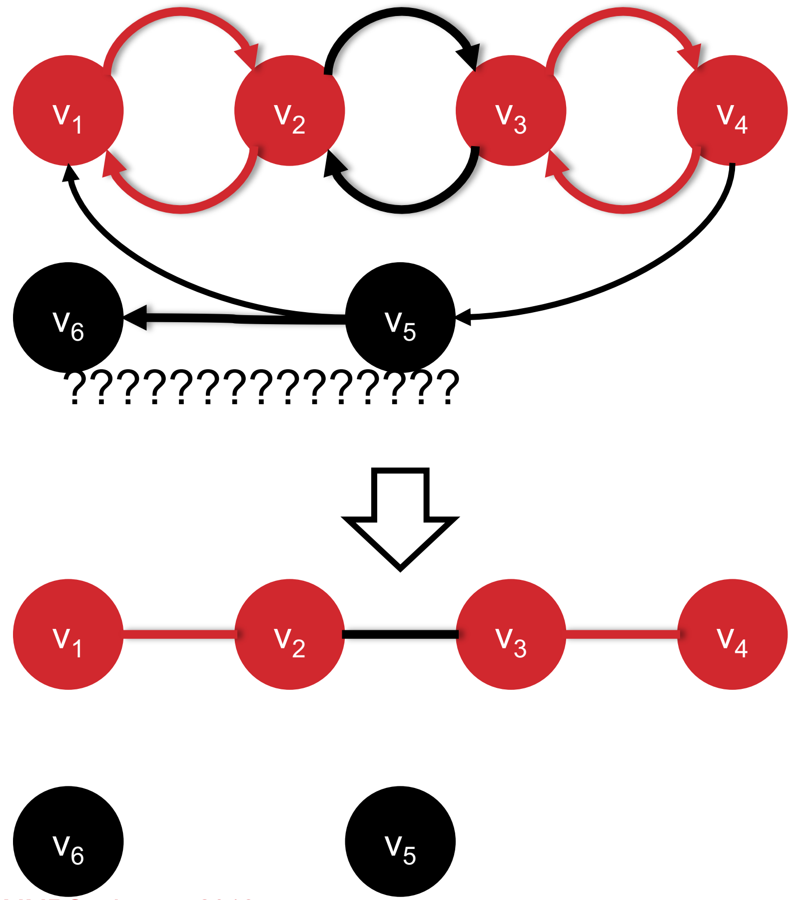
The clearing problem is to find the "best" disjoint set of cycles of length at most *L*, and chains

- Typically, $2 \leq L \leq 5$ for kidneys (e.g., L=3 at UNOS)
- NP-hard (for L>2) in theory, really hard in practice [Glorie et al. 2014, Anderson et al. 2015, Plaut et al. 2016, Dickerson et al. 2016 ...]
  [Abraham et al. 07, Biro et al. 09]

SPECIAL CASE: *L* = 2

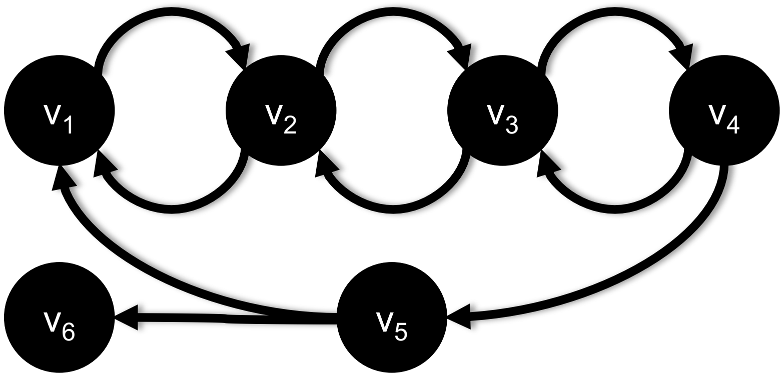
PTIME: translate to maximum matching on undirected graph



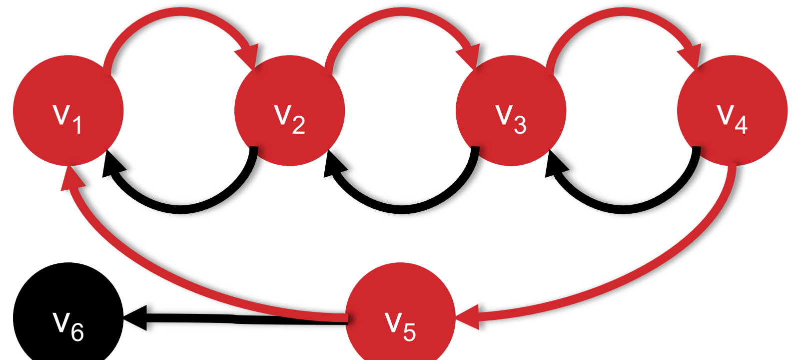


SPECIAL CASE: $L = \infty$

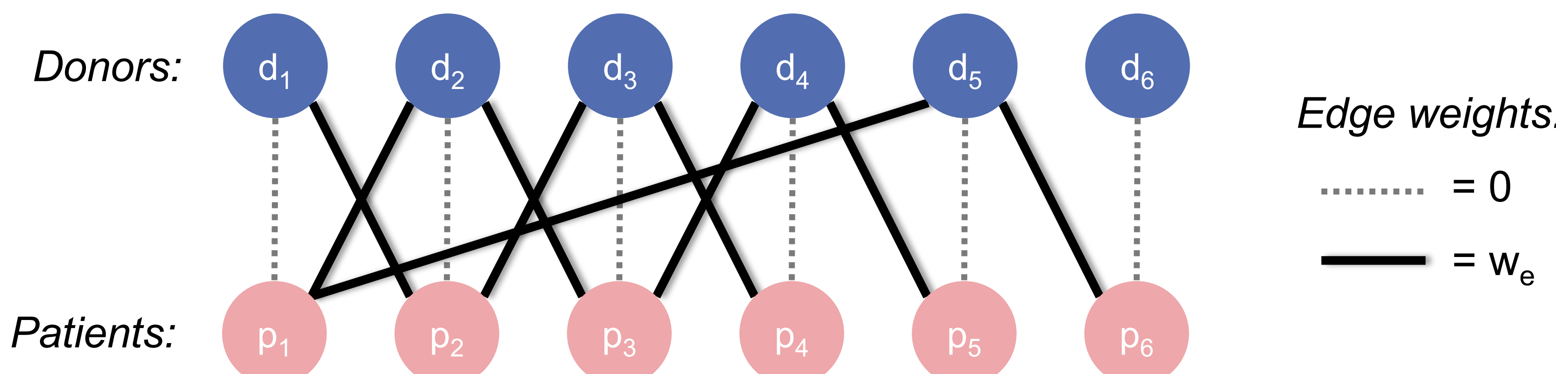
PTIME via formulation as maximum weight perfect matching



(Six pairs, no altruists.)



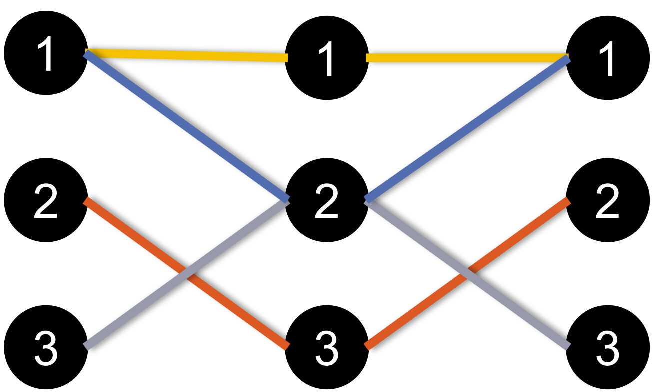
???????????????



GENERAL CASE: *L* = ?

NP-hard via reduction from 3D-matching:

- **Given disjoint sets X, Y, Z of size *q* …**
- **... and a set of triples T ⊆ X x Y x Z ...**
- **... is there a disjoint subset M ⊆ T of size *q*?**



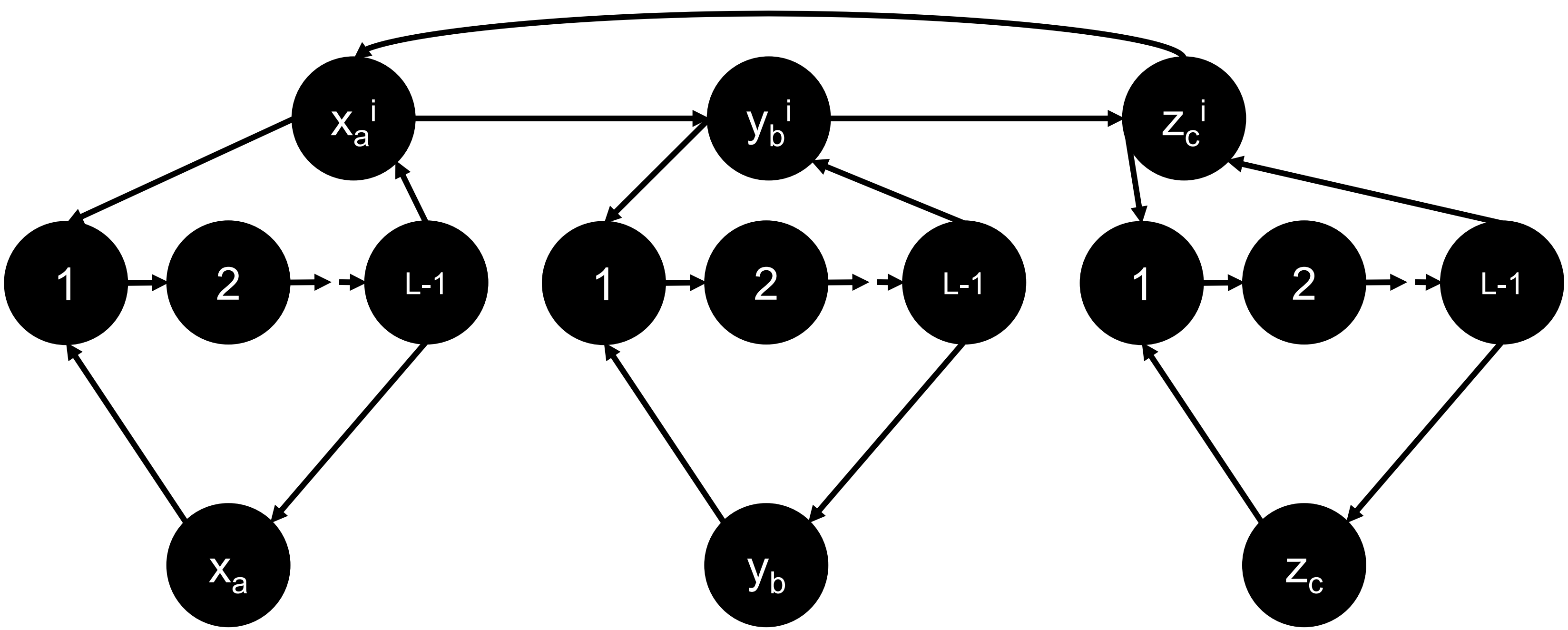
T = {
(1,1,1), ✓
(2,3,2), ✓
(1,2,1),
(3,2,3), ✓
}

???????????????

GENERAL CASE: *L* = ?

Construct a gadget for each $t_i = \{x_a, y_b, z_c\}$ in T

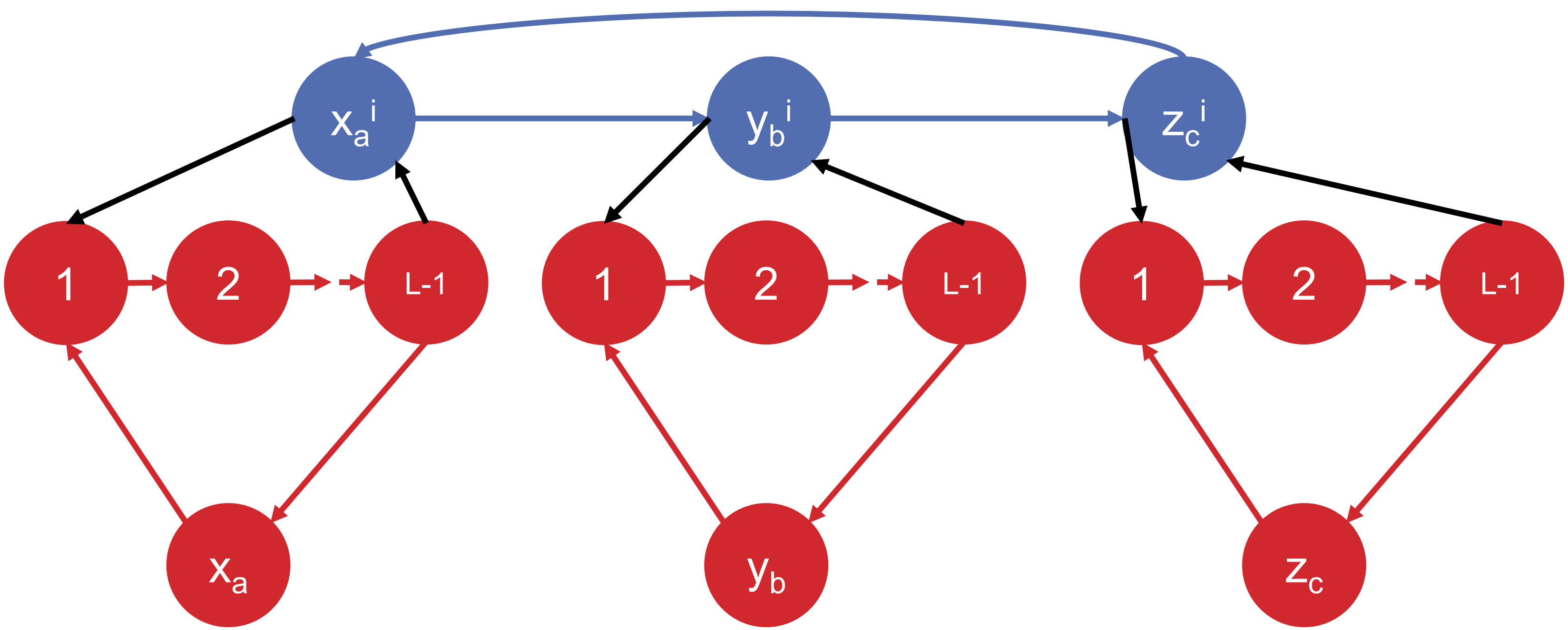
- Gadgets intersect only on vertices in X U Y U Z



GENERAL CASE: *L* = ?

M is perfect matching → construction has perfect cycle cover.

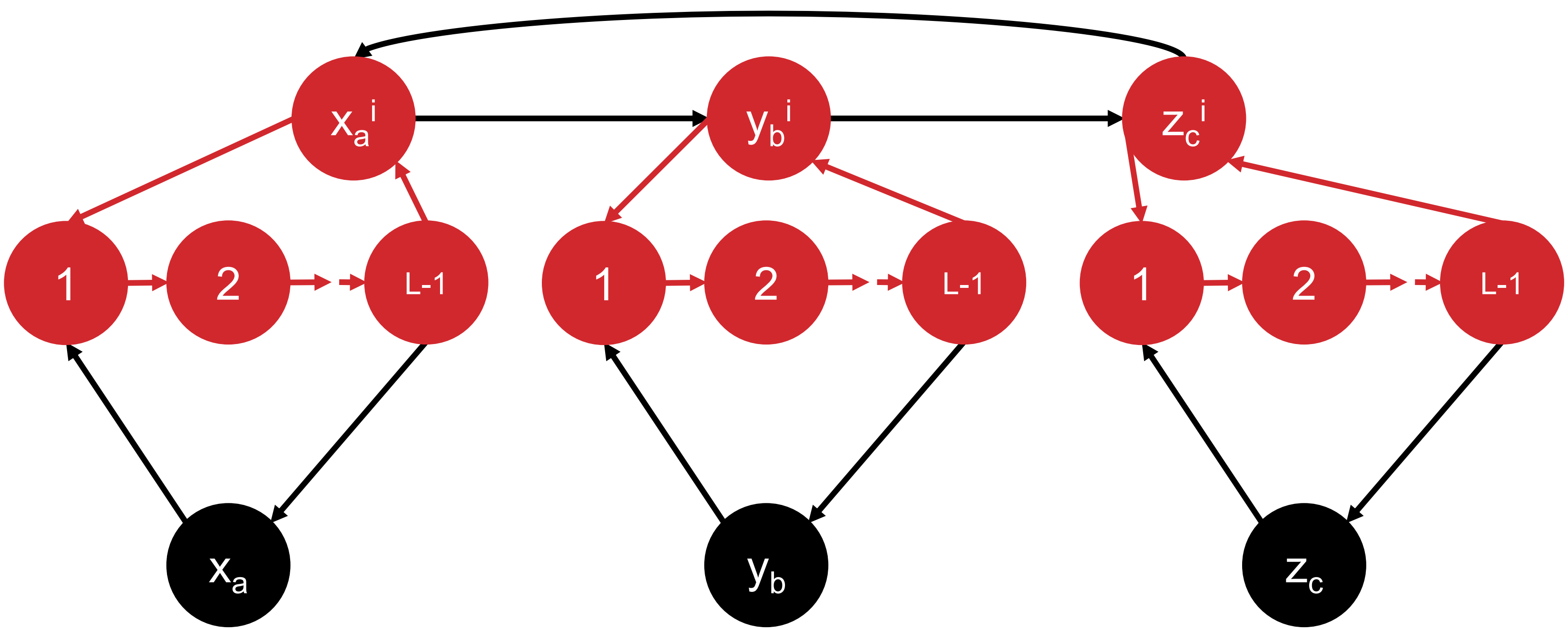
For t_i in *T*:



GENERAL CASE: *L* = ?

M is perfect matching → construction has perfect cycle cover.

For t_i not in *T*:



GENERAL CASE: *L* = ?

We have a perfect cycle cover → *M* is a perfect 3D matching

- Construction only has 3-cycles and *L*-cycles
- Short cycles (i.e., 3-cycles) are disjoint from the rest of the graph by construction

Thus, given a perfect cover (by assumption):

- Widgets either contribute according to t_i in M …
- … or t_i not in M.

Thus there is a perfect matching in the original 3D matching instance.

HOPELESS ...?

A SIMPLE INTEGER PROGRAM

("Best" = max weight, myopic matching)

[Roth et al. 04, 05, Abraham et al. 07]

Binary variable x_c for each feasible cycle or chain c

Maximize

$$u(M) = \Sigma\, w_c\, x_c$$

Subject to

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

"SIMPLE" ...?

BASIC APPROACH #1: THE EDGE FORMULATION

[Abraham et al. 2007]

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)

STATE OF THE ART FOR EDGE FORMULATION

[Anderson et al. PNAS-2015]

Builds on the prize-collecting traveling salesperson problem [Balas Networks-89]

- 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)

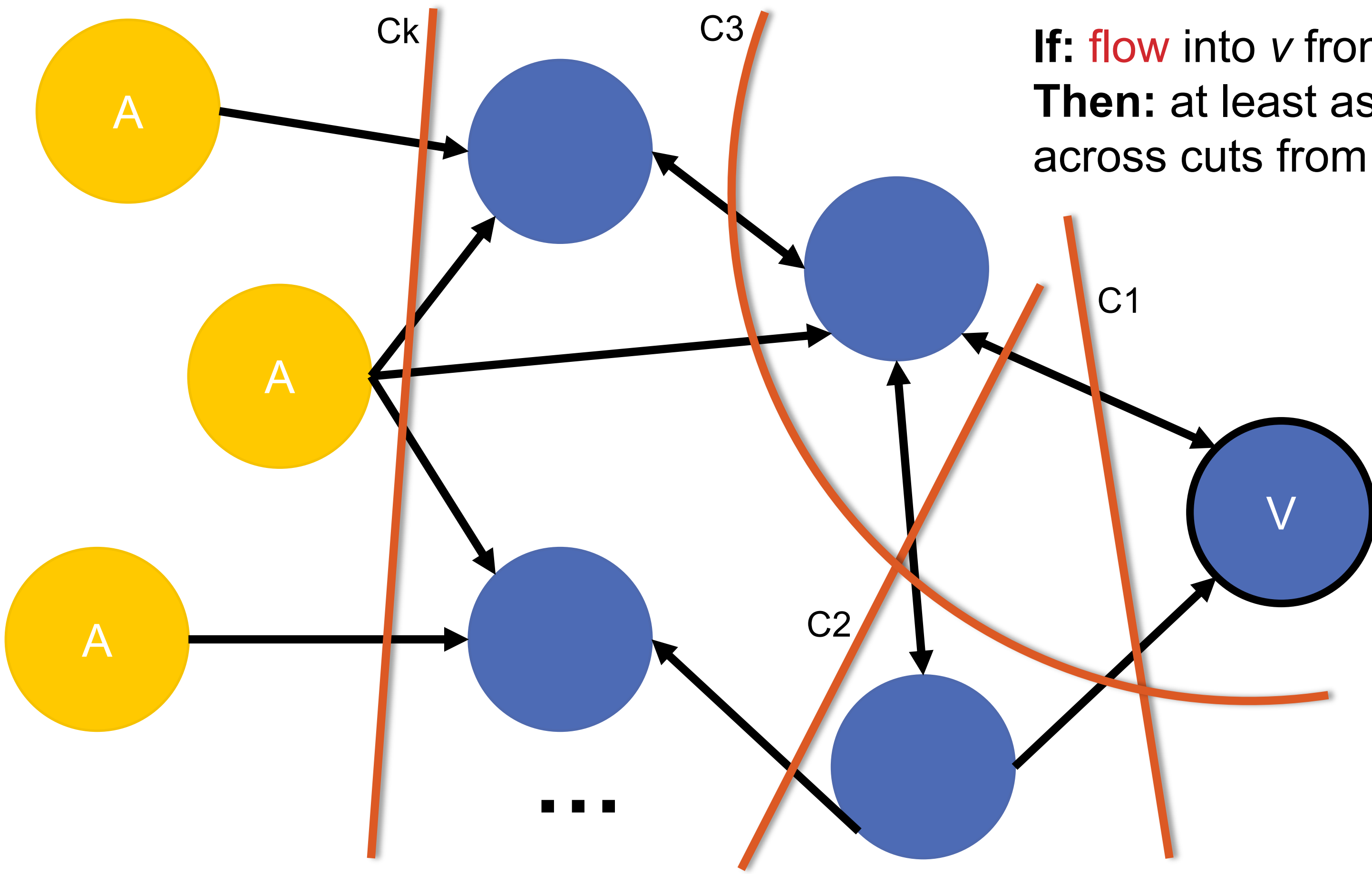
They maintain decision variables for all cycles of length at most *L*, but build chains in the final solution from decision variables associated with individual edges

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

BEST EDGE FORMULATION

[Anderson et al. 2015]



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

BASIC APPROACH #2: THE CYCLE FORMULATION

[Roth et al. 2004, 2005, Abraham et al. 2007]

Binary variable x_c for each feasible cycle or chain c

Maximize

$$u(M) = \Sigma\, w_c\, x_c$$

Subject to

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

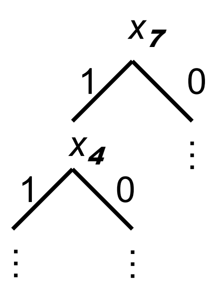
SOLVING THE CYCLE FORMULATION IP

Too large to write down

- $O(\max\{\ |P|^L, |A||P|^{K-1}\ \})$ variables
- $|A| = 5$, $|V|=300$, $L=3$, $K=20$ … $|A||P|^{K-1} \approx 5 \times 10^{47}$

Approach: branch-and-price [Barnhart et al. 1998]:

- 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, and/or 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

- We do this intelligently by solving the pricing problem

Repeat until OPT(*P*') = OPT(*P*)

DFS TO SOLVE PRICING PROBLEM

[Abraham et al. EC-07]

Pricing problem:

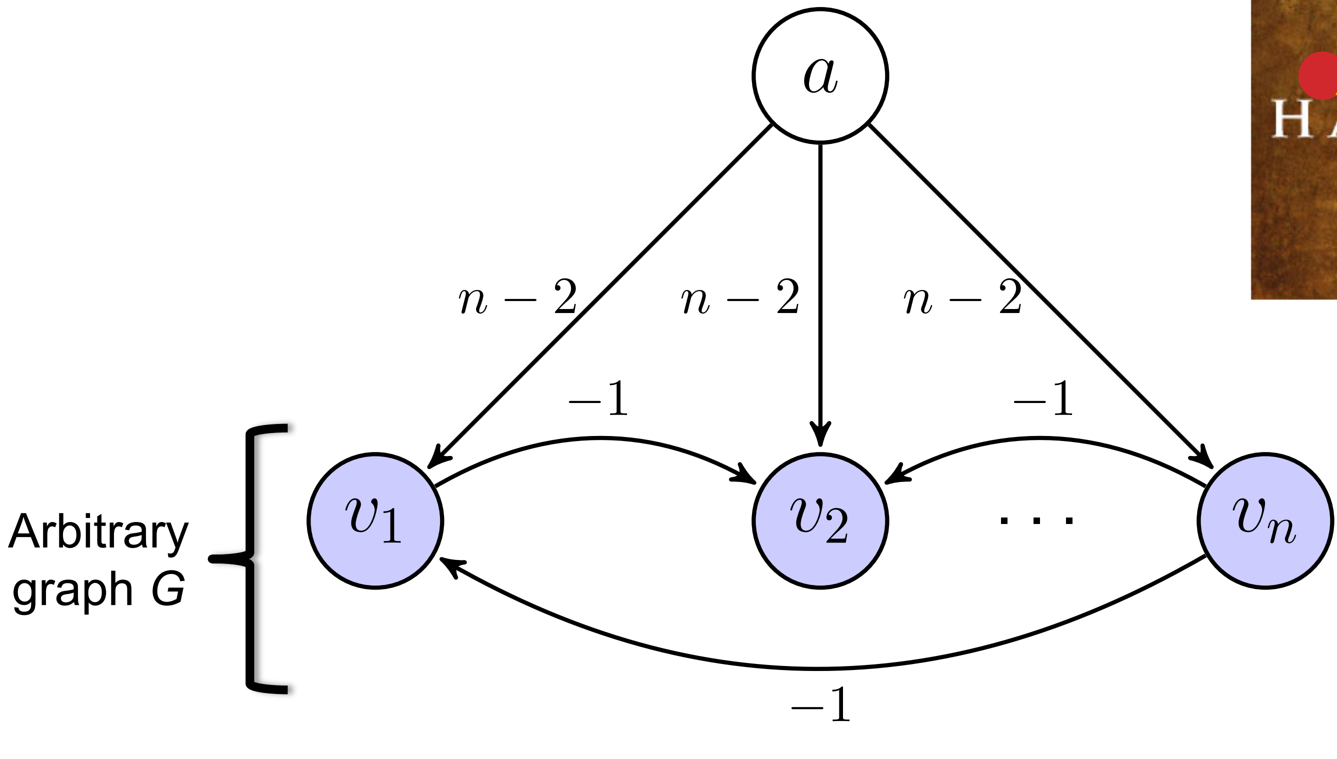
- Optimal dual solution $\boldsymbol{\pi}^*$ to reduced model
- Find non-basic variables with **positive price** (for a maximization problem)
    - 0 < weight of cycle – sum of duals in $\boldsymbol{\pi}^*$ of constituent vertices
    - Positive price for cycle → dual constraint is violated
    - No positive price cycles → no dual constraints violated

First approach [Abraham et al. EC-2007] explicitly prices all feasible cycles and chains through a DFS

- Can speed this up in various ways, but proving **no positive price cycles exist** still takes a long time

GENERAL PRICING OF CYCLES & CHAINS IS NP-HARD [Plaut et al. arXiv:1606.00117]

Reduce from Hamiltonian path



COMPARISON

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 $(L+1)/2$ – with ½ on each edge

Recent work focuses on balancing tight LP relaxations and model size [Constantino et al. 2013, Glorie et al. 2014, Klimentova et al. 2014, Alvelos et al. 2015, Anderson et al. 2015, Mak-Hau 2015, Manlove&O'Malley 2015, Plaut et al. 2016, …]**:**

- Newest work: compact formulations, some with tightest relaxations known, all amenable to failure-aware matching

COMPACT FORMULATIONS [Constantino et al. EJOR-14]

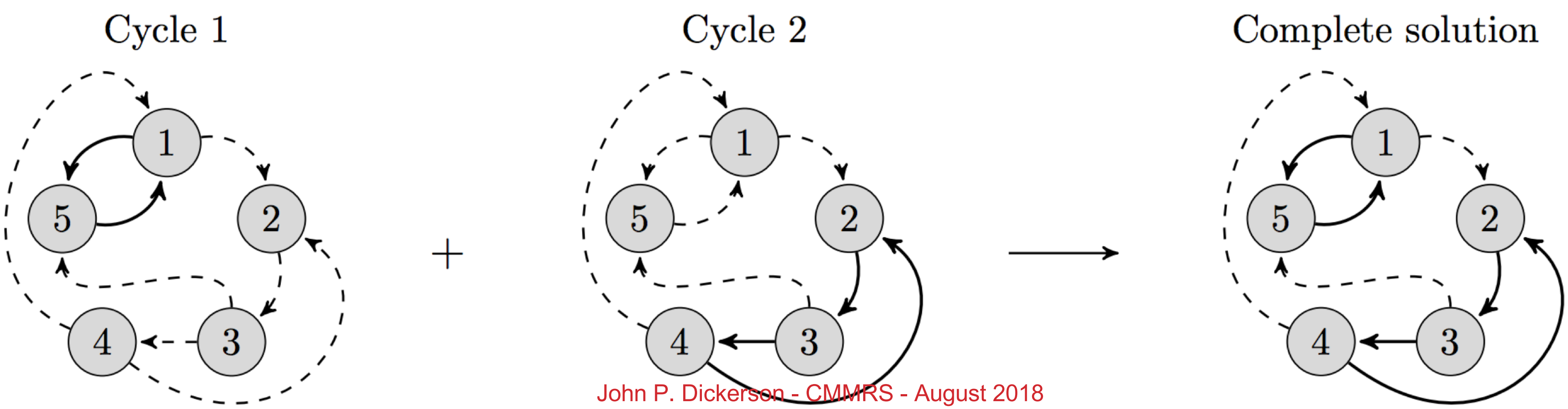
Previous models: exponential #constraints (CG methods)
or #variables (B&P methods)

Let F be upper bound on #cycles in a final matching

Create F copies of compatibility graph

Search for a single cycle or chain in each copy

- (Keep cycles/chains disjoint across graphs)



COMPACT FORMULATIONS

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

$$\text{maximize} \quad \sum_f \sum_{(i,j)\in A} w_{ij} x_{ij}^f \qquad \text{1A}$$

$$\text{subject to} \quad \sum_{j:(j,i)\in A} x_{ij}^f = \sum_{j:(i,j)\in A} x_{ij}^f \qquad \forall i \in V, \forall f \in \{1,\ldots,F\} \qquad \text{1B}$$

$$\sum_f \sum_{j:(i,j)\in A} x_{ij}^f \leq 1 \qquad \forall i \in V \qquad \text{1C}$$

$$\sum_{(i,j)\in A} x_{ij}^f \leq k \qquad \forall f \in \{1,\ldots,F\} \qquad \text{1D}$$

$$x_{ij}^f \in \{0,1\} \qquad \forall (i,j) \in A, \forall f \in \{1,\ldots,F\} \qquad \text{1E}$$

1A: max edge weights over all graph copies

1B: give a kidney <-> get a kidney within that copy

1C: only use a vertex once

1D: cycle cap

Polynomial #constraints and #variables!

PIEF: A COMPACT MODEL FOR CYCLES ONLY

[Dickerson Manlove Plaut Sandholm Trimble EC-16]

Builds on Extended Edge Formulation of Constantino et al.

- **O(|*V*|) copies of graph, 1 binary variable per edge per copy**
- **Enforce at most one cycle per graph copy used**
- **Track positions of edges in cycles for LP tightness**

THEOREM

The tightest known non-compact LP relaxation

$Z_{CF} = Z_{PIEF}$

(disallowing chains)

(EC-16 paper also presents HPIEF, which is a compact formulation for cycles and chains, but with weaker Z_{HPIEF})

PICEF: POSITION-INDEXED CHAIN-EDGE FORMULATION

In practice, cycle cap L is small and chain cap K is large

Idea: enumerate all cycles but not all chains [Anderson et al. 2015]

- That work required $O(|V|^K)$ constraints in the worst case
- This work requires $O(K|V|) = O(|V|^2)$ constraints

MAIN IDEA

Track not just if an edge is used in a chain, but where in a chain an edge is used.

For edge (i,j) in graph: $K'(i,j) = \{1\}$ if i is an altruist

$K'(i,j) = \{2, \ldots, K\}$ if i is a pair

PICEF: POSITION-INDEXED CHAIN-EDGE FORMULATION

Maximize

$$u(M) = \Sigma_{ij \text{ in } E} \Sigma_{k \text{ in } K'(i,j)} w_{ij} y_{ijk} + \Sigma_{c \text{ in } C} w_c z_c$$

Subject to

$\Sigma_{ij \text{ in } E} \Sigma_{k \text{ in } K'(i,j)} y_{ijk} + \Sigma_{c\,:\,i \text{ in } c} z_c \leq 1$ for every i in *Pairs*

Each pair can be in at most one chain or cycle

$\Sigma_{ij \text{ in } E}\, y_{ij1} \leq 1$ for every i in *Altruists*

Each altruist can trigger at most one chain via outgoing edge at position 1

$\Sigma_{j:ij \text{ in } E}\, y_{ijk+1} - \Sigma_{j:ji \text{ in } E \,\wedge\, k \text{ in } K'(j,i)}\, y_{jik} \leq 0$ for every i in *Pairs* and k in $\{1, \ldots, K\text{-}1\}$

Each pair can be have an outgoing edge at position k+1 in a chain iff it has an incoming edge at position k in a chain

WHAT IF THERE ARE STILL TOO MANY VARIABLES?

In particularly dense graphs or if, in the future, longer cycle caps are allowed, PICEF may need too many cycle variables

Solve via branch and price by storing only a subset of columns in memory, then solving pricing problem

- Search for variables with positive price, bring into model
- Previously: that search is exponential in chain cap [Abraham et al. 2007, Glorie et al. 2014, Plaut et al. 2016]
- General: pricing chains & cycle is NP-hard [arXiv:1606.00117]

But we only need to price cycles, not chains!

P R I C I N G

PICEF is the first branch-and-price-based model with provably correct polynomial-time pricing

POLYNOMIAL-TIME CYCLE PRICING

[Glorie et al. MSOM-2014, Plaut et al. AAAI-2016]

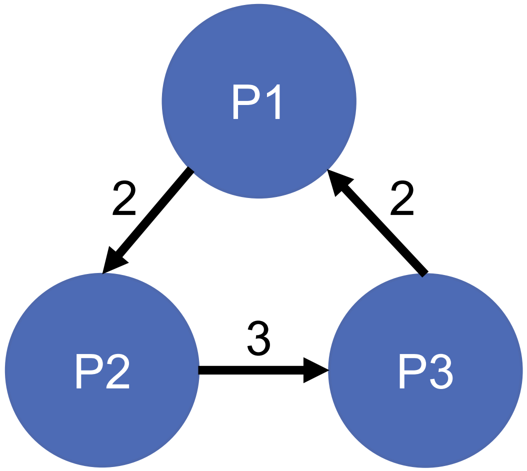
Solve a structured problem that implicitly prices variables

- Variable = x_c for cycle (not chain) c
- Price of x_c = $w_c - \Sigma_{v \text{ in } c}\, \delta_v$

Example

- Price: $\underbrace{(2+3+2)}_{w_c} - (\delta_{P1}+\delta_{P2}+\delta_{P3})$

$$= \Sigma_{e \text{ in } c}\, w_e - \Sigma_{v \text{ in } c}\, \delta_v$$
$$= \Sigma_{(u,v) \text{ in } c}\, [w_{(u,v)} - \delta_v]$$



Idea: 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'***

ADAPTED BELLMAN-FORD PRICING FOR CYCLES ONLY

[Glorie et al. MSOM-2014, Plaut et al. AAAI-2016]

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 $O(|V||E|L^2)$**

HOW DO ALL THESE MODELS PERFORM IN PRACTICE?

Test on real and simulated match runs from:

- US UNOS exchange: 143+ transplant centers
- UK NLDKSS: 20 transplant centers

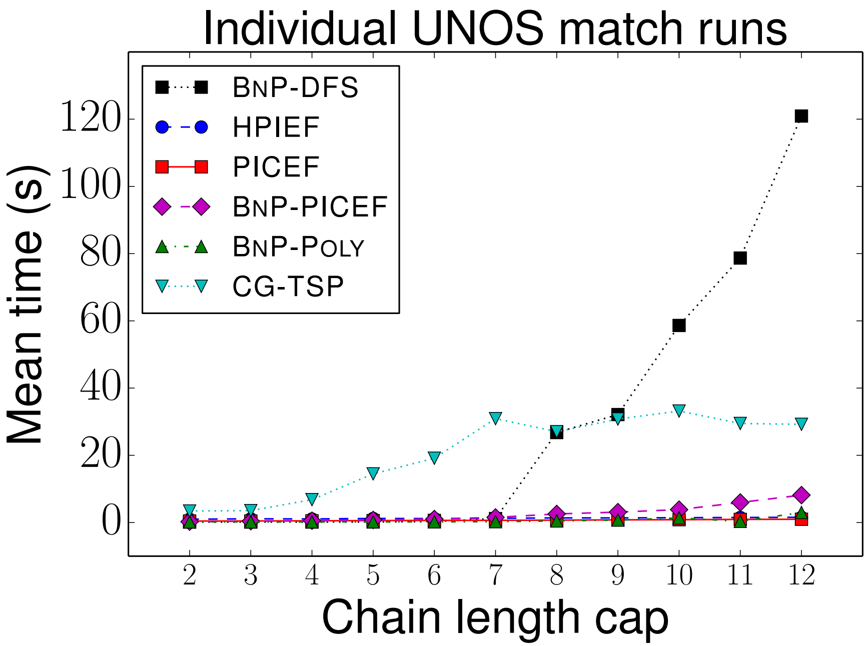
Following are tests against actual code for:

- BnP-DFS [Abraham et al. EC-07]
- BnP-Poly [Glorie et al. MSOM-14, Plaut et al. AAAI-16]
- CG-TSP [Anderson et al. PNAS-15]

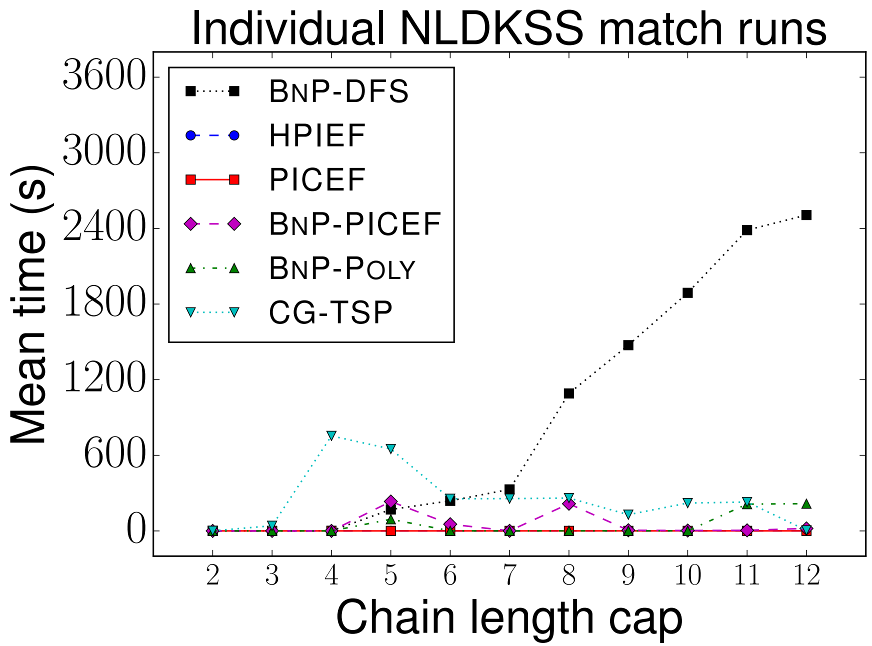
REAL MATCH RUNS

UNOS & NLDKSS

UNOS: 286 match runs

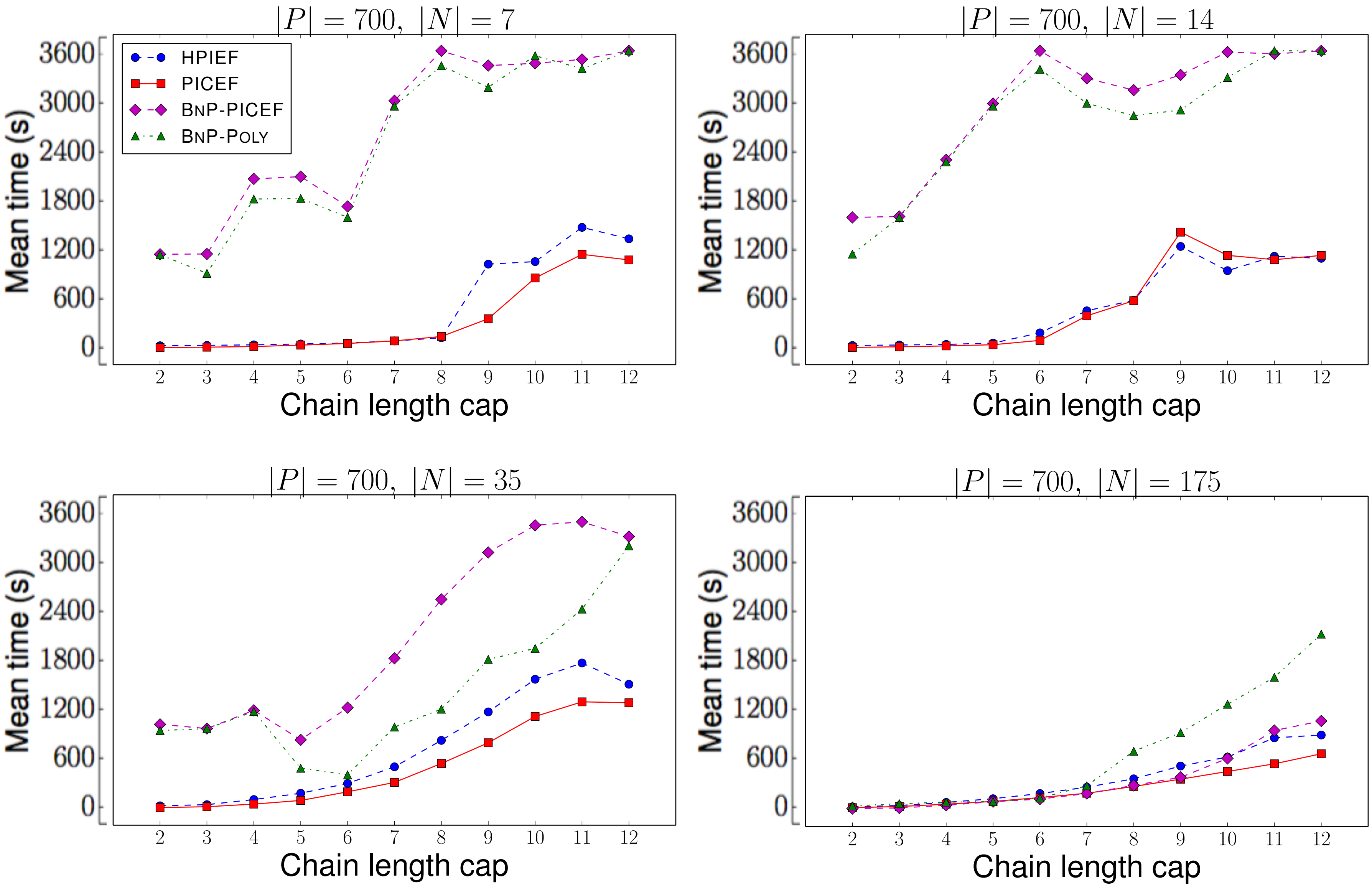


NLDKSS: 17 match runs



GENERATED DATA

$|P|$=700, INCREASING %ALTRUISTS



Solvers that are not shown timed out (within one-hour period).

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 (for humans)

Want humans in the loop at a **high level**
(and then CS/Opt handles the implementation)

EFFICIENT, ETHICAL, & FAIR MATCHING MARKET DESIGN VIA OPTIMIZATION

JOHN P DICKERSON

Talk #3

COMPUTER SCIENCE
UNIVERSITY OF MARYLAND

The Cornell, Maryland, Max Planck
Pre-doctoral Research School
August 7, 8, & 10, 2018

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 (for humans)

Want humans in the loop at a **high level**
(and then CS/Opt handles the implementation)

MANAGING SHORT-TERM UNCERTAINTY

[EC-13, EC-15, EC-16, Management Science 2018]

With A. Blum, N. Haghtalab, D. Manlove, B. Plaut, A. Procaccia, T. Sandholm, A. Sharma, J. Trimble

MATCHED ≠ TRANSPLANTED

Only around 10-15% of UNOS matched structures result in an actual transplant

- Similarly low % in other exchanges [ATC 2013]

Many **reasons for this. How to handle?**

One way: encode *probability of transplantation* rather than just feasibility

- for individuals, cycles, chains, and full matchings

FAILURE-AWARE MODEL

Compatibility graph *G*

- Edge (v_i, v_j) if v_i's donor can donate to v_j's patient
- Weight w_e on each edge e

Success probability q_e for each edge e

Discounted utility of cycle *c*

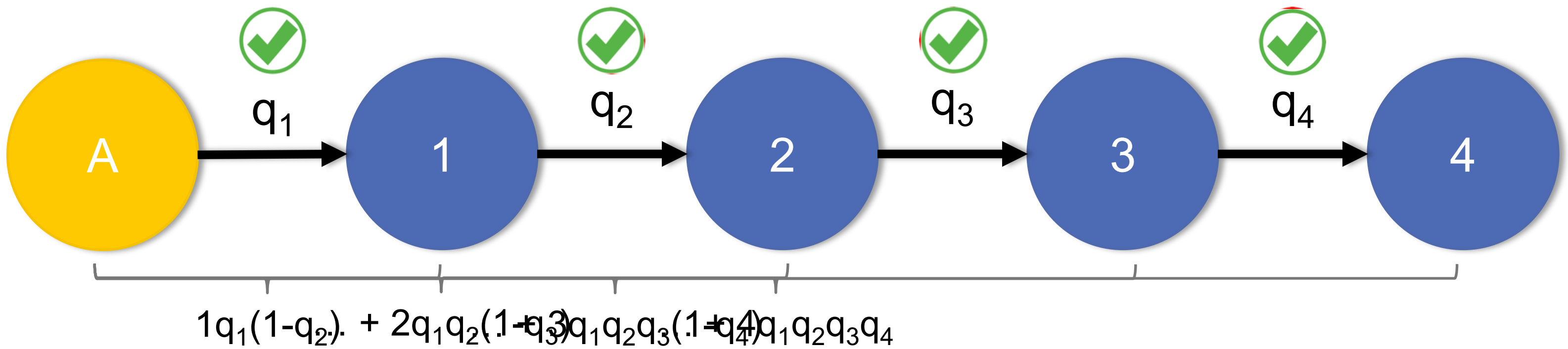
$$u(c) = \sum w_e \cdot \prod q_e$$

Value of successful cycle

Probability of success

FAILURE-AWARE MODEL

Discounted utility of a *k*-chain *c*



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

Exactly first *i* transplants

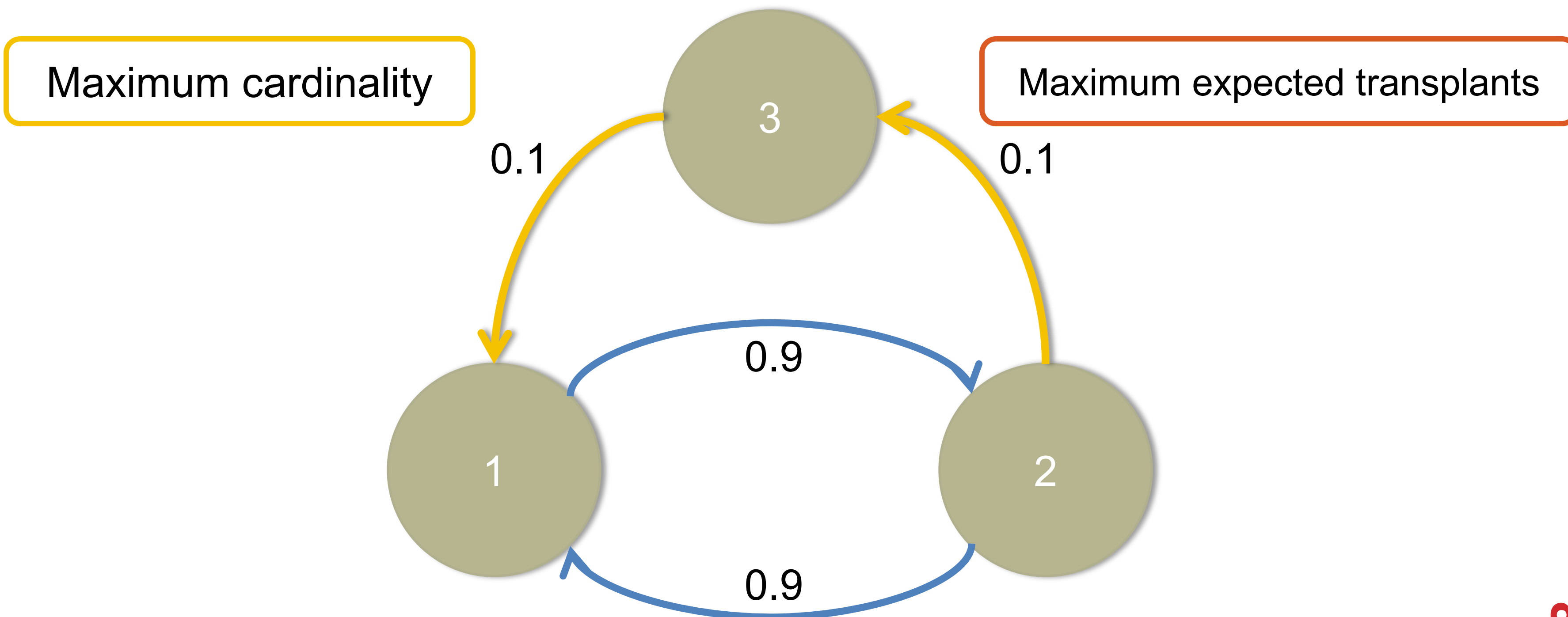
Chain executes in entirety

Cannot simply “reweight by failure probability”

DISCOUNTED CLEARING PROBLEM

(“Best” = max expected cardinality | limited recourse)

Find matching M^* with highest discounted utility



SOLVING THIS NEW PROBLEM

Theorem:
In a sparse random graph model, for any failure probability *p*, w.h.p. there exists a matching that is "linearly better" than *any* max-cardinality matching

Maybe this is a good idea …

Practice: Solved via branch-and-price

- One binary decision variable per cycle/chain
- Upper-bounding is now NP-hard
- Pricing problem is (empirically) much easier

$G(n, t(n), p)$: random graph with

- n patient-donor pairs
- $t(n)$ altruistic donors
- Probability $\Theta(1/n)$ of incoming edges

Constant transplant success probability q

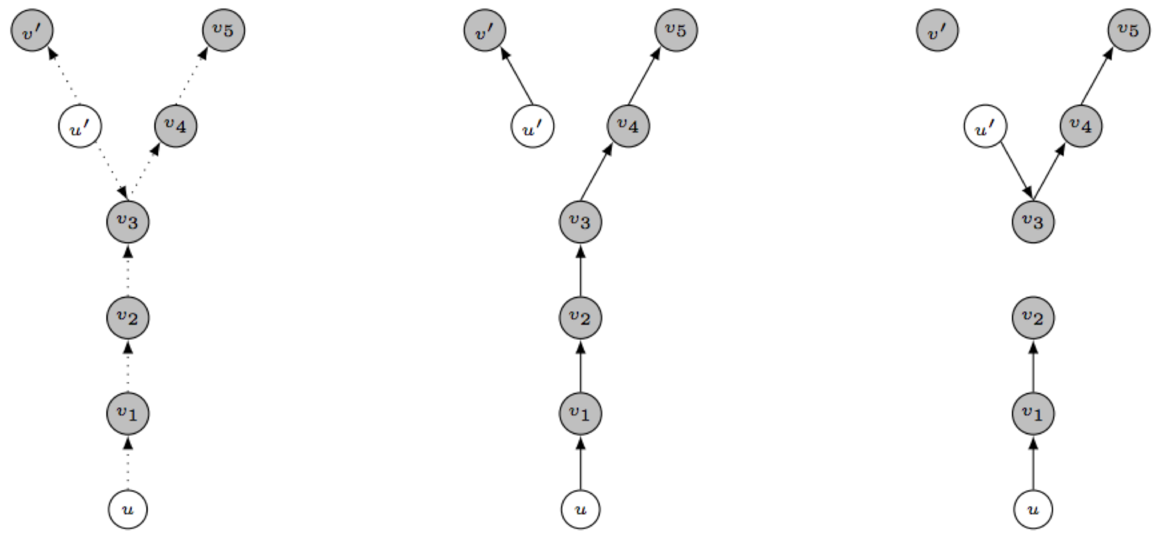
THEOREM

Theorem from last slide, but a little bit more formal:

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

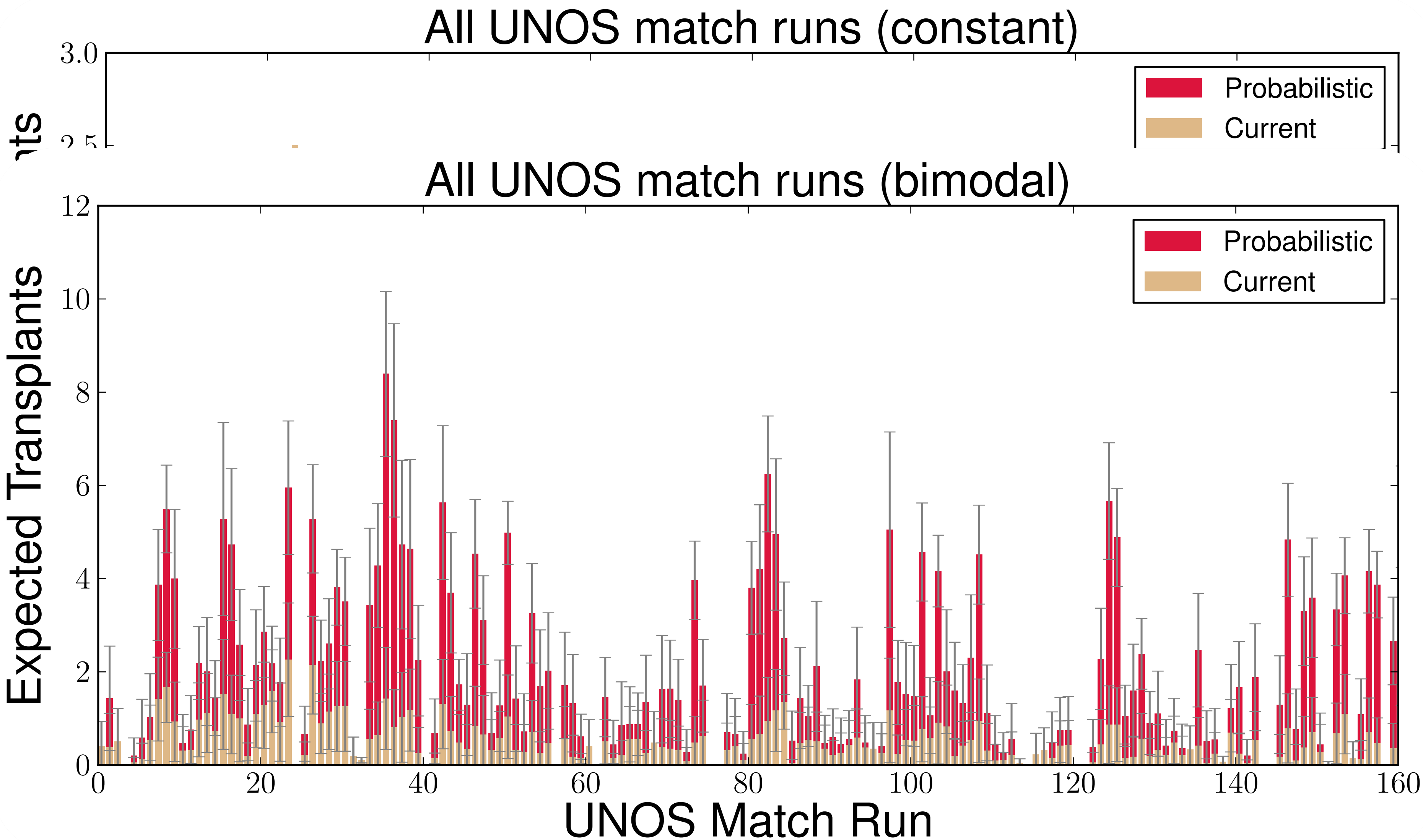


(a) A Y gadget. (b) The maximum cardinality matching M_Y. (c) The matching M'_Y.

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*)



Under discussion for implementation at UNOS

PRE-MATCH EDGE TESTING

Idea: perform a *small amount* of costly testing before a match run to test for (non)existence of edges

- E.g., more extensive medical testing, donor interviews, surgeon interviews, …

Cast as a *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

GENERAL THEORETICAL RESULTS

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

THEOREM

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

Stochastic k-set packing:
$(2/k - \varepsilon)$ 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

THEOREM

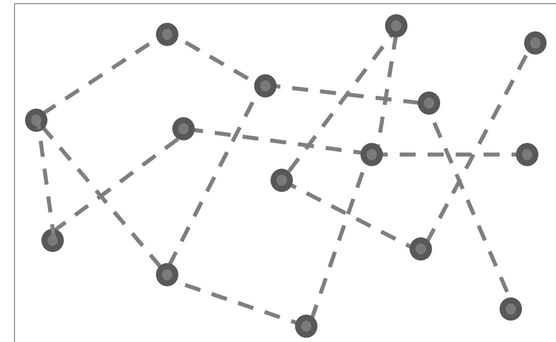
Stochastic matching:
$(0.5-\varepsilon)$ approximation with $O_\varepsilon(1)$ queries per vertex, in 1 round

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

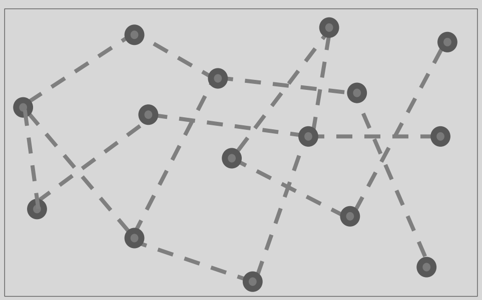
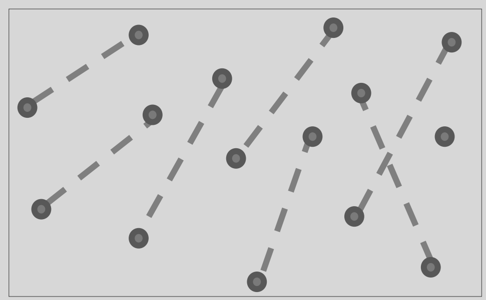
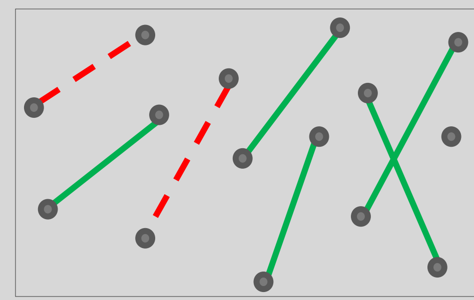
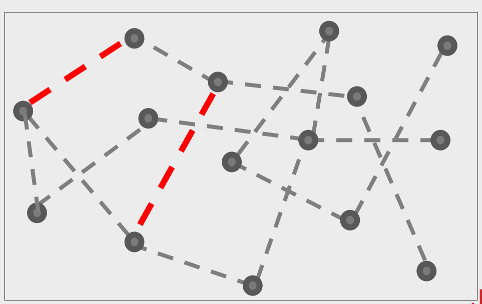
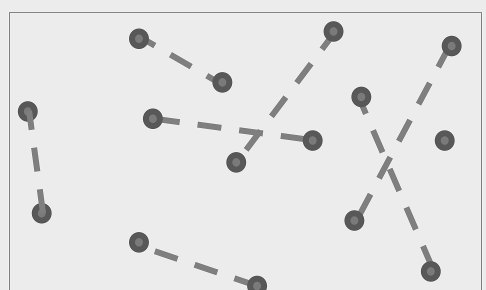
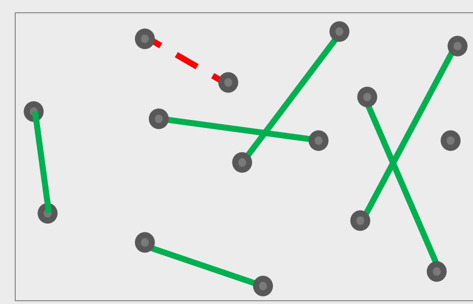
ADAPTIVE ALGORITHM

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:			

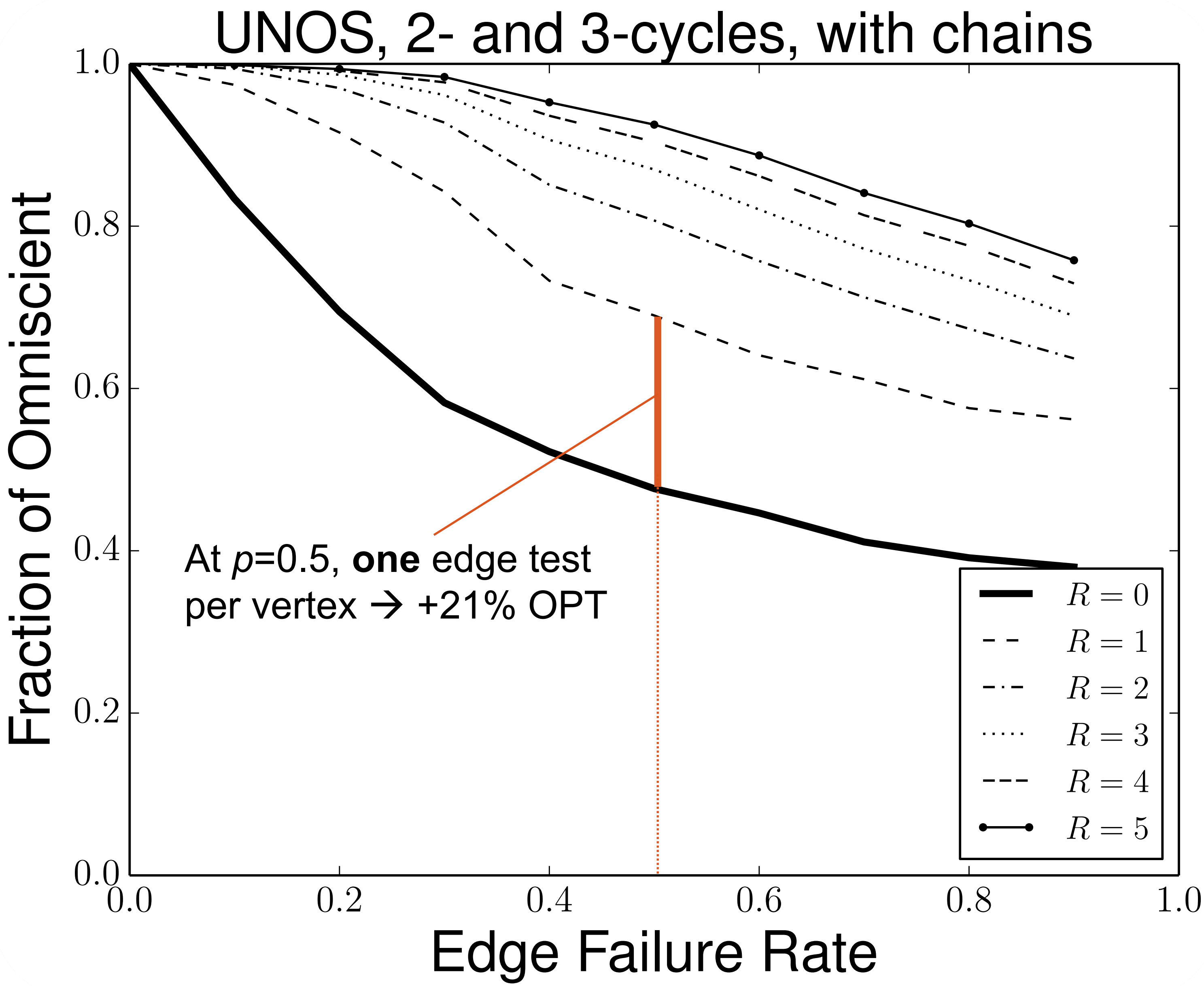
INTUITION FOR ADAPTIVE ALGORITHM

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, aka normal matching, simply augmenting paths

Augmenting structures might not exist, but can query in parallel in a single round

- Structures are constant size → exist with constant probability
- Structures are disjoint → queries are independent
- → Close a constant gap per round



Even 1 or 2 extra tests would result in a huge lift

In theory and practice, we're helping the global bottom line by considering post-match failure …

… But can this hurt some individuals?

BALANCING EQUITY AND EFFICIENCY

[AAMAS-14, AAAI-15, AAAI-18, Invited to AIJ, u.r. 2018]

With D. McElfresh, A. Procaccia and T. Sandholm

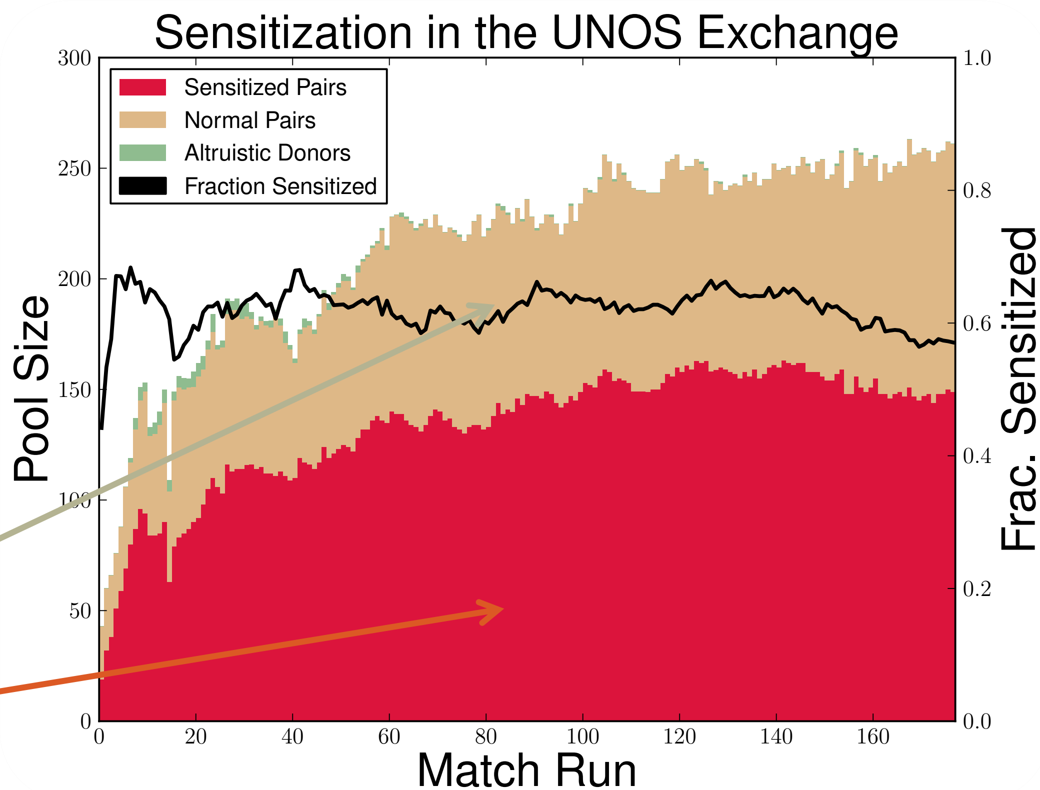
SENSITIZATION AT UNOS

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

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:

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

FROM THEORY TO PRACTICE

We show that the price of fairness is low in theory

$$POF(\mathcal{M}, u_{H \succ L}) \leq {}^{2}/_{33}$$

Fairness criterion: *extremely* strict.

Theoretical assumptions (standard):

- Big, dense graphs ("$n \rightarrow \infty$")
- 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 economic 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 →

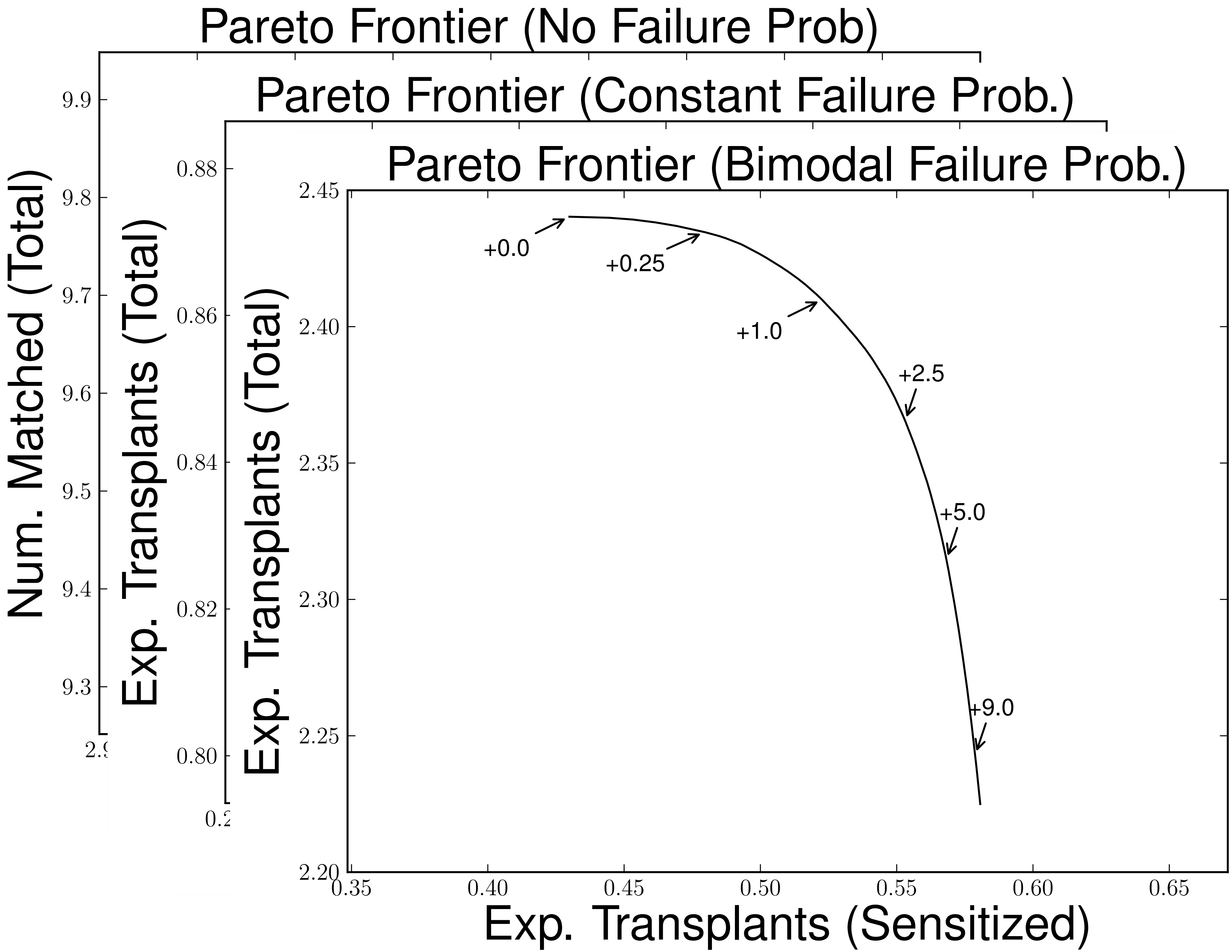
works with all present-day exchange solvers

Difficult to find a "good" β?

- Empirical exploration helps strike a balance

UNOS MATCH RUNS

WEIGHTED FAIRNESS, VARYING FAILURE RATES



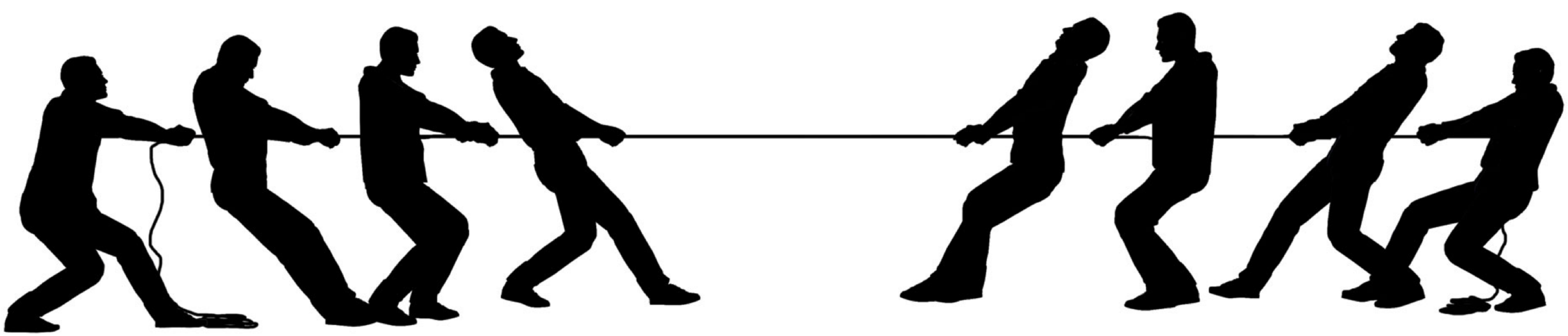
Pareto Frontier (No Failure Prob)
Pareto Frontier (Constant Failure Prob.)
Pareto Frontier (Bimodal Failure Prob.)
Num. Matched (Total)
Exp. Transplants (Total)
Exp. Transplants (Total)
Exp. Transplants (Sensitized)
+0.0
+0.25
+1.0
+2.5
+5.0
+9.0
9.9
9.8
9.7
9.6
9.5
9.4
9.3
0.88
0.86
0.84
0.82
0.80
2.45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

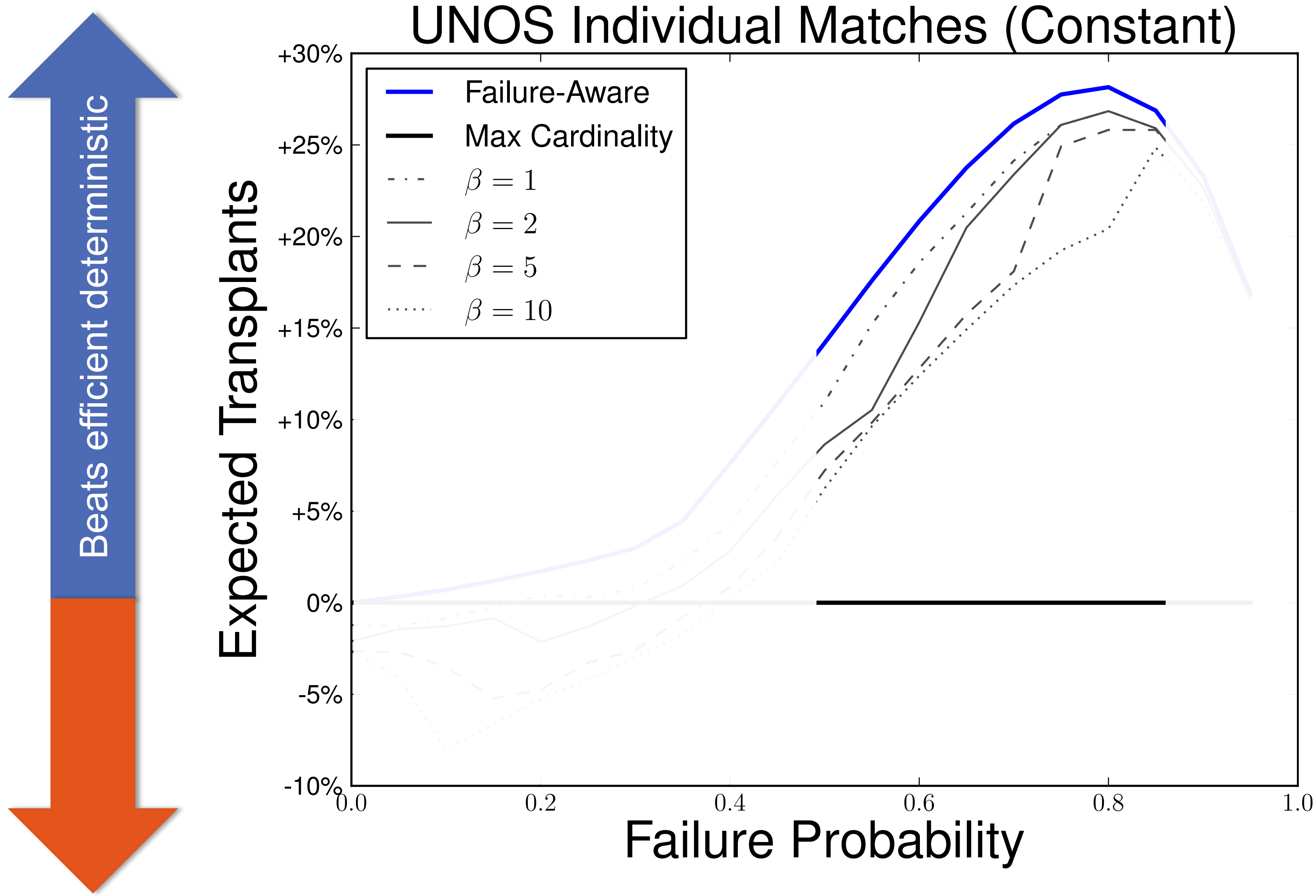
CONTRADICTORY GOALS

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)

Fairness vs. efficiency can be balanced in theory and in practice in a static model …

… But how should we match over time?

LEARNING TO MATCH IN A DYNAMIC ENVIRONMENT

[AAAI-12, AAAI-15, NIPS-15 MLHC, w.p. 2018]

With A. Procaccia and T. Sandholm

DYNAMIC KIDNEY EXCHANGE

Kidney exchange is a naturally dynamic event

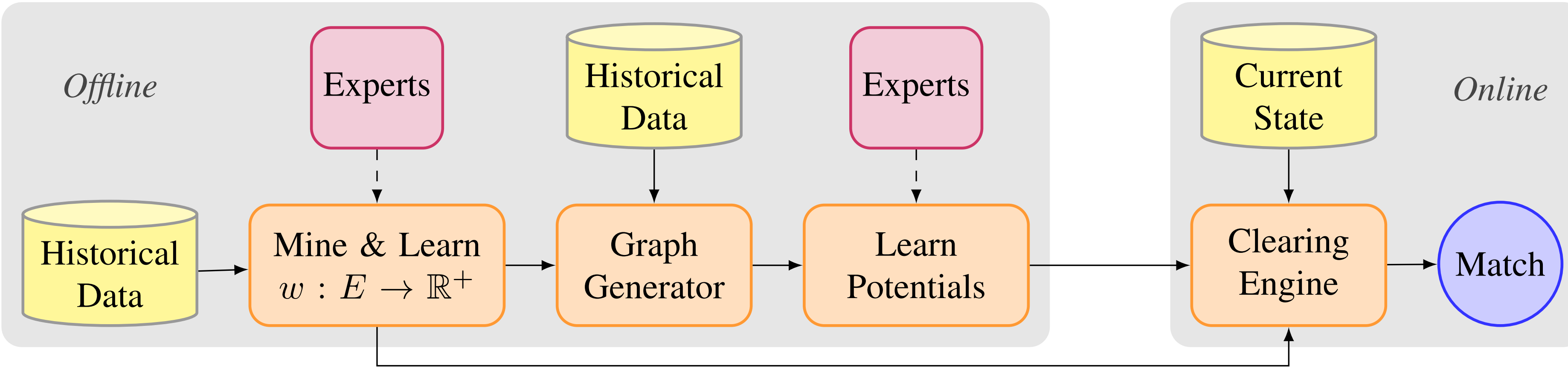
Can be described by the evolution of its graph

- 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		

How should we balance matching now versus waiting to match?

FUTUREMATCH: LEARNING TO MATCH IN DYNAMIC ENVIRONMENTS

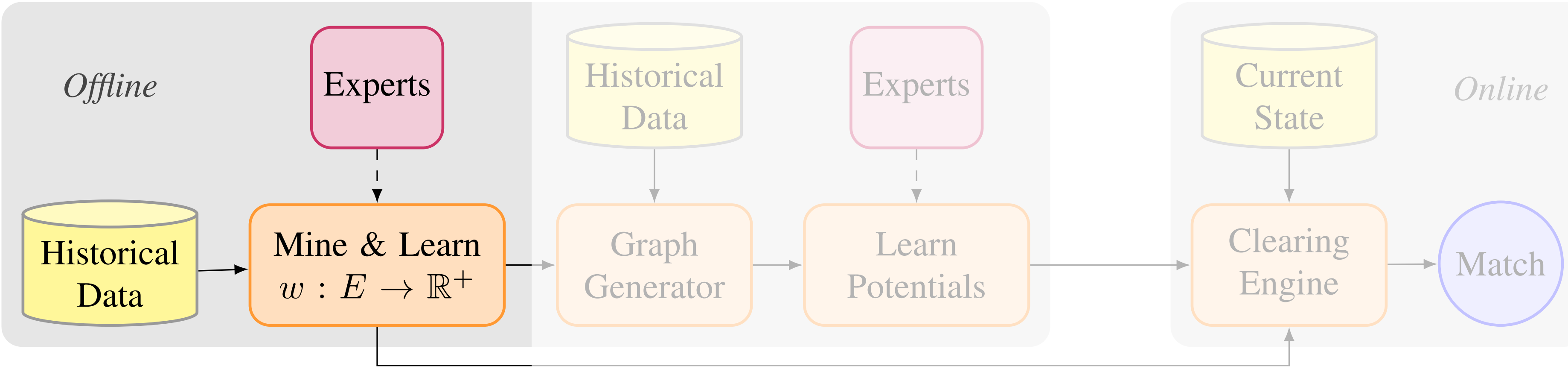


Offline (run once or periodically)

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

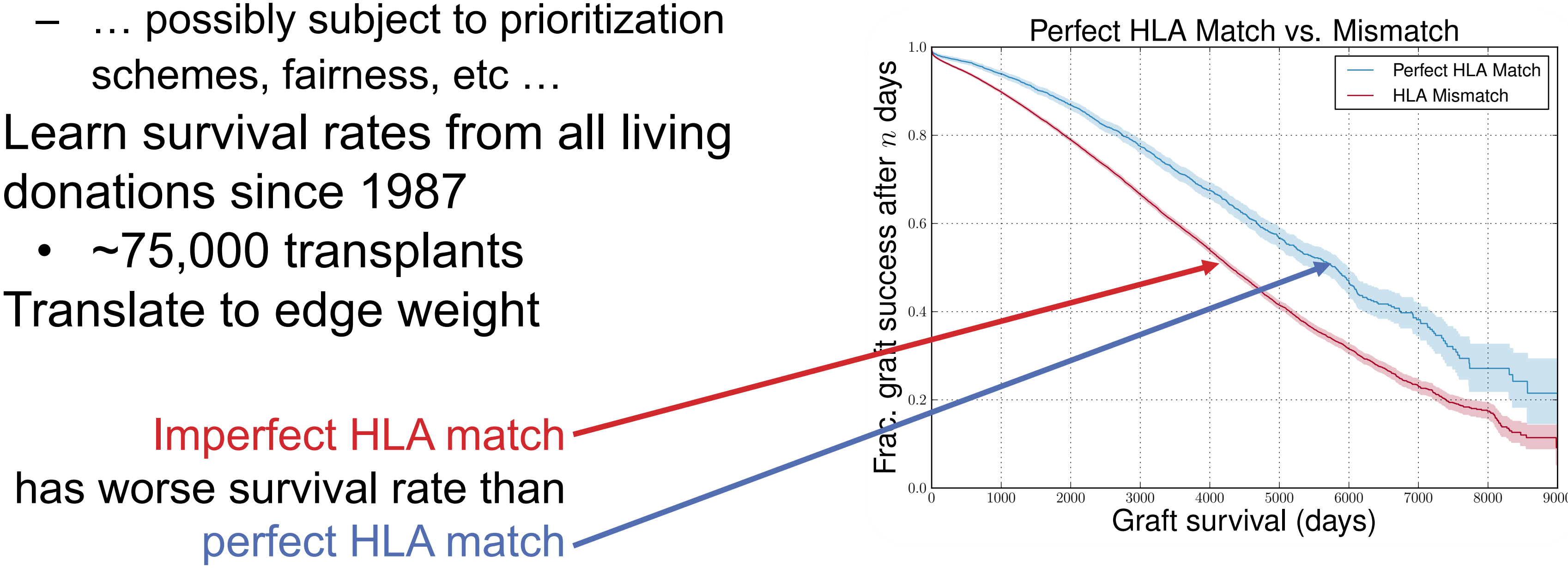
Online (run every match)

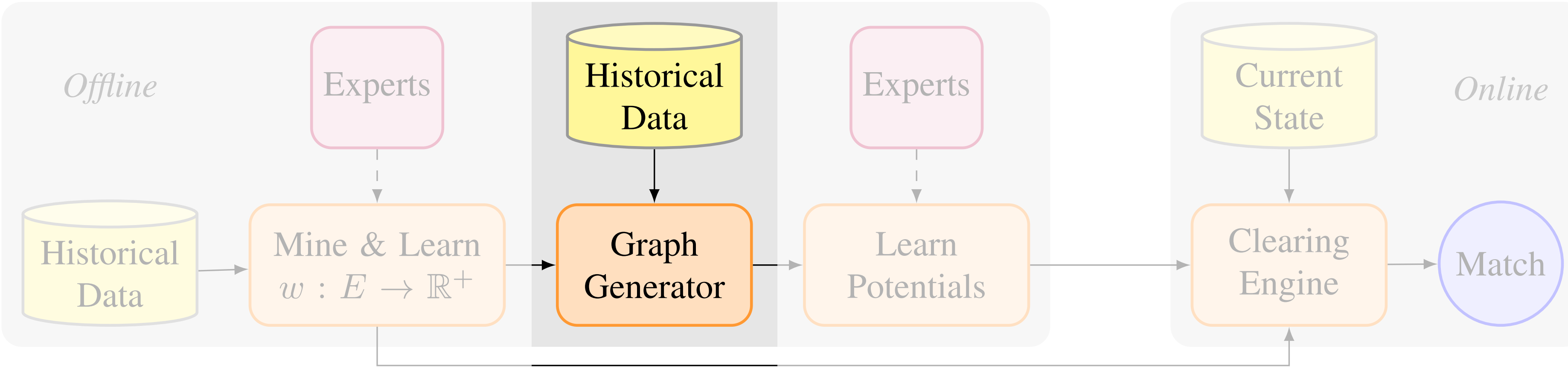
1. Combine *w* and potentials to form new edge weights on real input graphs
2. Solve maximum weighted matching and return match



Example objective (MaxLife)

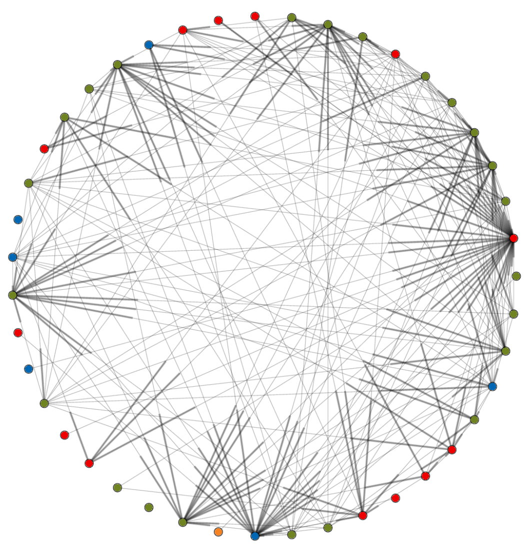
- Maximize aggregate length of time donor organs last in patients …
  - … possibly subject to prioritization schemes, fairness, etc …
- Learn survival rates from all living donations since 1987
  - ~75,000 transplants
- Translate to edge weight



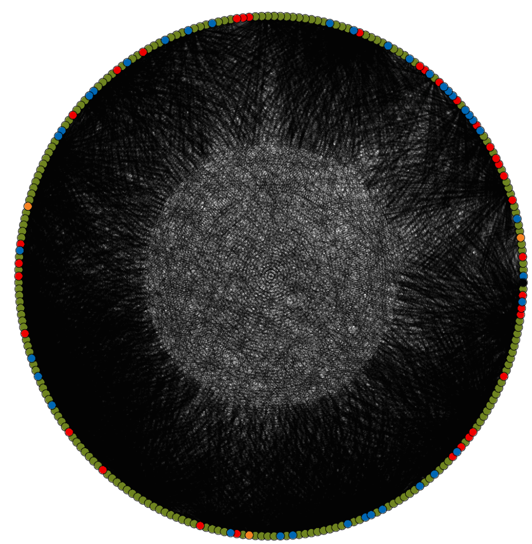


300+ match runs with real UNOS data

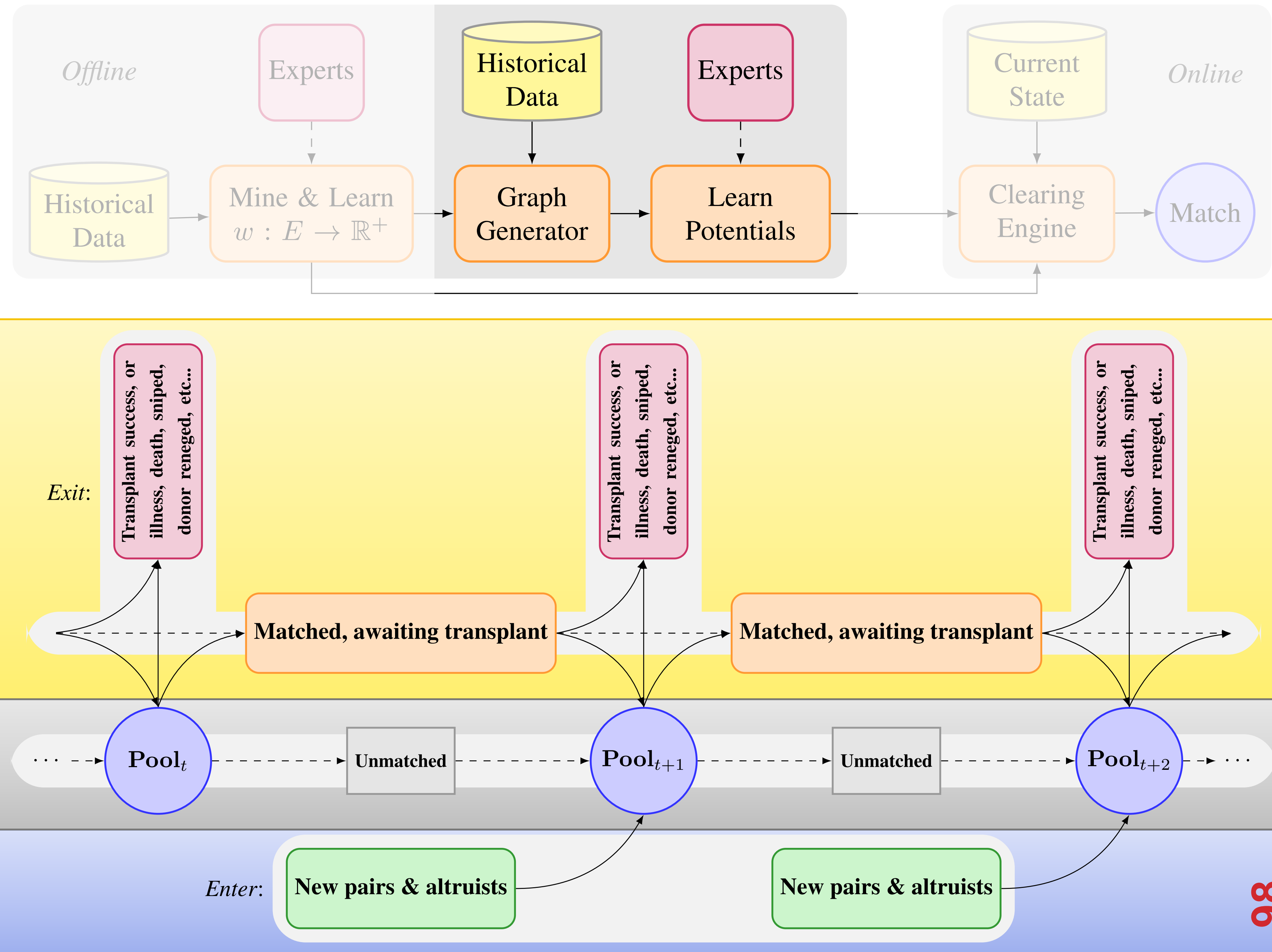
Important to use realistic distribution



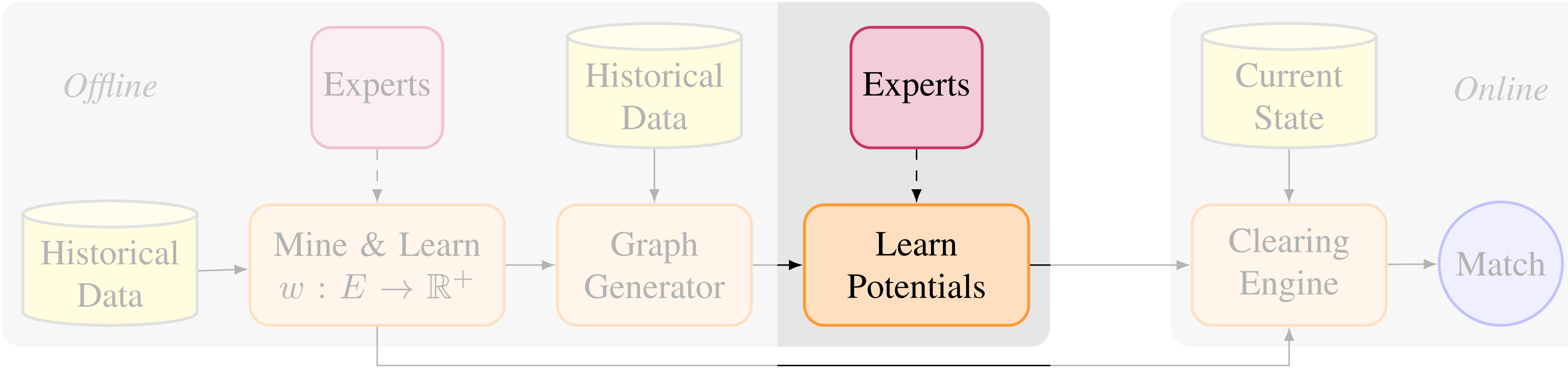
UNOS
(first match run)



UNOS
(recent snapshot)



Offline
Experts
Historical Data
Experts
Current State
Online
Historical Data
Mine & Learn
$w : E \to \mathbb{R}^+$
Graph Generator
Learn Potentials
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

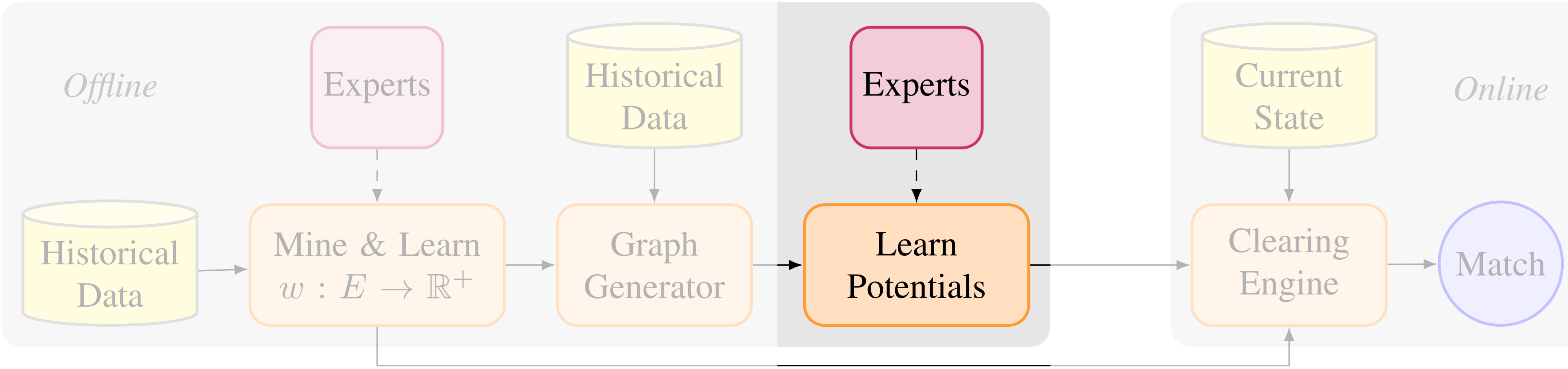


Full optimization problem is very difficult

- 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 is 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}

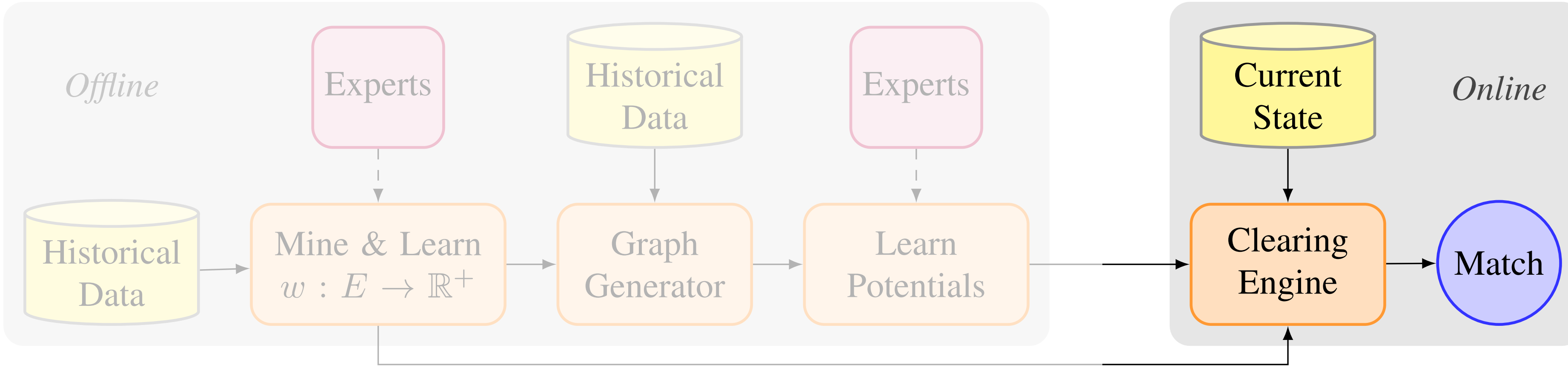
- 16 patient-donor types, 4 altruist types
- O-donors better than A-donors, so: $P_{\bullet\text{-}O} > P_{\bullet\text{-}A}$

Heavy one-time computation to learn potential of each type *θ* – we use SMAC

[Hutter Hoos Leyton-Brown 2011]

PITTSBURGH SUPERCOMPUTING CENTER

IBM



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 weight preprocess → no or low runtime hit!

PITTSBURGH SUPERCOMPUTING CENTER

EXPERIMENTAL RESULTS & IMPACT

Best Data-Intensive Application/Use

HPCwire Readers' Choice Awards 2014

Presented at Supercomputing

Tied with IBM Watson

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

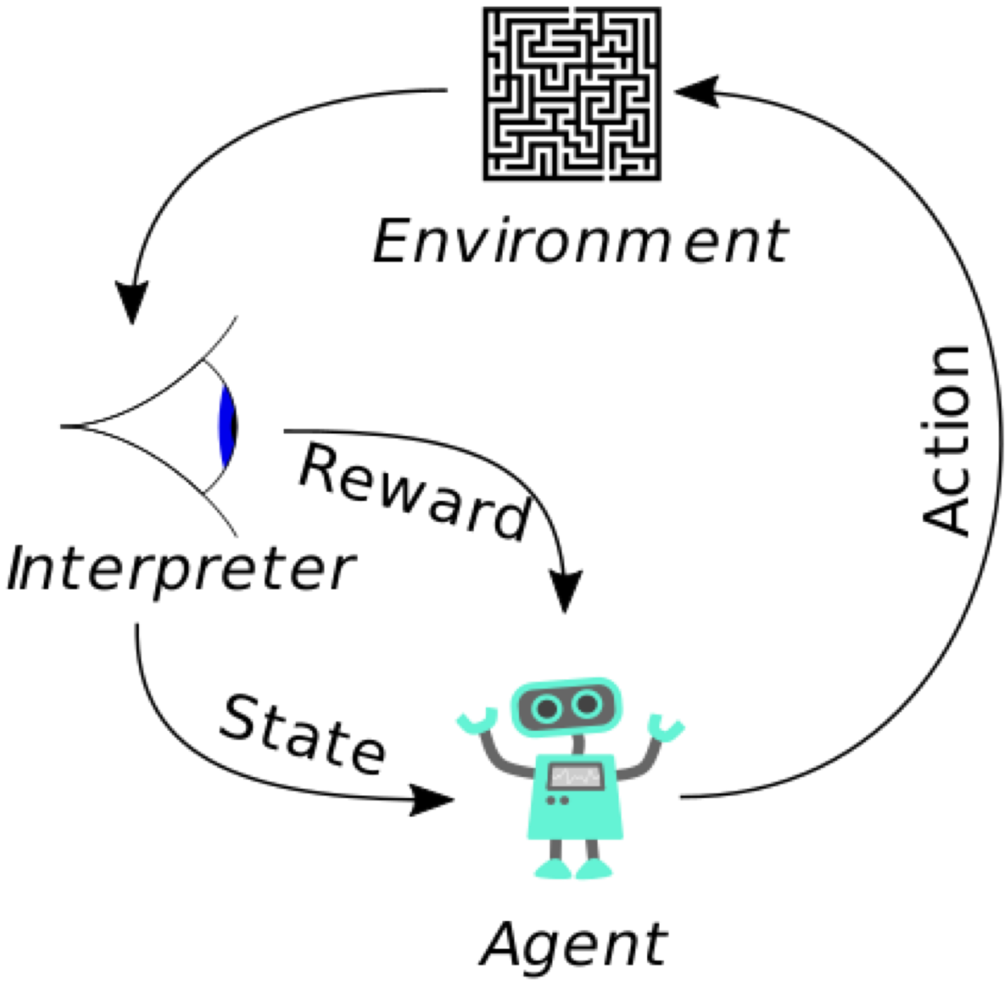
Sweet spot depends on distribution:

- Luckily, we can generate – and learn from – realistic families of graphs!

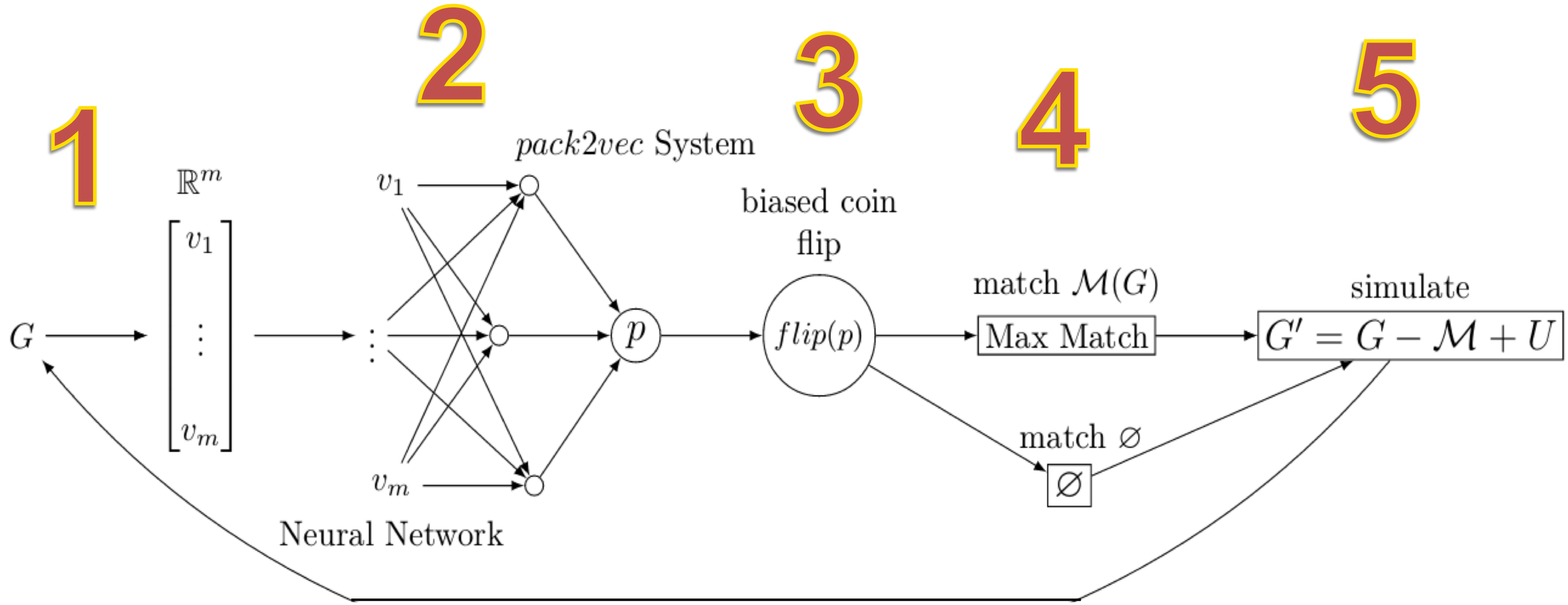
FutureMatch now used for policy recommendations at UNOS

THE GENERAL APPROACH …

REINFORCEMENT LEARNING

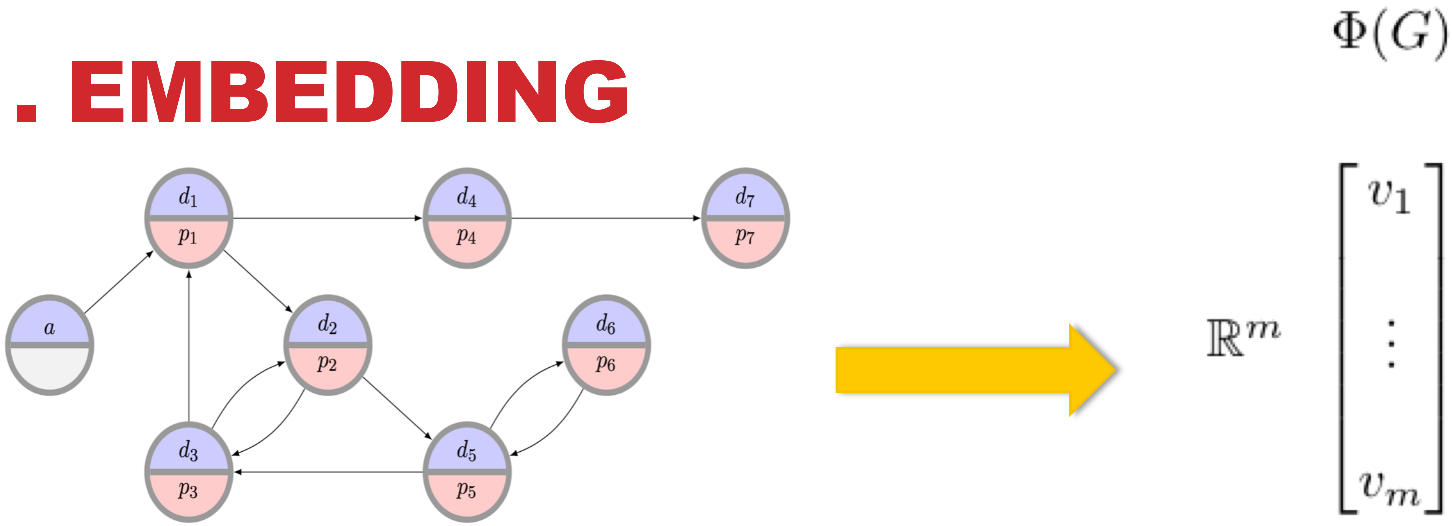


LEARNING TO MATCH IN DYNAMIC ENVIRONMENTS: OUR SYSTEM



1. **Embed** current compatibility graph into fixed-dimensional space
2. **Neural network** uses those vectors to learn appropriate policy
3. Flip a **biased coin**
4. If heads: find and **match** maximum cardinality matching
5. **Simulate** kidney exchange environment and grow the graph

1. EMBEDDING



Neural networks take a fixed-sized vector as input

- Our state space: graphs of any size
- Need: embed the graph as a vector and still maintain certain properties, such as node neighborhood structure. We use random walks to do so [Li, Campbell, Caceres 2017]

Use random walk on a carefully selected initial distribution to capture temporal changes in probability distribution

- Encode distance between pairs of probability distributions
- Empirically, this approach can distinguish between **Erdős–Rényi** and **Stochastic Block Model** graphs

SANITY CHECK FOR EMBEDDINGS: DISTANCE FUNCTIONS

Distance function to evaluate the degree of similarity/difference of two graphs.

- Goal: when two graphs are similar (largely different), their embedding vectors are close (far away) in terms of Euclidean distance;
- Optimal Distance Metric [Xu. et al. 2013]
  - Recognizes isomorphic graphs
  - NP-hard
- Symmetric Kullback-Leibler Divergence
  - Measures divergence rate between two probability distributions
  - Uses in-degree of vertices

2. EMBEDDING TO NEURAL NET

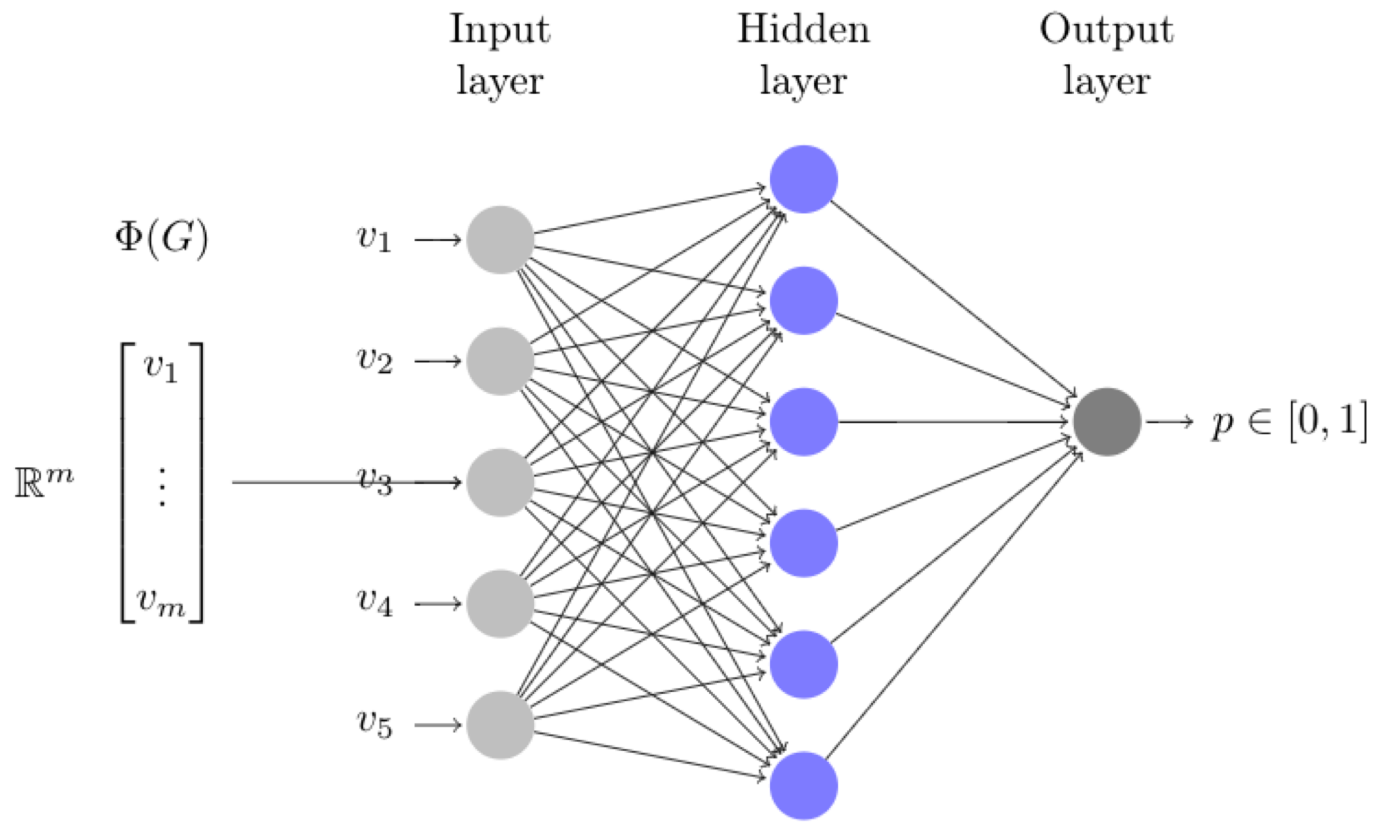
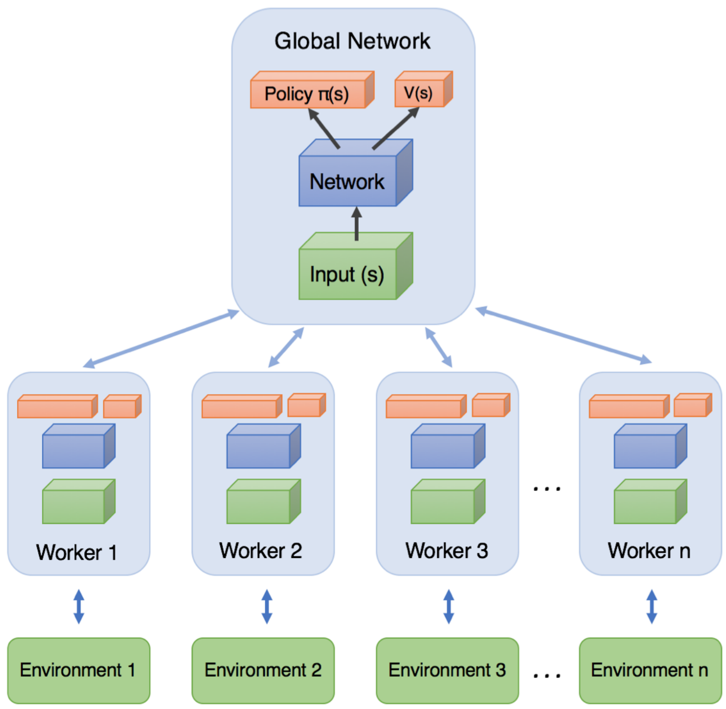
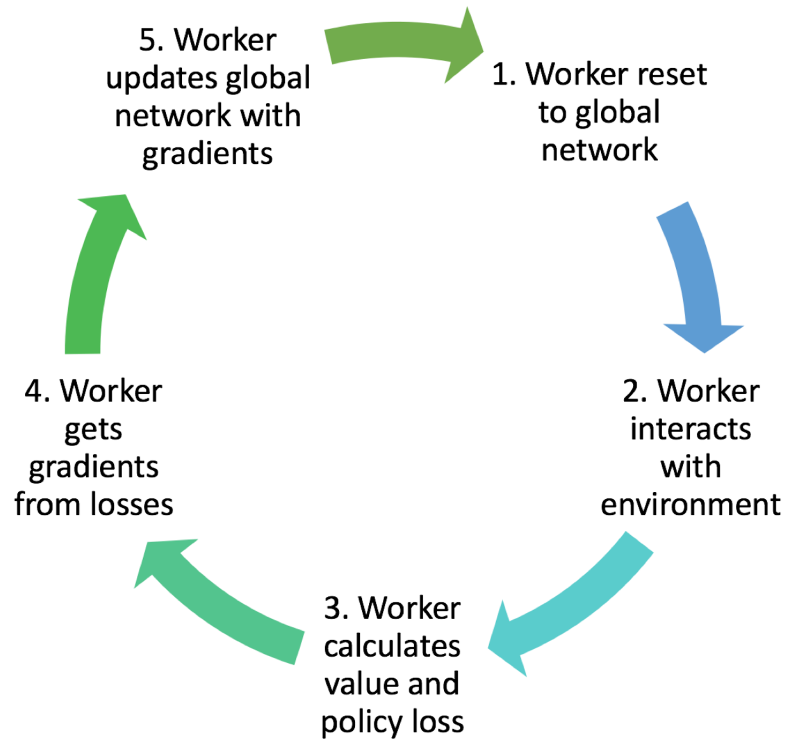


Figure 1: High-level neural network architecture.

Feed an embedded graph into, e.g., a neural network to output a learned probability for our biased coin flip

2. LEARNING ALGORITHM

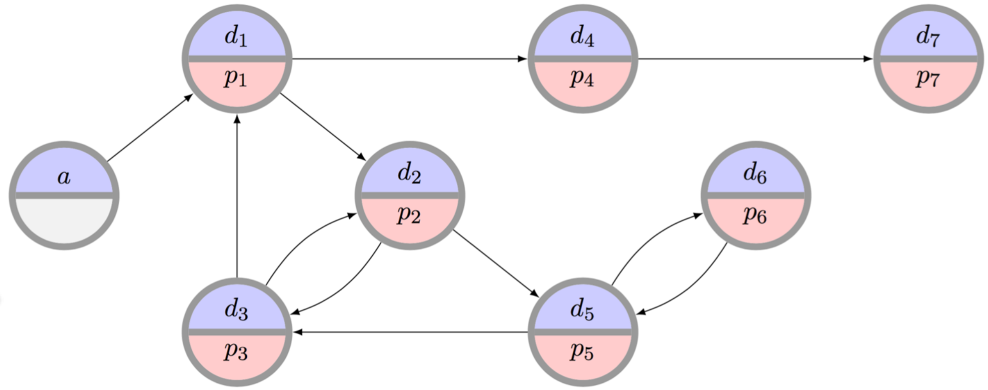




Using an adaptation of Asynchronous Advantage Actor-Critic (A3C) method [Mnih 2016]

3. BIASED COIN FLIP W/LEARNED PROBABILITY

1



MAX MATCH

0

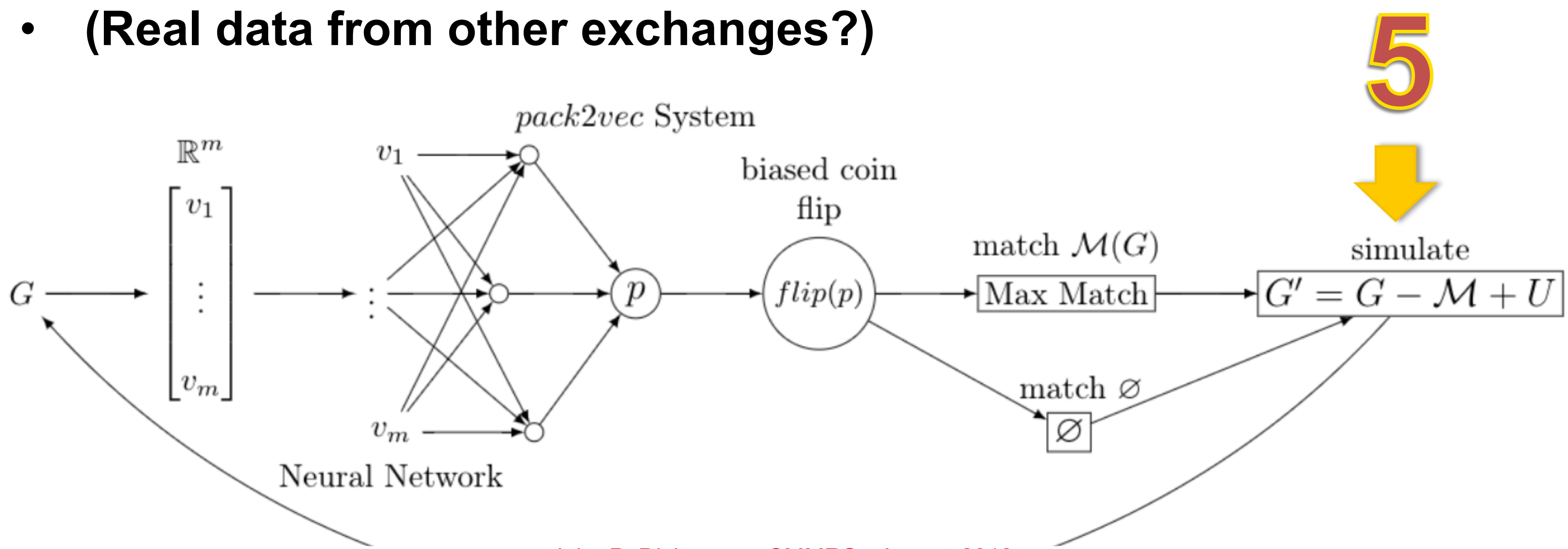
MATCH NOTHING (wait)

4. MAX MATCHING (THE CLEARING PROBLEM)—OR NOT

5. KIDNEY EXCHANGE SIMULATION – CHANGING THE INPUT GRAPH

To train the neural network, we must be able to simulate kidney exchange (graphs). We use several evolution models.

- Homogeneous Erdős–Rényi graphs [Akbarpour et al. 2017]
- Heterogeneous Erdős–Rényi graphs [Ashlagi et al. 2013]
- Real data from the UNOS exchange
- **(Real data from other exchanges?)**



EARLY RESULTS

We replicate results from prior theory papers:

- In some models, dynamic matching helps
- In some models, dynamic matching does not help

Still iterating on:

- Neural net structure
- Action space (binary coin flip vs. multiple match types)
- Learning method (A3C vs. DQN vs. more standard methods)

But …

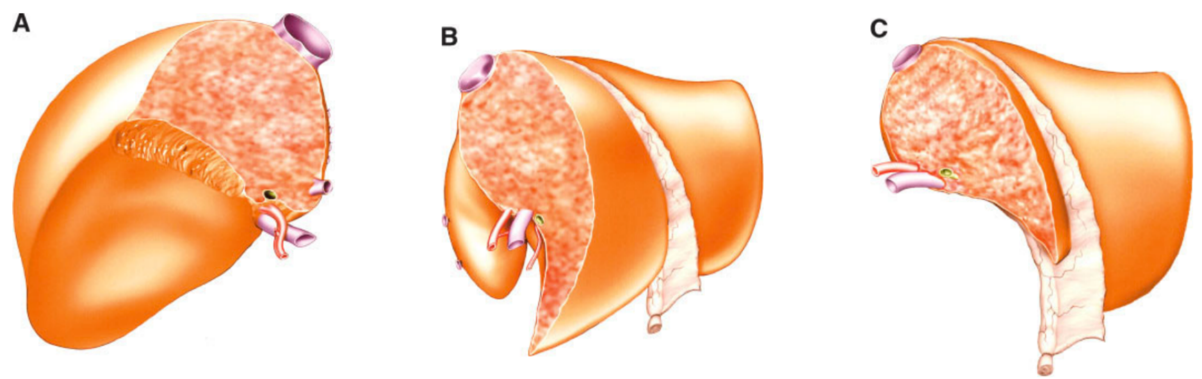
- **Seems promising.** Can learn matching policies beyond simply batching for T time periods; can realize gains over greedy.
- **Policies depend on graph structure.**

THE CUTTING EDGE

MOVING BEYOND KIDNEYS: LIVERS [Ergin, Sönmez, Ünver *w.p. 2015*]

Similar matching problem (mathematically)

Right Lobe Segments 5-8	Left + Caudate Lobes Segments 2-4	Left Lateral Segment Segments 2-3



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: MULTI-ORGAN EXCHANGE

[Dickerson Sandholm AAAI-14, JAIR-17]

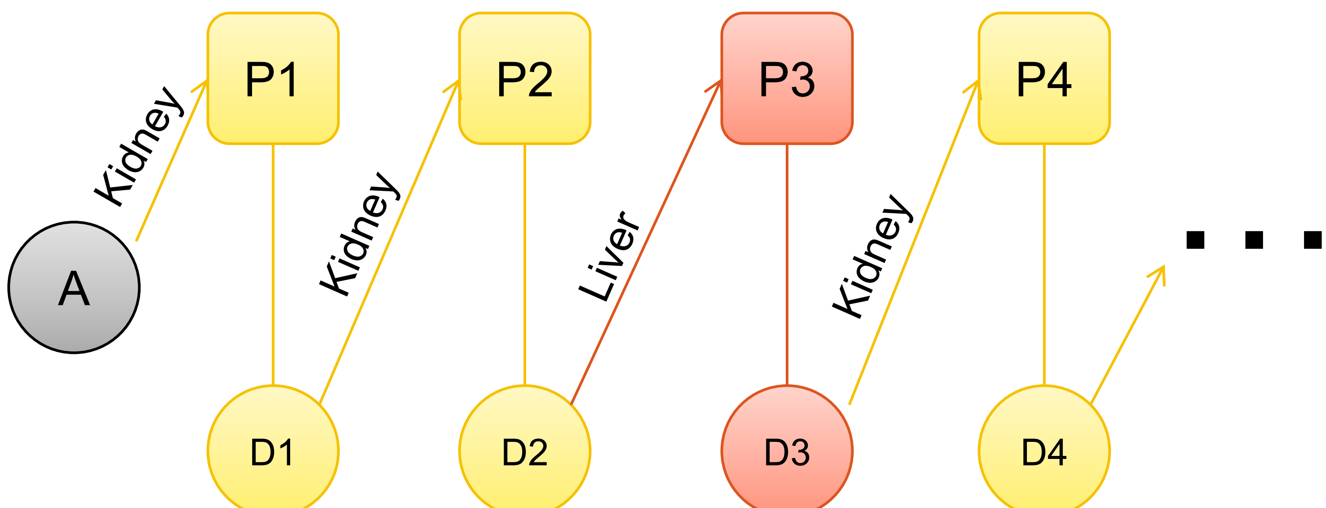
Chains are great! [Anderson et al. 2015, Ashlagi et al. 2014, Rees et al. 2009]

Kidney transplants are "easy" and popular:

- Many altruistic donors

Liver transplants: higher mortality, morbidity:

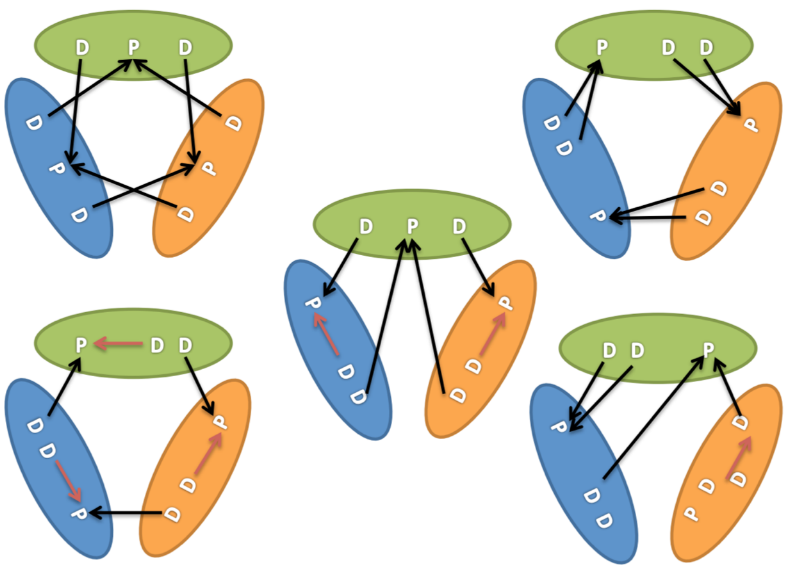
- (Essentially) no altruistic donors



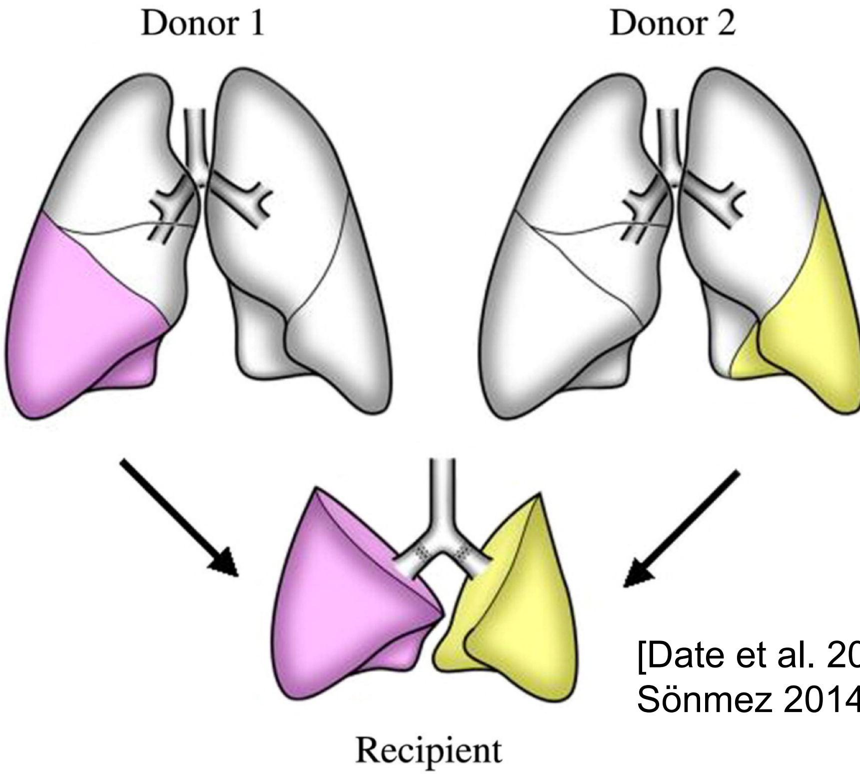
MOVING BEYOND KIDNEYS: LUNGS [Ergin, Sönmez, Ünver *w.p. 2014*]

Fundamentally different matching problem

- **Two** donors needed



3-way lung exchange configurations



[Date et al. 2005; Sönmez 2014]

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

KIDNEY CLUBS

A GENERAL MODEL OF ORGAN EXCHANGE

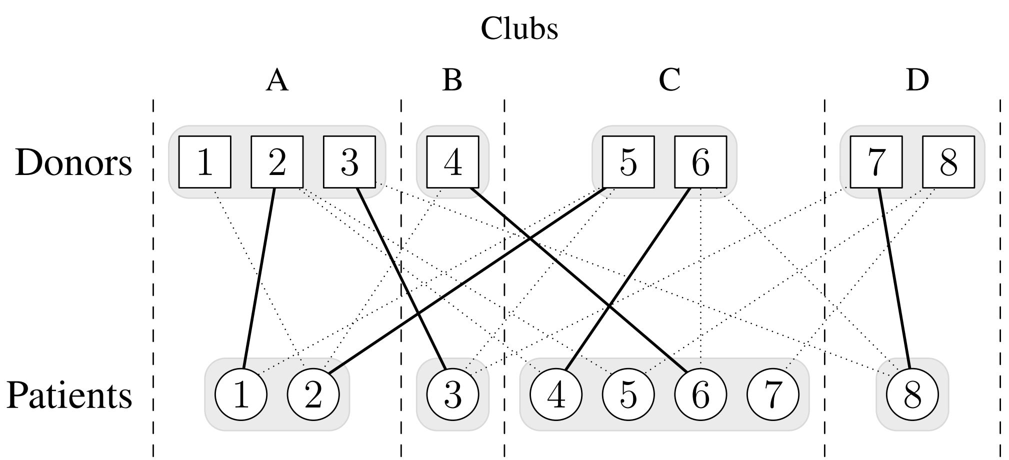
Joint with G. Farina, R. Formica, S. Kulkarni, R. Leishman, T. Sandholm, D. Stewart, C. Thiessen
Appears in IJCAI-17, AI-OR-SOC-GOOD@AAAI-17, AGT@IJCAI-17, ATC-17

KIDNEY CLUBS (I)

A club: set of healthy donors equally willing to donate one of their kidneys in exchange for an equal (or greater) number of kidneys received by a target set of patients

A club is made of:

- set of donors
- set of patients
- matching multiplier
- matching debt



KIDNEY CLUBS (II)

Idea: donors in club c are willing to donate outside of the club only if doing so results in a tangible benefit (i.e. kidneys donated) to patients in club c

Mechanism enforces:

Multiplier

Debt

$$n_d^{\text{ext}}(t) \leq \alpha_c n_p^{\text{ext}}(t) + \gamma_c$$

Kidneys donated externally

Kidneys received from other clubs

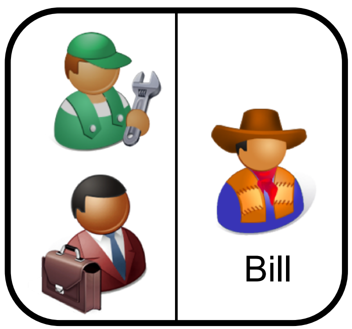
Debt accumulates with time. It is guaranteed that a club first receives kidneys and then might donate some in the future, always respecting the inequality above

“Chicken and egg” problem avoided, because some operations might happen simultaneously

EXPRESSIVITY (I)

The (uncapped) standard model is a special case:

(Generalized) Donor-Patient Pair

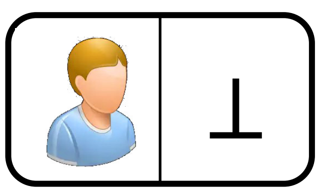


Forms a club on its own

The multiplier is 1, i.e. donors will donate only if Bill receives a kidney from elsewhere

The debt is initially 0

Non-directed donor



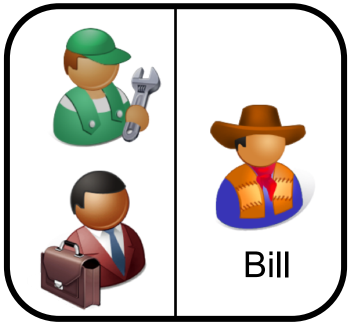
Forms a club on its own. There are no patients

The multiplier is irrelevant, the initial debt is 1

EXPRESSIVITY (II)

The kidney clubs model allows new possibilities

1.



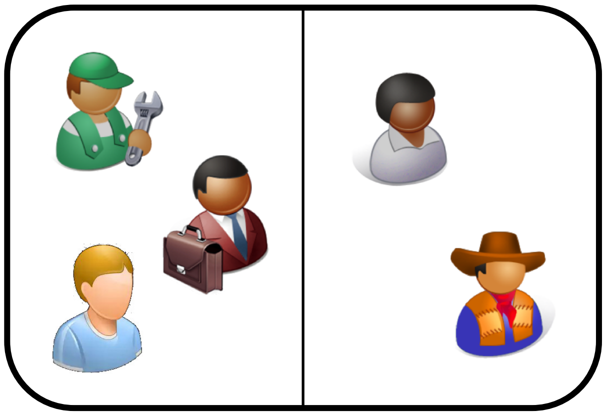
What if **both donors** are willing to donate?

They would agree to trade two (rather than one) of their own kidneys in exchange for the one Bill needs

EXPRESSIVITY (III)

The kidney clubs model allows new possibilities

2.



"**Organ alliances**" allow donations to the system in exchange for future donations from the system to the club

Ad-hoc forms of this have existed or do exist (e.g., LifeSharers, US military, Israeli military)

INCENTIVE ISSUES

All the standard incentive issues from the standard model, along with …

(1) Intra-club donations cannot be discouraged:

- The debt towards the system cannot increase when donations happen intra-club; otherwise, clubs might be incentivized to hide information

(2) Kidneys donated to a club via an inter-club donation have to be competitive with intra-club donations:

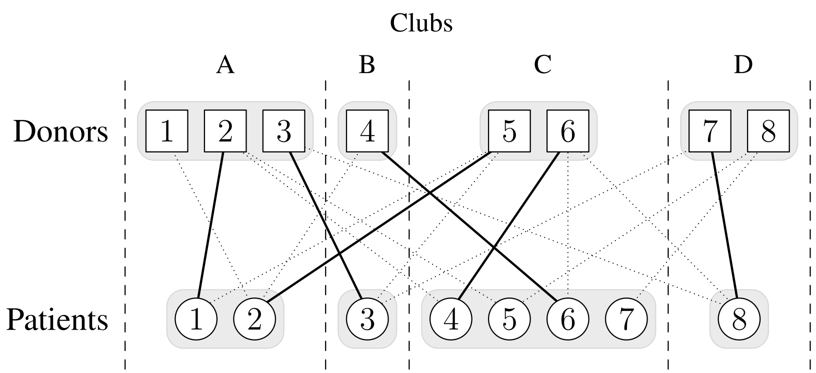
- Kidneys received via inter-club donations have to be at least as good as kidneys that can be received via intra-club donations

THE CLEARING PROBLEM: UNCAPPED FORMULATION

Assuming all operations happen simultaneously and there is no limit on the length of cycles and chains …

- The standard model's clearing problem is in PTIME
- The kidney clubs model's clearing problem is NP-hard

Easy MILP formulation:



$$\max \sum_{(d,p)\in E} w_{dp}\, x_{dp}$$

$$① \quad \sum_{\substack{p\in\mathcal{P}\\(d,p)\in E}} x_{dp} \le 1 \qquad \forall\, d\in\mathcal{D}$$

$$② \quad \sum_{\substack{d\in\mathcal{D}\\(d,p)\in E}} x_{dp} \le 1 \qquad \forall\, p\in\mathcal{P}$$

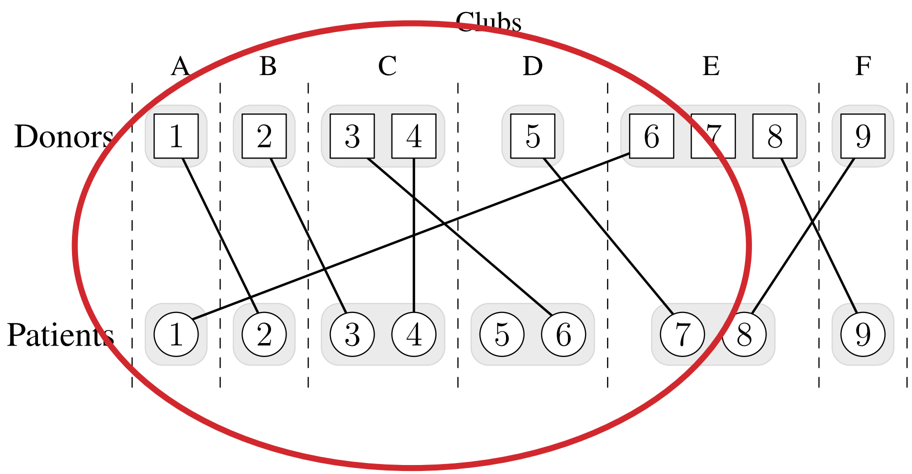
$$③ \quad \sum_{d\in D_c} \sum_{\substack{p\in\mathcal{P}\setminus P_c\\(d,p)\in E}} x_{dp} \le \gamma_c + \alpha_c \sum_{p\in P_c} \sum_{\substack{d\in\mathcal{D}\setminus D_c\\(d,p)\in E}} x_{dp} \qquad \forall\, c\in\mathcal{C}$$

$$④ \quad x_{dp}\in\{0,1\} \qquad \forall\,(d,p)\in E$$

FROM UNCAPPED TO CAPPED

The previous assumption is not realistic

- Too many operations might need to be carried out at the same time (typically, want limits of 3 or 4 simultaneous operations):



Patients: {1, 2, 3, 6, 7}
Donors: {1, 2, 3, 5, 6}

Need to be operated on at the same time (5 pairs!)

OPERATION FRAMES

Idea: introduce a temporal partial order among the edges

The partial order defines a DAG; operation frames are the vertices of this DAG

Lets us add constraints of the form “no more than K people get operated on at the same time in this frame”

$$\max \sum_{(d,p)\in E} \sum_{t\in T} h(t)\, w_{dp}\, x^t_{dp}$$

① $$\sum_{\substack{p\in\mathcal{P}\\(d,p)\in E}} \sum_{t\in T} x^t_{dp} \leq 1 \qquad \forall\, d \in \mathcal{D}$$

② $$\sum_{\substack{d\in\mathcal{D}\\(d,p)\in E}} \sum_{t\in T} x^t_{dp} \leq 1 \qquad \forall\, p \in \mathcal{P}$$

③ $$\sum_{\tau\rightsquigarrow t} \sum_{d\in D_c} \sum_{\substack{p\in\mathcal{P}\setminus P_c\\(d,p)\in E}} x^\tau_{dp} \leq \gamma_c + \alpha_c \sum_{\tau\rightsquigarrow t} \sum_{p\in P_c} \sum_{\substack{d\in\mathcal{D}\setminus D_c\\(d,p)\in E}} x^\tau_{dp} \qquad \forall\, c \in \mathcal{C},\ t\in T$$

④ $$\sum_{(d,p)\in E} x^t_{dp} \leq K \qquad \forall\, t \in T$$

⑤ $$x^t_{dp} \in \{0,1\} \qquad \forall\,(d,p)\in E,\ t\in T$$

DOES IT HELP?

EXPERIMENT

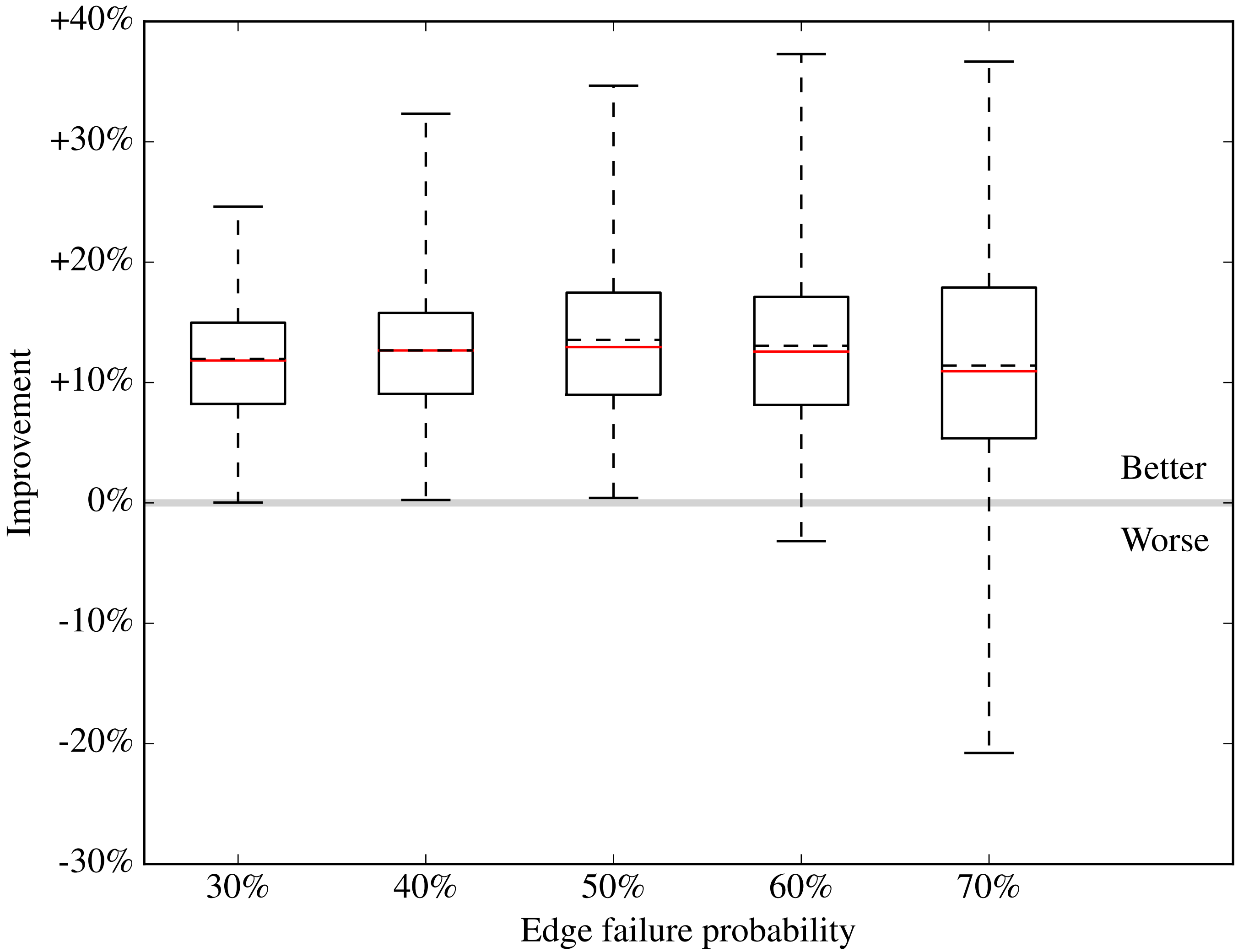
Compare the status quo of one donor matched per patient to matching (up to, if available) two donors per patient

Randomly sample any patient-donor pair / altruist who has ever participated in the UNOS exchange

- Some have more than one donor
- Multi-donor patients trigger (up to) one cycle and one chain

Match twice per week, 3-cycles, 4-chains

Use the same priority-points-based system as UNOS


+40%
+30%
+20%
+10%
0%
-10%
-20%
-30%
Improvement
30%
40%
50%
60%
70%
Edge failure probability
Better
Worse

OPEN QUESTIONS

Is this ethical?

- Many patients do arrive already with more than one donor …
- Can we use two donors per patient?
- Three donors per patient …?
- N donors per patient …?

Dynamics:

- Operation frames encode some notion of dynamic planning
- Can we take a prior over who will arrive/depart into account?

Incentive issues …

MANAGING STRATEGIC BEHAVIOR

MECHANISM DESIGN & MATCHING MARKETS

MANAGING INCENTIVES

Clearinghouse cares about global welfare:

- How many patients received kidneys (over time)?

Transplant centers care about their individual welfare:

- How many of my own patients received kidneys?

Patient-donor pairs care about their individual welfare:

- Did I receive a kidney?
- (Most work considers just clearinghouse and centers)

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 and altruists than reporting any other subset

→ No reason to strategize

EFFICIENCY

Does the mechanism result in the absolute best possible solution?

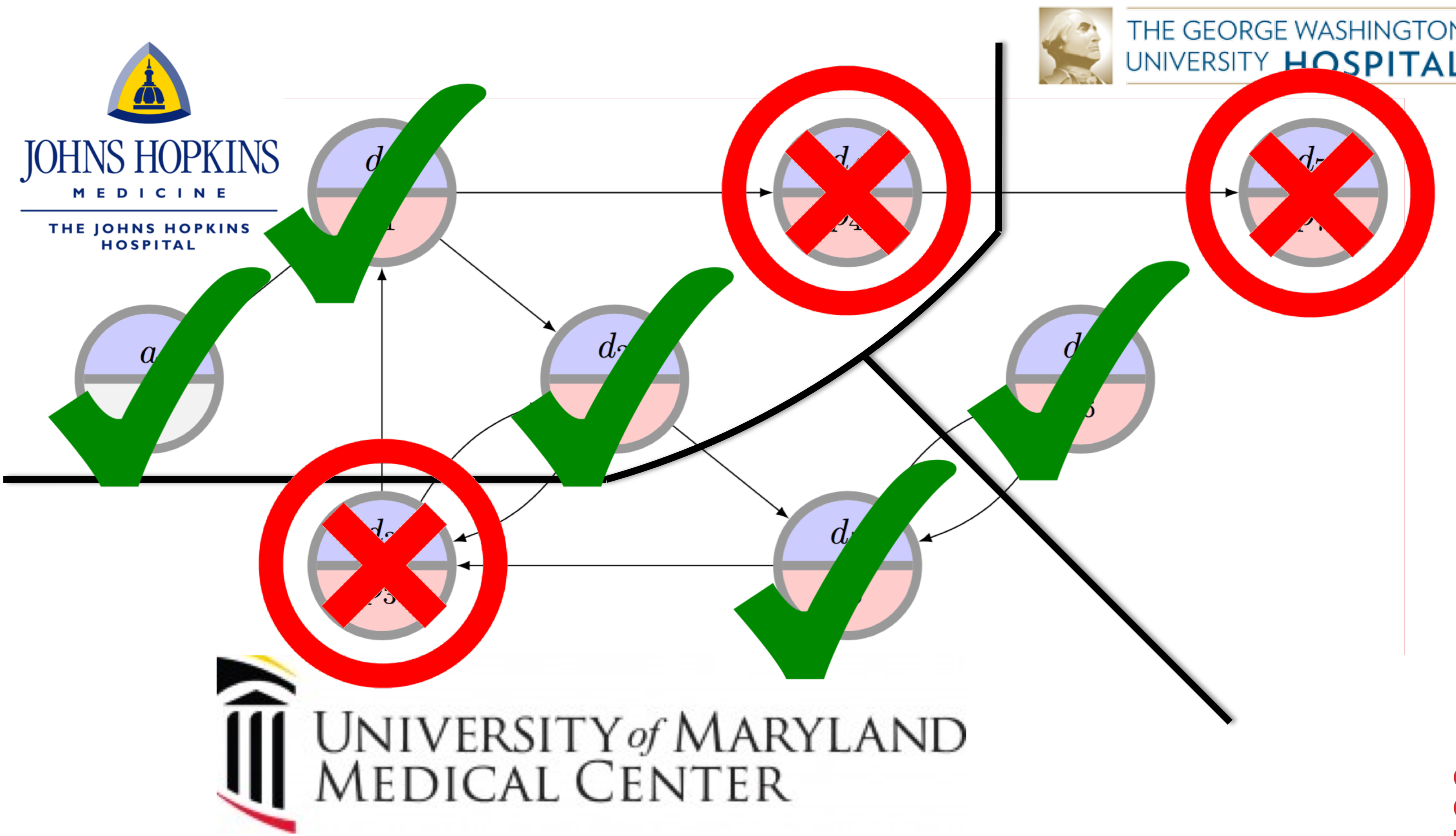
Efficiency:

- 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

PRIVATE VS GLOBAL MATCHING



0%

FIRST: ONLY CYCLES (NO CHAINS)

THE BASIC KIDNEY EXCHANGE GAME [Ashlagi & Roth 2014, and earlier]

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 to 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 → *k*-maximal, but not vice versa

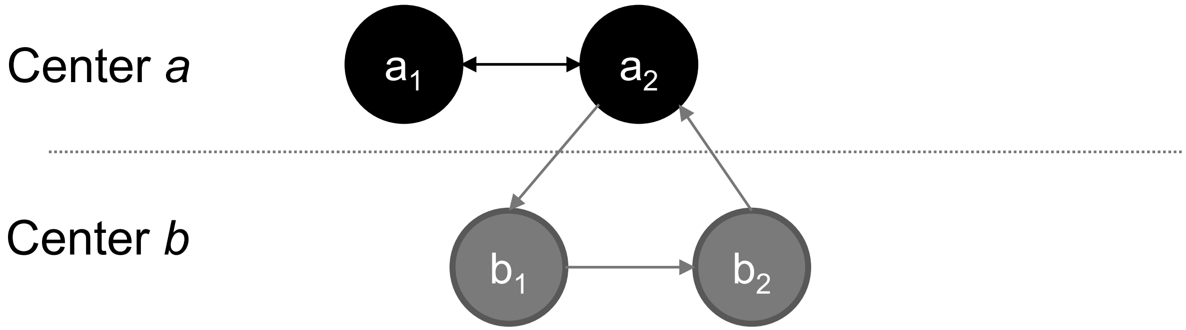
INDIVIDUALLY RATIONAL?

[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)**

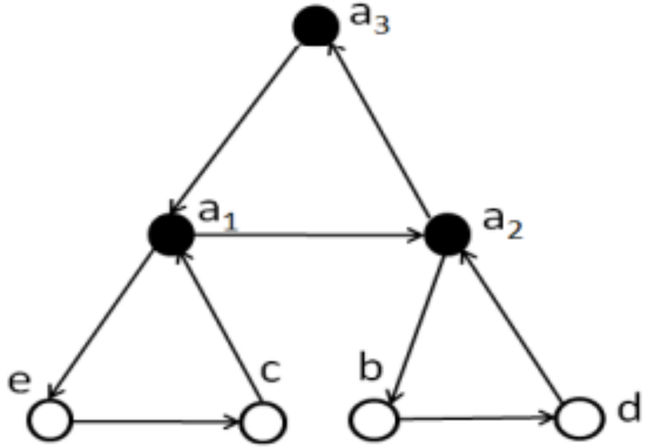


IT CAN GET MUCH WORSE

[Ashlagi & Roth 2014, and earlier]

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 [Ashlagi & Roth 2014, and earlier]

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 [Ashlagi & Roth 2014, and earlier]

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

Idea: Every 2-maximal allocation is also 2-efficient

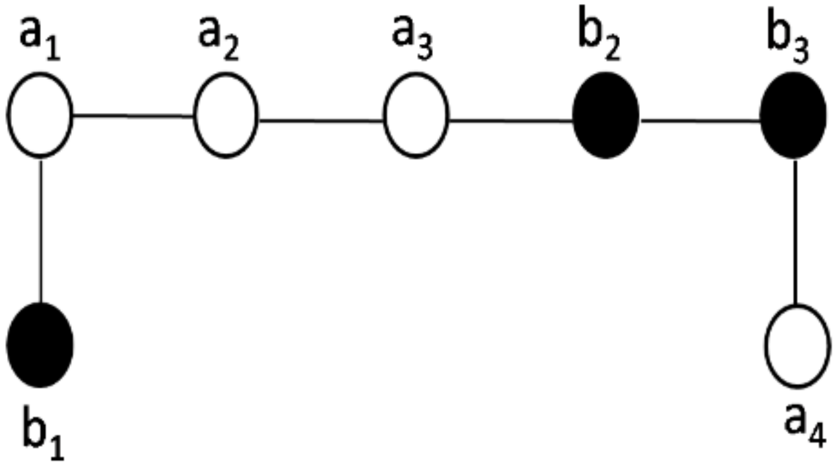
- This is a PTIME problem with, e.g., a standard $O(|V|^3)$ bipartite augmenting paths matching algorithm

By Restriction #1, 2-maximal IR always exists → this 2-efficient IR always exists

RESTRICTION #3 [Ashlagi et al. 2015]

Theorem: No IR mechanism is both maximal and strategyproof (even for k=2)

Suppose mechanism is IR and maximal . . .



HOPELESS ...?

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

LOTS OF OPEN PROBLEMS HERE

Dynamic mechanisms are more realistic, but …

- Vertices disappear after one time period
- All hospitals the same size
- No weights on edges
- No uncertainty on edges or vertices
- Upper bound on number of vertices per hospital
- Distribution might change over time
- ...

IS LIFE ALWAYS SO (NP-)HARD?

ONE SIMPLE ASSUMPTION COMPLEXITY THEORY HATES!

[Dickerson Kazachkov Procaccia Sandholm arxiv:1605.07728]

- **Observation: real graphs are constructed from a few thousand if statements**
  - If the patient and donor have compatible blood types …
  - ... and if they are compatible on 61 tissue type features ...
  - ... and if their insurances match, and ages match, and ...
  - ... then draw a directed edge; otherwise, don't

THEOREM

Given a constant number of if statements and a constant cycle cap, the clearing problem is in **polynomial time**

- **Hypothesis: real graphs can be represented by a small constant number of bits per vertex – we'll test later**

A NEW MODEL FOR KIDNEY EXCHANGE

[Dickerson et al. arxiv:1605.07728]

- **Graph $G = (V, E)$ with patient-donor pair v_i in V with**
  - Attribute vectors d_i and p_i such that the qth element of d_i (resp. p_i) takes on one of a fixed number of types
  - E.g., d_i^q or p_i^q takes a blood type in {O, A, B, AB}
  - Call Θ the set of all possible "types" of d and p
- **Then, given compatibility function $f : \Theta \times \Theta \rightarrow \{0,1\}$ that uniquely determines if an edge between d_i and p_j exists**
  - We can create any compatibility graph (for large enough vectors in D and P)
- **(Altruists are patient-donor pairs where the "patient" is compatible with all donors → chains are now cycles)**

CLEARING IS NOW IN POLYNOMIAL TIME

THEOREM

Given constant L and $|\Theta|$,
the clearing problem is in polynomial time

- **Let $f(\theta,\theta') = 1$ if there is a directed edge from a donor with type θ to a patient with type θ'**
- **For all $\theta = (<\theta_{1,p},\theta_{1,d}> \ldots, <\theta_{r,p},\theta_{r,d}>)$ in Θ^{2r} let**
  $f_C(\theta) = 1$ if $f(\theta_{t,d},\theta_{t+1,p}) = 1$ and $f(\theta_{r,d},\theta_{1,p}) = 1$
- **Given cycle cap L, define**
  $T(L) = \{ \theta \text{ in } \Theta^{2r} : r \leq L \text{ and } f_C(\theta) = 1 \}$

CLEARING IS NOW IN POLYNOMIAL TIME

- **T(*L*) is all vectors of types that create feasible cycles of length up to *L***

Algorithm 1 L-CYCLE-COVER

---

1. $\mathcal{C}^* \leftarrow \emptyset$
2. **for** every collection of numbers $\{m_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta} \in \mathcal{T}(L)}$ such that $\sum_{\boldsymbol{\theta} \in \mathcal{T}(L)} m_{\boldsymbol{\theta}} \leq n$
   - **if** there exists cycle cover $\mathcal{C}$ such that $\|\mathcal{C}\|_V > \|\mathcal{C}^*\|_V$ and for all $\boldsymbol{\theta} \in \mathcal{T}(L)$, $\mathcal{C}$ contains $m_{\boldsymbol{\theta}}$ cycles consisting of vertices of the types in $\boldsymbol{\theta}$ **then** $\mathcal{C}^* \leftarrow \mathcal{C}$
3. **return** $\mathcal{C}^*$

CLEARING IS NOW IN POLYNOMIAL TIME

- **Each set {m_θ} says we have $m_{\theta 1}$ cycles of type θ_1, $m_{\theta 2}$ cycles of θ_2, …, $m_{\theta|T(L)|}$ cycles of $\theta_{|T(L)|}$, constrained to at most n cycles total**

Algorithm 1 L-CYCLE-COVER

---

1. $\mathcal{C}^* \leftarrow \emptyset$
2. **for** every collection of numbers $\{m_{\boldsymbol\theta}\}_{\boldsymbol\theta \in \mathcal{T}(L)}$ such that $\sum_{\boldsymbol\theta \in \mathcal{T}(L)} m_{\boldsymbol\theta} \leq n$
   - **if** there exists cycle cover $\mathcal{C}$ such that $\|\mathcal{C}\|_V > \|\mathcal{C}^*\|_V$ and for all $\boldsymbol\theta \in \mathcal{T}(L)$, $\mathcal{C}$ contains $m_{\boldsymbol\theta}$ cycles consisting of vertices of the types in $\boldsymbol\theta$ **then** $\mathcal{C}^* \leftarrow \mathcal{C}$
3. **return** $\mathcal{C}^*$

CLEARING IS NOW IN POLYNOMIAL TIME

- **Check to see if this collection is a legal cycle cover – just check that each type θ isn't used too many times in m_θ**

Algorithm 1 L-CYCLE-COVER

1. $\mathcal{C}^* \leftarrow \emptyset$
2. **for** every collection of numbers $\{m_\theta\}_{\theta \in \mathcal{T}(L)}$ such that $\sum_{\theta \in \mathcal{T}(L)} m_\theta \leq n$
   - **if** there exists cycle cover $\mathcal{C}$ such that $\|\mathcal{C}\|_V > \|\mathcal{C}^*\|_V$ and for all $\boldsymbol{\theta} \in \mathcal{T}(L)$, $\mathcal{C}$ contains $m_{\boldsymbol{\theta}}$ cycles consisting of vertices of the types in $\boldsymbol{\theta}$ **then** $\mathcal{C}^* \leftarrow \mathcal{C}$
3. **return** $\mathcal{C}^*$

CLEARING IS NOW IN POLYNOMIAL TIME

- **Return the legal cycle cover such that the sum over θ of m_θ is maximized – aka the largest legal cycle cover**

Algorithm 1 L-CYCLE-COVER

---

1. $\mathcal{C}^* \leftarrow \emptyset$
2. **for** every collection of numbers $\{m_\theta\}_{\theta \in \mathcal{T}(L)}$ such that $\sum_{\theta \in \mathcal{T}(L)} m_\theta \leq n$
   - **if** there exists cycle cover $\mathcal{C}$ such that $\|\mathcal{C}\|_V > \|\mathcal{C}^*\|_V$ and for all $\theta \in \mathcal{T}(L)$, $\mathcal{C}$ contains m_θ cycles consisting of vertices of the types in θ **then** $\mathcal{C}^* \leftarrow \mathcal{C}$
3. **return** $\mathcal{C}^*$

FLIPPING ATTRIBUTES IS ALSO EASY

- **The human body tries to reject transplanted organs**
  - Before transplantation, can immunnosupress some "bad" traits of the patient to increase transplant opportunity
  - Takes a toll on the patient's health
- **Suppose we can pay some cost to change attributes**
- **For all θ, θ' in Θ, let**
  $c : \Theta \times \Theta \rightarrow R$ be cost of flipping $\theta \rightarrow \theta'$
- **Flip-and-Cover: maximize match size minus cost of flips**

THEOREM

Given constant *L* and $|\Theta|$,
the Flip-and-Cover problem is in polynomial time

A CONCRETE INSTANTIATION: THRESHOLDING

- **Associate with each patient and donor a *k*-bit vector**
  - Count "conflict bits" that overlap at same position
  - If more than threshold *t* conflict bits, don't draw an edge
- **Example: *k* = 2, blood containing antigens A and B**
  - $\Theta = 2^{\{\text{ has-A, has-B }\}} \times 2^{\{\text{ no-A, no-B }\}}$

    Donor blood type | Patient blood type

    Donor type A = [1, 0]
    Patient type AB = [0, 0] ✓

    Donor type A = [1, 0]
    Patient type O = [1, 1] ✗
- **Draw edge if $<d_i, p_j> \leq t$; do not draw edge otherwise**

A
S
I
D
E

Related to **intersection graphs**:
Each vertex has a set; draw edge between vertices iff sets intersect (by at least *p* elements)

UPPER BOUND: SOMETIMES YOU NEED LOTS OF BITS

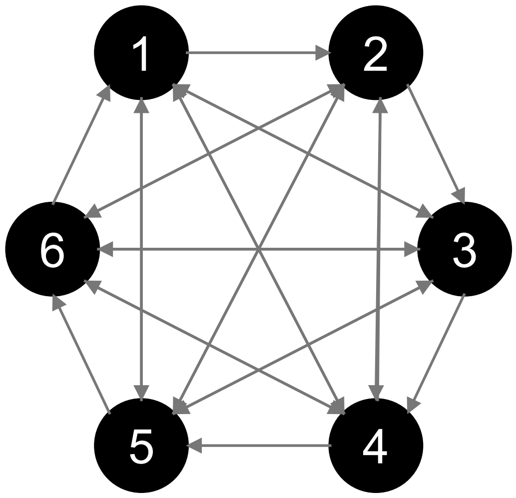
THEOREM

For any $n > 2$, there exists a graph on n vertices that is not $(k,0)$-representable for all $k < n$

For each vertex *i*, draw edge to each vertex except vertices *i*-1 and *i*

BWOC assume (*k*,0)-representable, *k* < *n*:

- **Consider vertex 1**
- **(1, *n*) not in *E*; (1, *i*) in *E* otherwise**
- **Then there is a conflict bit between vertex 1 and *n* that is not "turned on" anywhere else**
- **Do for *n* vertices → require *k* ≥ *n***



HARDNESS: HOW MANY BITS DO I NEED FOR THIS GRAPH?

Given: an input graph $G = (V, E)$
subset F of $C(V, 2)$

fixed positive k, nonnegative t

Does there exist:

k-length bit vectors d_i, p_i for all v_i in V

such that for (i,j) in F, also (i,j) in E iff $\langle d_i, p_j \rangle \leq t$

THEOREM

The (k,t)-representation problem is NP-complete
(proof via reduction from 3SAT)

COMPUTING (*K*, *T*)-REPRESENTATIONS: QCP

If an edge does not exist, make sure the overlap is greater than *t*

If an edge exists in the graph, assert the source donor vector and sink patient

- **Quadratically-constrained discrete feasibility program:**
  - Constraint matrix not positive semi-definite → non-convex
- **State-of-the-art nonlinear solvers (e.g., Bonmin) fail**

[Bonami et al. 2008]

COMPUTING (K, T)-REPRESENTATIONS: IP

$$
\begin{array}{lrr}
\min & \sum_{v_i \in V} \sum_{v_j \neq v_i \in V} \xi_{ij} & \\
\text{s.t.} & d_i^q \geq c_{ij}^q \wedge p_j^q \geq c_{ij}^q & \forall v_i \neq v_j \in V, q \in [k] \\
& d_i^q + p_j^q \leq 1 + c_{ij}^q & \forall v_i \neq v_j \in V, q \in [k] \\
& \sum_q c_{ij}^q \leq t + (k-t)\xi_{ij} & \forall (v_i, v_j) \in E \\
& \sum_q c_{ij}^q \geq (t+1)\xi_{ij} & \forall (v_i, v_j) \in E \\
& \sum_q c_{ij}^q \geq t + 1 - k\xi_{ij} & \forall (v_i, v_j) \notin E \\
& \sum_q c_{ij}^q \leq k - (k-t)\xi_{ij} & \forall (v_i, v_j) \notin E \\
& d_i^q, p_i^q \in \{0,1\} & \forall v_i \in V, q \in [k] \\
& c_{ij}^q, \xi_{ij} \in \{0,1\} & \forall v_i \neq v_j \in V, q \in [k]
\end{array}
$$

- **Integer program minimizes number of "conflict edges"**
  - CPLEX struggles to find non-trivial solutions
  - CPLEX cannot find feasible solution (when forcing all $\xi_{ij} = 0$)

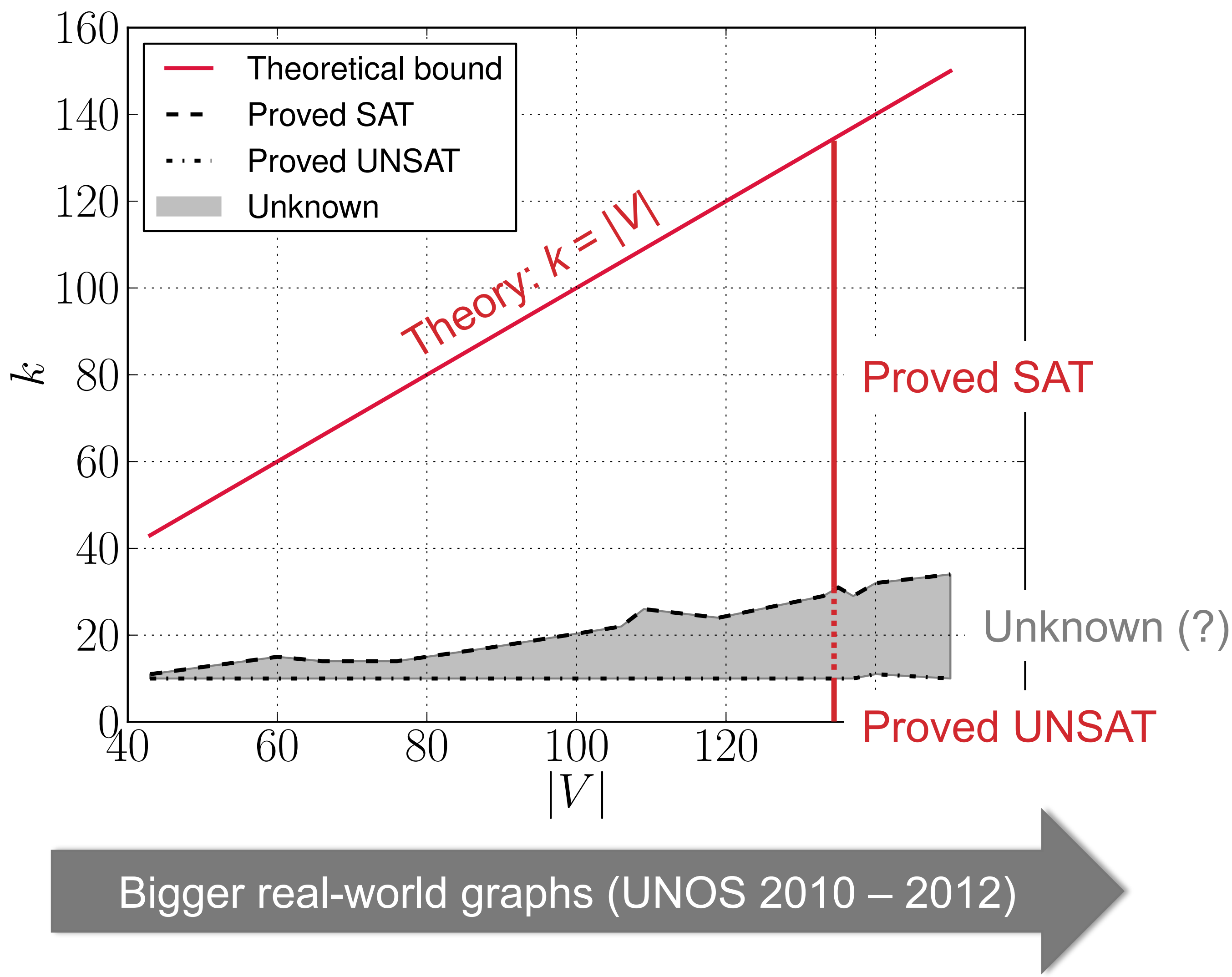
COMPUTING (K,0)-REPRESENTATIONS: SAT

Specific case of t = 0: if an edge does not exist, force any overlap

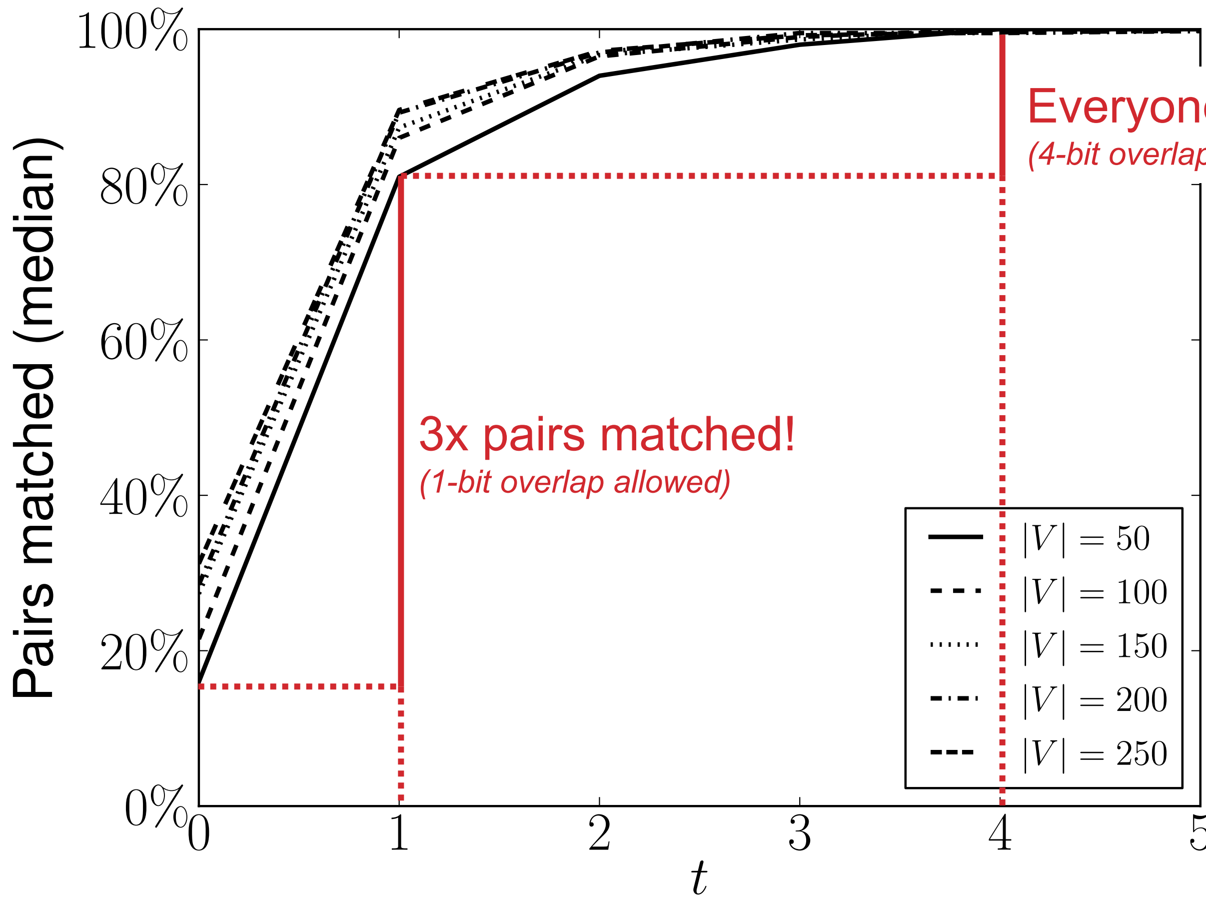
Specific case of t = 0: if an edge exists, allow no overlap

- **When t = 0, can use a compact SAT formulation**
  - Interesting because it closely mimics real life
- **We can solve small- and medium-sized graphs**
  - Use Lingeling, a good parallel SAT solver [Biere 2014]

CAN WE REPRESENT REAL GRAPHS WITH A SMALL NUMBER OF BITS?

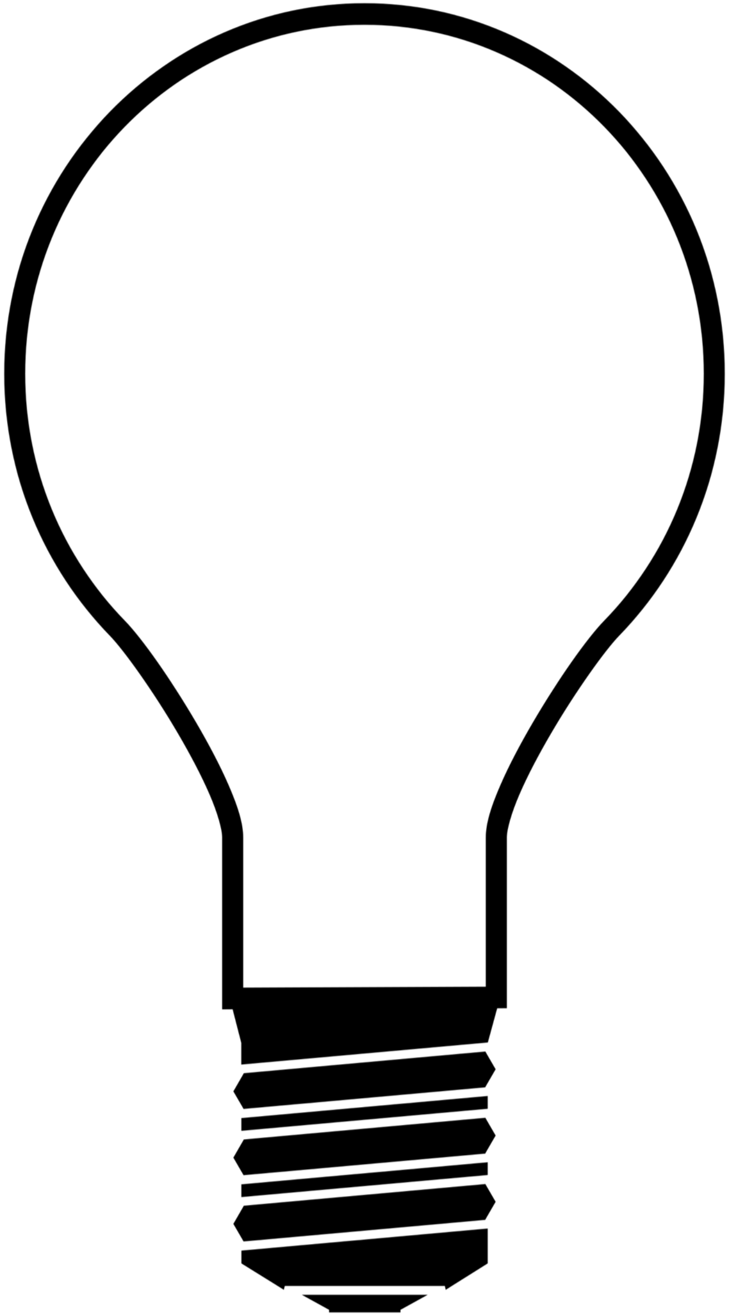


RELAXING THE THRESHOLD

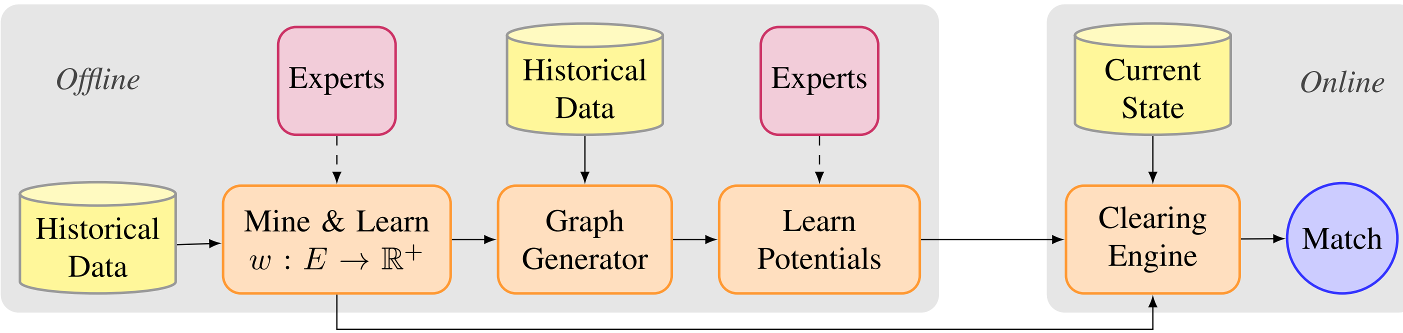


*all bits created equal, and not actually flipping bits – just relaxing global threshold

Loosen bit threshold *t* on real UNOS graphs



QUESTIONS?



Best Data-Intensive Application/Use
HPCwire Readers' Choice Awards 2014

More information:

http://jpdickerson.com

Code:

/JohnDickerson/KidneyExchange

Joint work with:

OPTIMIZED MARKETS

UNOS DONATE LIFE
UNITED NETWORK FOR ORGAN SHARING

Funding & support:

SIEBEL SCHOLARS

PITTSBURGH SUPERCOMPUTING CENTER

NDSEG Fellowship

Facebook Fellowship

BACKUP SLIDES

JOHN P DICKERSON