Maximizing Expectation on Vertex-disjoint Cycle Packing

João Pedro PEDROSO

INESC Porto and Universidade do Porto, Portugal [1]
jpp@fc.up.pt

International Conference on Computational Science and its Applications
Guimarães, July 2014

[1] This work is financed by the ERDF - European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the FCT - Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project *Kidney Exchange Programme*, PTDC/EGE-GES/110940/2009, and includes contributions from its research team.

Background: kidney exchange programs

- in many countries, recent legislation allows patients needing a kidney transplant to receive it from a living donor
- what to do when the transplant from that donor is not possible?
  - blood type
  - other incompatibilities
- patient-donor pair may enter a kidney exchange program (KEP)

U.PORTO
FACULDADE DE CIÊNCIAS UNIVERSIDADE DO PORTO

Kidney exchanges

- idea: allow two (or more) patients in incompatible pairs to exchange their donors
- each recipient receives a compatible kidney from the donor of another pair

Incompatible pairs $P_1 - D_1$ and $P_2 - D_2$ exchange donors

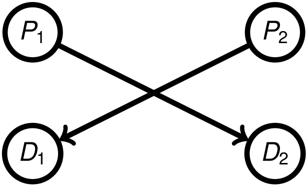
- P_1 receives a transplant from D_2 and vice versa

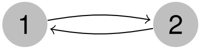
Graph representation:

- vertices are patient-donor pairs
- arcs link a patient to compatible donors

Kidney exchanges

- idea: allow two (or more) patients in incompatible pairs to exchange their donors
- each recipient receives a compatible kidney from the donor of another pair





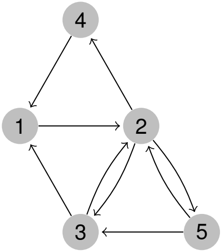
Incompatible pairs $P_1 - D_1$ and $P_2 - D_2$ exchange donors

- P_1 receives a transplant from D_2 and vice versa

Graph representation:

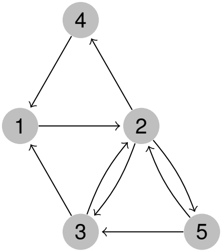
- vertices are patient-donor pairs
- arcs link a patient to compatible donors

Kidney exchanges: example



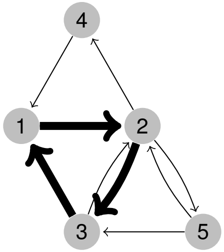
- instance with five pairs
- what is the maximum number of transplants?
- what if the allowed number of simultaneous transplants is limited?
- how to optimize if there is some probability of vertex/arc failure?

Kidney exchanges: example



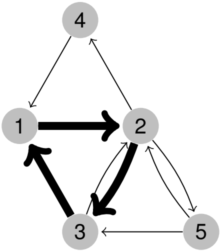
- feasible exchange: a set of vertex-disjoint cycles (*e.g.*, 1 – 2 – 3 – 1)
- size of an exchange: sum of the lengths of its cycles
- maximum exchange in this example: 4 (cycle 1 – 2 – 5 – 3 – 1)

Kidney exchanges: example



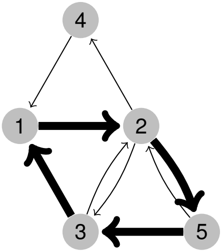
- feasible exchange: a set of vertex-disjoint cycles (*e.g.*, 1 – 2 – 3 – 1)
- size of an exchange: sum of the lengths of its cycles
- maximum exchange in this example: 4 (cycle 1 – 2 – 5 – 3 – 1)

Kidney exchanges: example



- feasible exchange: a set of vertex-disjoint cycles (*e.g.*, 1 – 2 – 3 – 1)
- size of an exchange: sum of the lengths of its cycles
- maximum exchange in this example: 4 (cycle 1 – 2 – 5 – 3 – 1)

Kidney exchanges: example



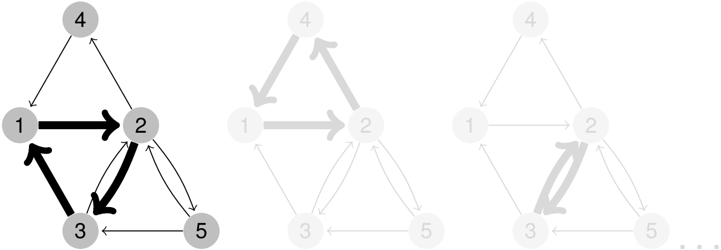
- feasible exchange: a set of vertex-disjoint cycles (*e.g.*, 1 – 2 – 3 – 1)
- size of an exchange: sum of the lengths of its cycles
- maximum exchange in this example: 4 (cycle 1 – 2 – 5 – 3 – 1)

Kidney exchanges: maximum cycle size

- In many situations the length of each cycle is limited:
  - limitations in the number of operation rooms
  - number of surgeons available
- If maximum cycle size is $K = 3$, several solutions are possible.

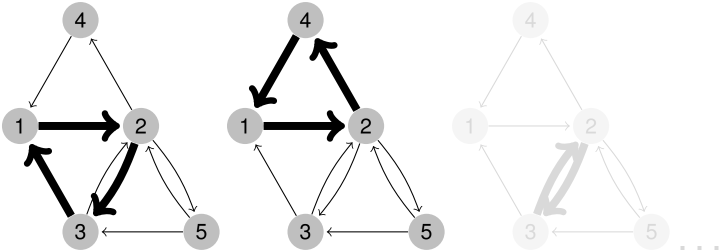
Kidney exchanges: maximum cycle size

- In many situations the length of each cycle is limited:
  - limitations in the number of operation rooms
  - number of surgeons available
- If maximum cycle size is $K = 3$, several solutions are possible.



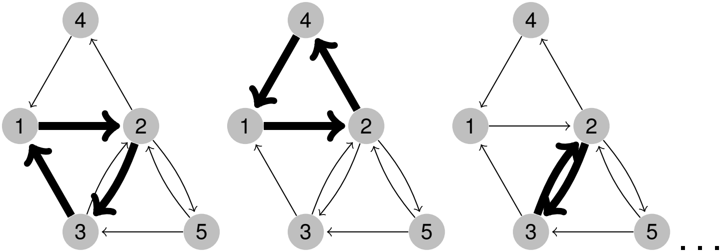
Kidney exchanges: maximum cycle size

- In many situations the length of each cycle is limited:
  - limitations in the number of operation rooms
  - number of surgeons available
- If maximum cycle size is $K = 3$, several solutions are possible.

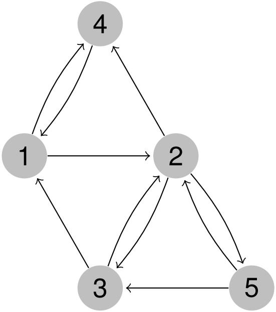


Kidney exchanges: maximum cycle size

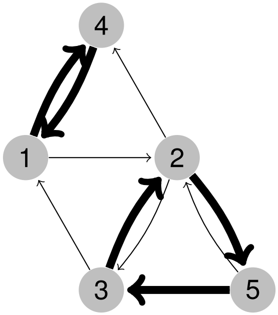
- In many situations the length of each cycle is limited:
  - limitations in the number of operation rooms
  - number of surgeons available
- If maximum cycle size is $K = 3$, several solutions are possible.



Another example



Another example



Maximum cycle size and NP-hardness

- In many situations the length of each cycle is limited
- If length is not limited $\longrightarrow$ ***assignment problem*** (polynomial algorithms are known, *e.g.*, hungarian algorithm).
- If length is limited to 2 $\longrightarrow$ ***matching problem*** (polynomial algorithms are known: Edmonds algorithm).
- If length is limited to 3, 4, ... $\longrightarrow$ ***problem is NP-hard*** (no polynomial algorithms are known).

U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Maximum cycle size and NP-hardness

- In many situations the length of each cycle is limited
- If length is not limited $\longrightarrow$ ***assignment problem*** (polynomial algorithms are known, *e.g.*, hungarian algorithm).
- If length is limited to 2 $\longrightarrow$ ***matching problem*** (polynomial algorithms are known: Edmonds algorithm).
- If length is limited to 3, 4, ... $\longrightarrow$ ***problem is NP-hard*** (no polynomial algorithms are known).

Maximum cycle size and NP-hardness

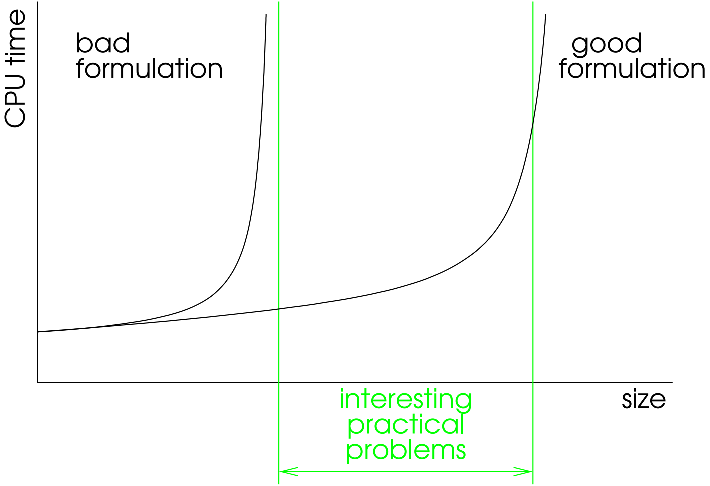
- In many situations the length of each cycle is limited
- If length is not limited $\longrightarrow$ ***assignment problem*** (polynomial algorithms are known, *e.g.*, hungarian algorithm).
- If length is limited to 2 $\longrightarrow$ ***matching problem*** (polynomial algorithms are known: Edmonds algorithm).
- If length is limited to 3, 4, ... $\longrightarrow$ ***problem is NP-hard*** (no polynomial algorithms are known).

U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Maximum cycle size and NP-hardness

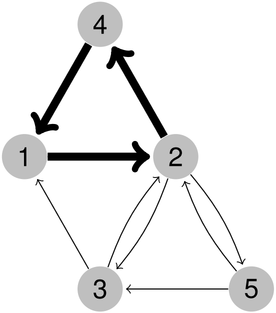
- In many situations the length of each cycle is limited
- If length is not limited $\longrightarrow$ ***assignment problem*** (polynomial algorithms are known, *e.g.*, hungarian algorithm).
- If length is limited to 2 $\longrightarrow$ ***matching problem*** (polynomial algorithms are known: Edmonds algorithm).
- If length is limited to 3, 4, ... $\longrightarrow$ ***problem is NP-hard*** (no polynomial algorithms are known).

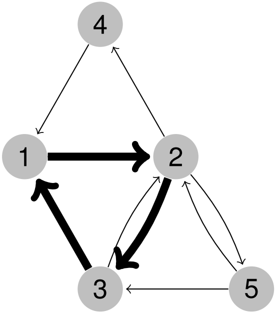
NP-hard problems



Mathematical programming formulations

- There are several possibilities for modeling the problem in mathematical programming
- One of the most successful is the *cycle formulation:*
  - enumerate all cycles in the graph with length at most K
  - for each cycle c, let variable x_c be 1 if c is chosen, 0 otherwise
  - every feasible solution corresponds to a set of vertex-disjoint cycles





Cycle formulation

$$\text{maximize} \quad \sum_{c} w_c x_c \tag{1a}$$

$$\text{subject to} \quad \sum_{c:i \in c} x_c \leq 1 \quad \forall i \tag{1b}$$

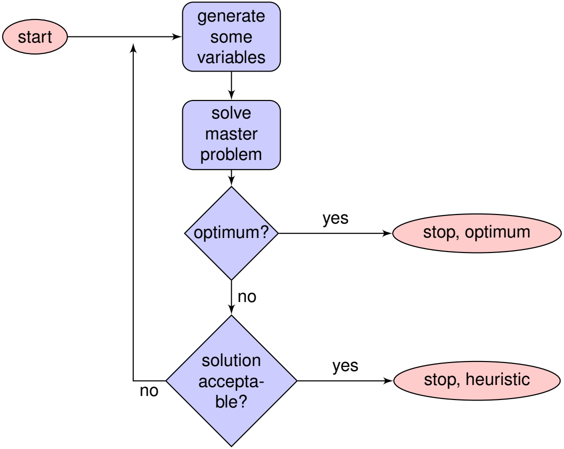
$$x_c \in \{0, 1\} \quad \forall c$$

- case of $0-1$ weights: $w_c = |c|$, (length of cycle c)
- objective: maximize the weight of the exchange
- constraints: every vertex is at most in one cycle (*i.e.*, donate/receive at most one kidney)
- difficulty: number of variables

Cycle formulation

- Exponential number of variables
- Not all are needed for solving the problem
- Use only those necessary $\longrightarrow$ column generation

Column generation



Previous results

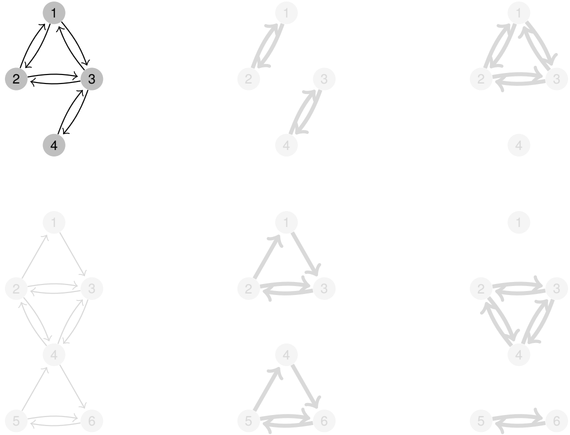
- Cycle formulation seems to be more than able to process foreseen number of patient-donor pairs in the KEP in Portugal
- Besides, it may allow to treat slightly different objectives:
  - produce robust solutions
  - maximize expectation of the number of transplants
- What if the *“market”* becomes the European Union?

U.PORTO
FC FACULDADE DE CIÊNCIAS UNIVERSIDADE DO PORTO

Maximizing expectation

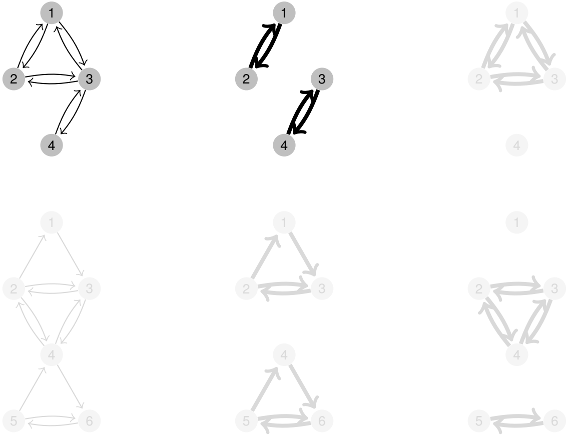
- Basis: cycle formulation
- On standard approach: cycle evaluation is the number of arcs in the cycle (*i.e.*, the *number of transplants*)
- Our proposal: use the expectation of the number of transplants instead
- Problem: not straightforward to tackle. . .
  1. computation of the expectation is heavy, even for small cycles
  2. optimization is just a small part in the solution process. . .

Unreliable vertices

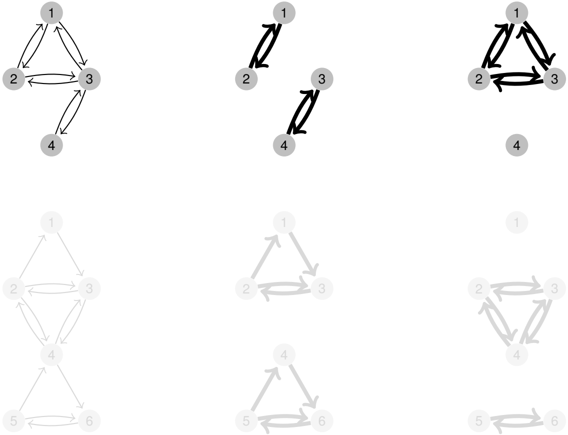


U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

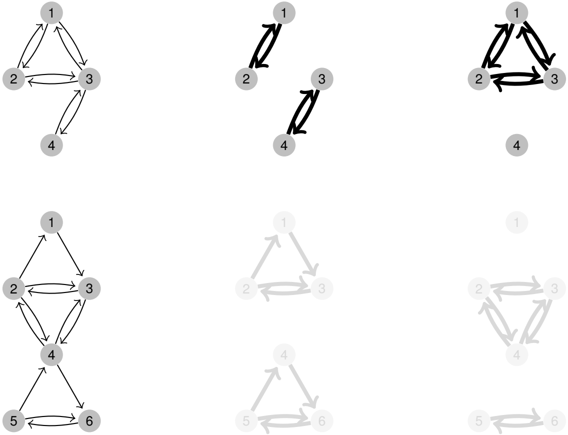
Unreliable vertices



Unreliable vertices

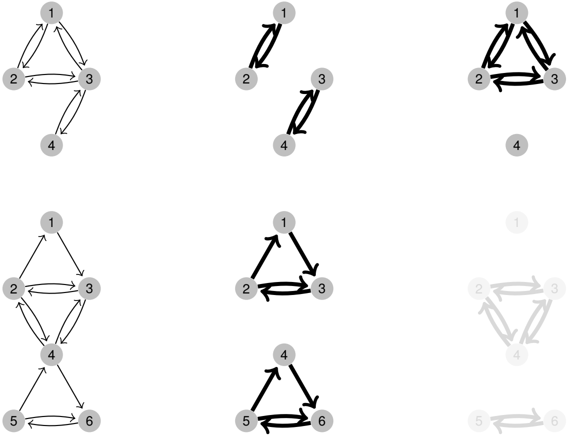


Unreliable vertices

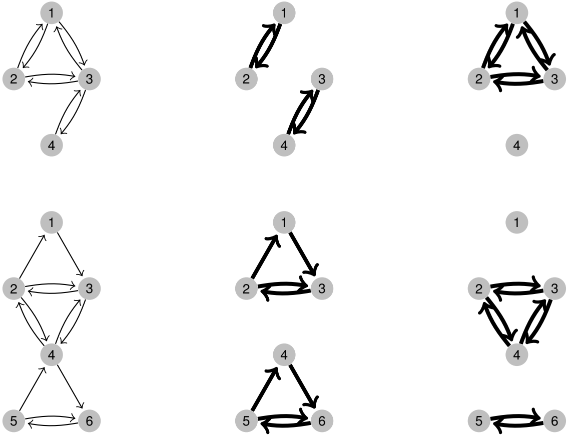


U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

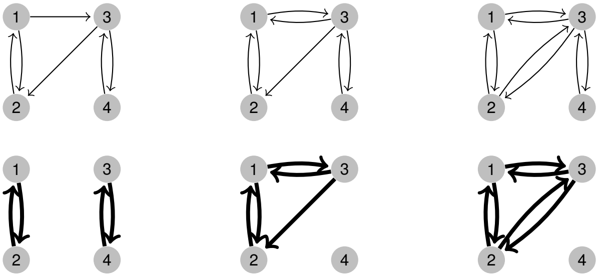
Unreliable vertices



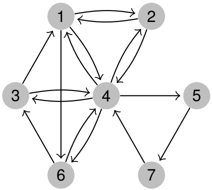
Unreliable vertices



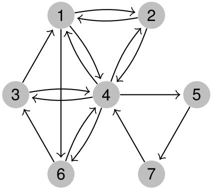
Unreliable arcs

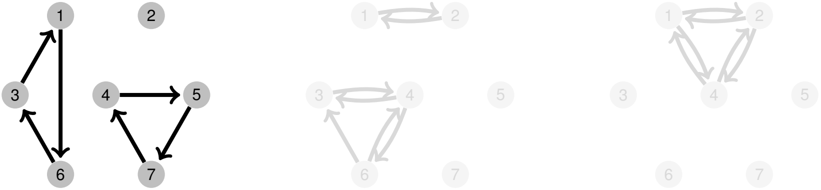


Arc withdrawal

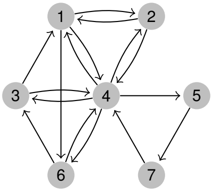


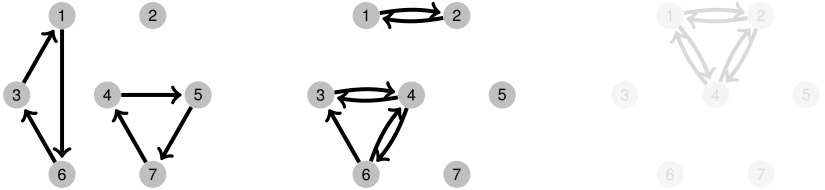
Arc withdrawal



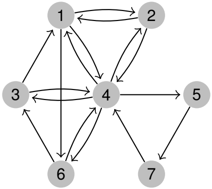


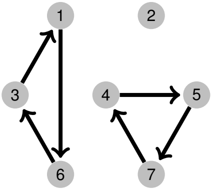
Arc withdrawal

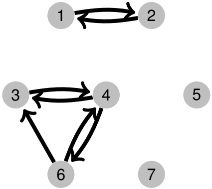


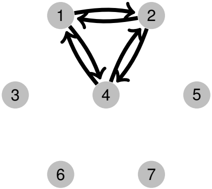


Arc withdrawal









Solution procedure

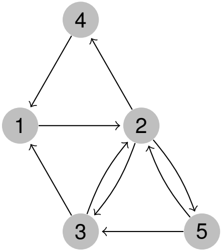
- Preprocessing
- Solution optimization
- Implementation

Solution procedure: preprocessing

- Preprocessing
  1. prepare a database of cycle configurations for the relevant sizes
  2. precompute formulas for expectations for these configurations
     HARD
  3. store this information in a database
- Example:
- Use the expectation as objective coefficient for each cycle

Solution procedure: preprocessing

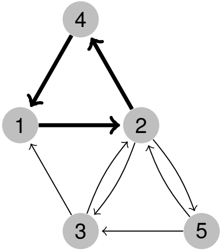
- Preprocessing
  1. prepare a database of cycle configurations for the relevant sizes
  2. precompute formulas for expectations for these configurations
     HARD
  3. store this information in a database
- Example:



- Use the expectation as objective coefficient for each cycle

Solution procedure: preprocessing

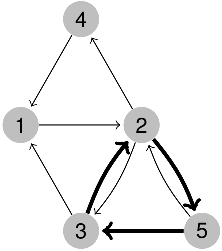
- Preprocessing
  1. prepare a database of cycle configurations for the relevant sizes
  2. precompute formulas for expectations for these configurations
     HARD
  3. store this information in a database
- Example: expectation formula?



- Use the expectation as objective coefficient for each cycle

Solution procedure: preprocessing

- Preprocessing
  1. prepare a database of cycle configurations for the relevant sizes
  2. precompute formulas for expectations for these configurations
     HARD
  3. store this information in a database
- Example: expectation formula?



- Use the expectation as objective coefficient for each cycle

Solution procedure: cycle configuration database

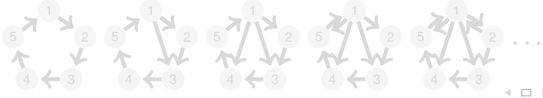
- Two-vertex graphs (1 graph)

vertices:	$2\,(1-p_1)\,(1-p_2)$
arcs:	$2\,(1-p_{12})\,(1-p_{21})$
both:	$2\,(1-p_1)\,(1-p_2)\,(1-p_{12})\,(1-p_{21})$

- Three-vertex graphs (4 graphs)

- Four-vertex graphs (61 graphs)

- Five-vertex graphs (3725 graphs)

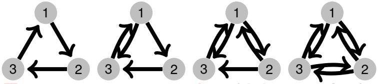


Solution procedure: cycle configuration database

- Two-vertex graphs (1 graph)

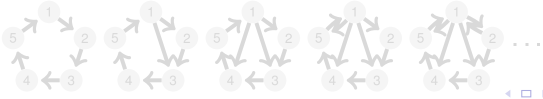
vertices:	$2\,(1-p_1)\,(1-p_2)$
arcs:	$2\,(1-p_{12})\,(1-p_{21})$
both:	$2\,(1-p_1)\,(1-p_2)\,(1-p_{12})\,(1-p_{21})$

- Three-vertex graphs (4 graphs)



- Four-vertex graphs (61 graphs)

- Five-vertex graphs (3725 graphs)

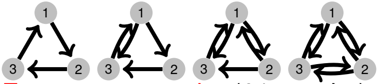


Solution procedure: cycle configuration database

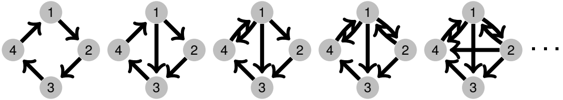
- Two-vertex graphs (1 graph)

vertices:	$2\,(1-p_1)\,(1-p_2)$
arcs:	$2\,(1-p_{12})\,(1-p_{21})$
both:	$2\,(1-p_1)\,(1-p_2)\,(1-p_{12})\,(1-p_{21})$

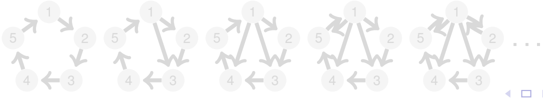
- Three-vertex graphs (4 graphs)



- Four-vertex graphs (61 graphs)



- Five-vertex graphs (3725 graphs)

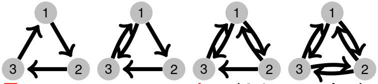


Solution procedure: cycle configuration database

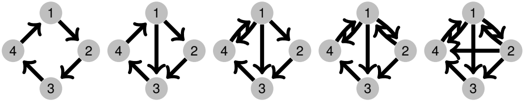
- Two-vertex graphs (1 graph)

vertices: $2\,(1 - p_1)\,(1 - p_2)$
arcs: $2\,(1 - p_{12})\,(1 - p_{21})$
both: $2\,(1 - p_1)\,(1 - p_2)\,(1 - p_{12})\,(1 - p_{21})$

- Three-vertex graphs (4 graphs)

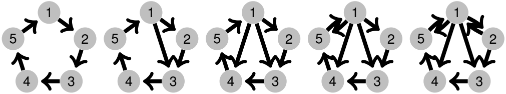


- Four-vertex graphs (61 graphs)



…

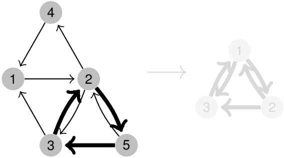
- Five-vertex graphs (3725 graphs)



…

- ...

1. Match enumerated graph with one in the database



2. Extract expectation formula from the database

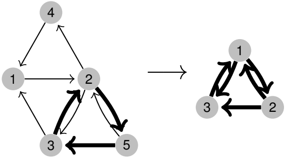
$3 - p_{21}p_{23} - p_{21}p_{31}p_{23} + p_{21}p_{32}p_{23} + p_{21}p_{31}p_{32}p_{23} + p_{31}p_{32}p_{23} - p_{32}p_{23} - p_{21}p_{31} - p_{21}p_{31}p_{32} - p_{31}p_{32} + p_{13}\left(p_{23}\left(p_{31} - 1\right)\left(-p_{32} + p_{21}\left(p_{32} + 1\right) + 1\right) - \left(p_{21} - 1\right)p_{31}\left(p_{32} - 1\right)\right) + p_{12}\left(-\left(p_{23}\left(p_{31} - 1\right) + p_{31} + 1\right)p_{32} + p_{21}\left(p_{23}\left(p_{31} - 1\right)\left(p_{32} - 1\right) + p_{32} + p_{31}\left(p_{32} + 1\right) - 1\right) + p_{13}\left(p_{23}\left(p_{31} + 1\right)\left(p_3 \right.\right.$

3. Map probabilities from original graph to the one stored

p_2	$\leftrightarrow$	p_1	p_{25}	$\leftrightarrow$	p_{12}
p_5	$\leftrightarrow$	p_2	p_{52}	$\leftrightarrow$	p_{21}
p_3	$\leftrightarrow$	p_3	p_{53}	$\leftrightarrow$	p_{23}
			p_{32}	$\leftrightarrow$	p_{31}
			p_{23}	$\leftrightarrow$	p_{13}

4. Compute expectation for the cycle $(2 - 5 - 3)$; it will be its coefficient at the objective

1. Match enumerated graph with one in the database



2. Extract expectation formula from the database

$3 - p_{21}p_{23} - p_{21}p_{31}p_{23} + p_{21}p_{32}p_{23} + p_{21}p_{31}p_{32}p_{23} + p_{31}p_{32}p_{23} - p_{32}p_{23} - p_{21}p_{31} - p_{21}p_{31}p_{32} -$
$p_{31}p_{32} + p_{13}\left(p_{23}\left(p_{31} - 1\right)\left(-p_{32} + p_{21}\left(p_{32} + 1\right) + 1\right) - \left(p_{21} - 1\right)p_{31}\left(p_{32} - 1\right)\right) +$
$p_{12}\left(-\left(p_{23}\left(p_{31} - 1\right) + p_{31} + 1\right)p_{32} + p_{21}\left(p_{23}\left(p_{31} - 1\right)\left(p_{32} - 1\right) + p_{32} + p_{31}\left(p_{32} + 1\right) - 1\right) + p_{13}\left(p_{23}\left(p_{31} + 1\right)\left(p_{3}\right.\right.\right.$

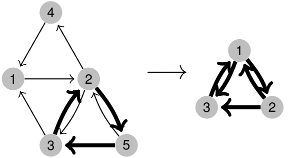
3. Map probabilities from original graph to the one stored

p_2	$\leftrightarrow$	p_1	p_{25}	$\leftrightarrow$	p_{12}
p_5	$\leftrightarrow$	p_2	p_{52}	$\leftrightarrow$	p_{21}
p_3	$\leftrightarrow$	p_3	p_{53}	$\leftrightarrow$	p_{23}
			p_{32}	$\leftrightarrow$	p_{31}
			p_{23}	$\leftrightarrow$	p_{13}

4. Compute expectation for the cycle $(2 - 5 - 3)$; it will be its coefficient at the objective

U.PORTO
FC FACULDADE DE CIÊNCIAS UNIVERSIDADE DO PORTO

1. Match enumerated graph with one in the database



2. Extract expectation formula from the database

$$3 - p_{21}p_{23} - p_{21}p_{31}p_{23} + p_{21}p_{32}p_{23} + p_{21}p_{31}p_{32}p_{23} + p_{31}p_{32}p_{23} - p_{32}p_{23} - p_{21}p_{31} - p_{21}p_{31}p_{32} -$$
$$p_{31}p_{32} + p_{13}\left(p_{23}\left(p_{31} - 1\right)\left(-p_{32} + p_{21}\left(p_{32} + 1\right) + 1\right) - \left(p_{21} - 1\right)p_{31}\left(p_{32} - 1\right)\right) +$$
$$p_{12}\left(-\left(p_{23}\left(p_{31} - 1\right) + p_{31} + 1\right)p_{32} + p_{21}\left(p_{23}\left(p_{31} - 1\right)\left(p_{32} - 1\right) + p_{32} + p_{31}\left(p_{32} + 1\right) - 1\right) + p_{13}\left(p_{23}\left(p_{31} + 1\right)\left(p_3\right.\right.$$

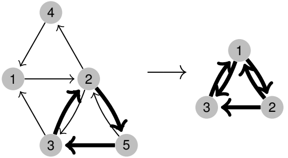
3. Map probabilities from original graph to the one stored

p_2	$\leftrightarrow$	p_1	p_{25}	$\leftrightarrow$	p_{12}
p_5	$\leftrightarrow$	p_2	p_{52}	$\leftrightarrow$	p_{21}
p_3	$\leftrightarrow$	p_3	p_{53}	$\leftrightarrow$	p_{23}
			p_{32}	$\leftrightarrow$	p_{31}
			p_{23}	$\leftrightarrow$	p_{13}

4. Compute expectation for the cycle $(2 - 5 - 3)$; it will be its coefficient at the objective

U.PORTO
FC FACULDADE DE CIÊNCIAS UNIVERSIDADE DO PORTO

1. Match enumerated graph with one in the database



2. Extract expectation formula from the database

$3 - p_{21}p_{23} - p_{21}p_{31}p_{23} + p_{21}p_{32}p_{23} + p_{21}p_{31}p_{32}p_{23} + p_{31}p_{32}p_{23} - p_{32}p_{23} - p_{21}p_{31} - p_{21}p_{31}p_{32} -$

$p_{31}p_{32} + p_{13}\left(p_{23}\left(p_{31}-1\right)\left(-p_{32}+p_{21}\left(p_{32}+1\right)+1\right)-\left(p_{21}-1\right)p_{31}\left(p_{32}-1\right)\right)+$

$p_{12}\left(-\left(p_{23}\left(p_{31}-1\right)+p_{31}+1\right)p_{32}+p_{21}\left(p_{23}\left(p_{31}-1\right)\left(p_{32}-1\right)+p_{32}+p_{31}\left(p_{32}+1\right)-1\right)+p_{13}\left(p_{23}\left(p_{31}+1\right)\left(p_3\right.\right.\right.$

3. Map probabilities from original graph to the one stored

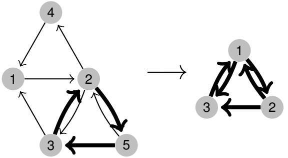
p_2	$\leftrightarrow$	p_1
p_5	$\leftrightarrow$	p_2
p_3	$\leftrightarrow$	p_3

p_{25}	$\leftrightarrow$	p_{12}
p_{52}	$\leftrightarrow$	p_{21}
p_{53}	$\leftrightarrow$	p_{23}
p_{32}	$\leftrightarrow$	p_{31}
p_{23}	$\leftrightarrow$	p_{13}

4. Compute expectation for the cycle $(2-5-3)$; it will be its coefficient at the objective

U.PORTO
FC FACULDADE DE CIÊNCIAS UNIVERSIDADE DO PORTO

1. Match enumerated graph with one in the database



2. Extract expectation formula from the database

$3 - p_{21}p_{23} - p_{21}p_{31}p_{23} + p_{21}p_{32}p_{23} + p_{21}p_{31}p_{32}p_{23} + p_{31}p_{32}p_{23} - p_{32}p_{23} - p_{21}p_{31} - p_{21}p_{31}p_{32} -$

$p_{31}p_{32} + p_{13}\left(p_{23}\left(p_{31} - 1\right)\left(-p_{32} + p_{21}\left(p_{32} + 1\right) + 1\right) - \left(p_{21} - 1\right)p_{31}\left(p_{32} - 1\right)\right) +$

$p_{12}\left(-\left(p_{23}\left(p_{31} - 1\right) + p_{31} + 1\right)p_{32} + p_{21}\left(p_{23}\left(p_{31} - 1\right)\left(p_{32} - 1\right) + p_{32} + p_{31}\left(p_{32} + 1\right) - 1\right) + p_{13}\left(p_{23}\left(p_{31} + 1\right)\left(p_{3}\right.\right.\right.$

3. Map probabilities from original graph to the one stored

p_2	$\leftrightarrow$	p_1		p_{25}	$\leftrightarrow$	p_{12}
				p_{52}	$\leftrightarrow$	p_{21}
p_5	$\leftrightarrow$	p_2		p_{53}	$\leftrightarrow$	p_{23}
				p_{32}	$\leftrightarrow$	p_{31}
p_3	$\leftrightarrow$	p_3		p_{23}	$\leftrightarrow$	p_{13}

4. Compute expectation for the cycle $(2 - 5 - 3)$; it will be its coefficient at the objective

Solution procedure: solution optimization

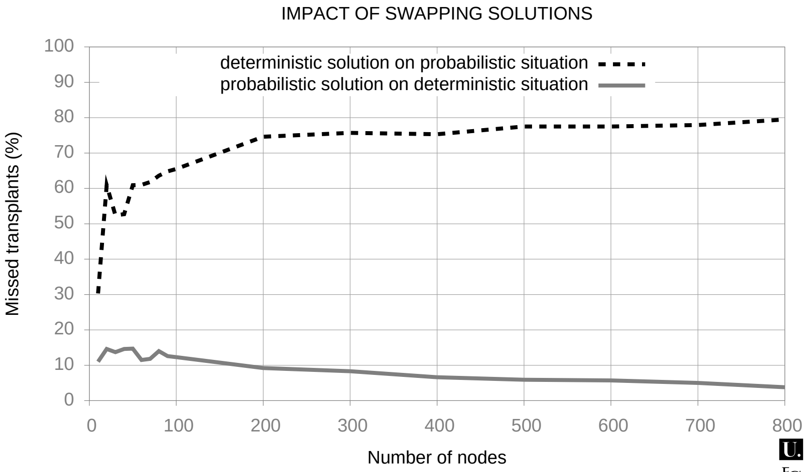
- Solution optimization
  1. read instance: compatibility between pairs, failure probability for vertices/arcs
  2. prepare compatibility graph
  3. enumerate cycles of relevant size
  4. setup optimization model
     1. one variable for each cycle
     2. constraints: each vertex in at most one cycle
     3. objective coefficient: expectation of number of transplants HARD
  5. solve optimization model easy?!!

U.PORTO
FACULDADE DE CIÊNCIAS UNIVERSIDADE DO PORTO

Solution procedure: implementation

- Implementation
  1. contact selected pairs
  2. verify solution (check back outs)
  3. make last-minute compatibility check
  4. make transplants

Results: cross-formulation performance



Conclusions

- There are many applications of information technologies in health care
- Applications involve many disciplines in computer science and informatics
- KEP: case where welfare of patients can be maximized
  - number of transplants
  - robustness of the solution
  - quality of the solution (maximize patient-donor compatibility)
- Careful implementation of operations research program leads to significant social benefits