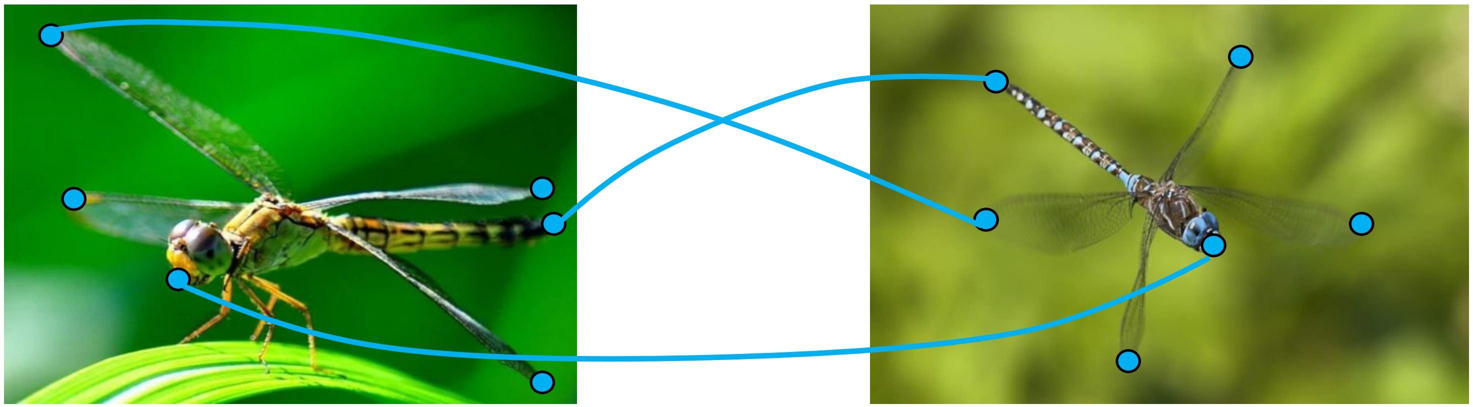
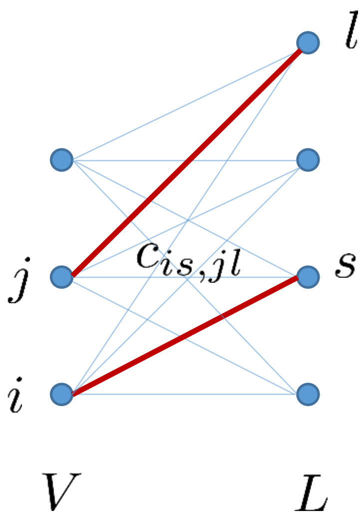
Algorithmic Techniques for Graph Matching and Their Recent Comparison Study

Bogdan Savchynskyy

24/09/2022

(Weighted) graph matching problem





$$\min_{x\in\{0,1\}^{V\times L}} \sum_{i,j\in V} \sum_{s,l\in L} c_{is,jl} x_{is} x_{jl}$$

$$\text{s.t.:} \sum_{s} x_{is} \leq 1 \;\forall i$$

$$\sum_{i} x_{is} \leq 1 \;\forall s$$

$$c_{is} = c_{is,is}$$

Related work and contribution

- More than 7 000 000 papers
- Hundreds of algorithms
- (At least) two communities: Operations research & Computer Vision
- Different goals: modeling, speed, precision

Related work and contribution

- More than 7 000 000 papers
- Hundreds of algorithms
- (At least) two communities: Operations research & Computer Vision
- Different goals: modeling, speed, precision

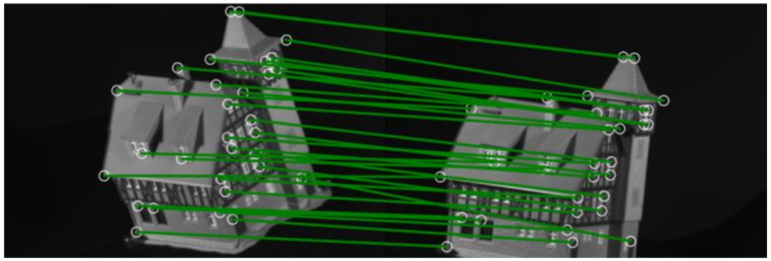
We limit our exposition to:

- Computer vision applications
- Ready cost matrix, no modeling/learning aspects
- Arbitrary costs (Lawler form)
- Open source code

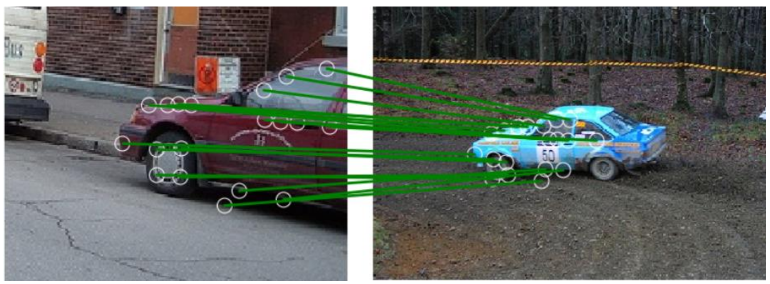
[Haller et al. 2022. A comparative study of graph matching algorithms in computer vision]

https://vislearn.github.io/gmbench/

Most popular datasets for evaluation



	hotel | **house-dense** | **house-sparse**
$\lvert V \rvert$ =	30
Density:	13% | 13% | 1.5 %
#instances:	105



	car
$\lvert V \rvert$ =	19-49
Density:	2.9%
#instances:	30

	motor(bike)
$\lvert V \rvert$ =	15-52
Density:	3.8%
#instances:	20

Most popular algorithmic baselines

- *Graduated assignment* (ga) — S. Gold, A. Rangarajan, 1996. A graduated assignment algorithm for graph matching.
- *Spectral matching* (sm) — M. Leordeanu, M. Hebert, 2005. A spectral technique for correspondence problems using pairwise constraints.
- *Spectral matching with affine constraints* (smac) — T. Cour, P. Srinivasan, J. Shi, 2007. Balanced graph matching
- *Integer projected fixed point* (ipfp-u/s) — M. Leordeanu, M. Hebert, R. Sukthankar, 2009. An integer projected fixed point method for graph matching and MAP inference
- *Reweighted random walks matching* (rrwm) — M. Cho, J. Lee, K. Mu Lee, 2010. Reweighted random walks for graph matching
- *Local sparse model* (lsm) — B. Jiang, J. Tang, C. Ding, B. Luo, 2015. A local sparse model for matching problem
- *Factorized graph matching* (ipfp-u/s) — F. Zhou, F. de la Torre, 2016. Factorized Graph Matching

Our findings

- Most popular datasets are easy to solve (exactly, < 1 sec)
- Most popular algorithmic baselines are very weak
- Even large problems can be solved fast:

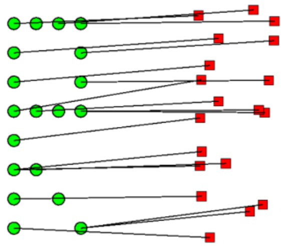
  V > 500, L > 1300, t < 1 sec

Outline

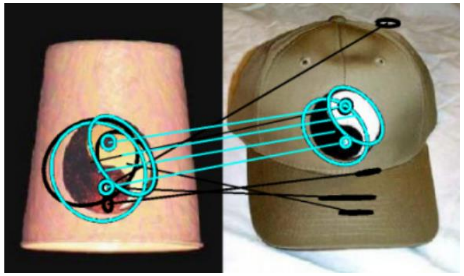
- Empirical evaluation
- Design of the best performing methods

Empirical evaluation

What additional datasets are in our benchmark



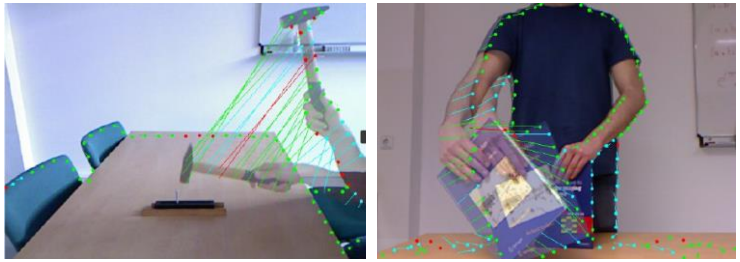
opengm
$|V|$ = 19-20
Density: 75%
#instances: 4



caltech-small
$|V|$ = 9-117
Density: 1%
#instances: 21

Additional small instances: $|V| < 120$

What datasets are in our benchmark

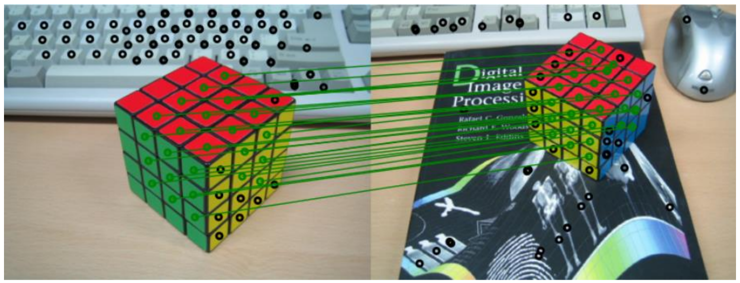


flow

$|V|$ = 48-126

Density: 0.4%

#instances: 6

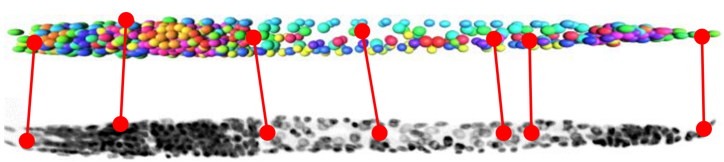


caltech-large

$|V|$ = 36-219

Density: 0.55%

#instances: 9

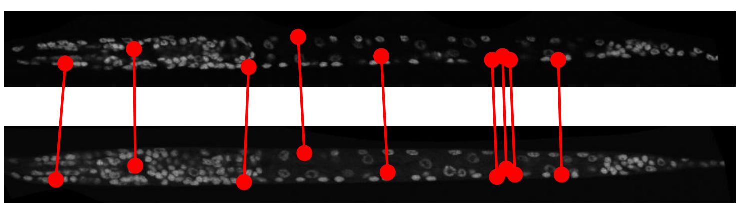


worms (atlas)

$|V|/|L|$ = 558/1300

Density: 0.00038%

#instances: 30



(worm) pairs

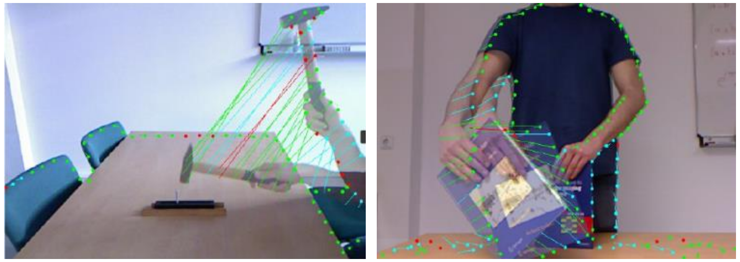
$|V|$ = 511-565

Density: 0.0019%

#instances: 16

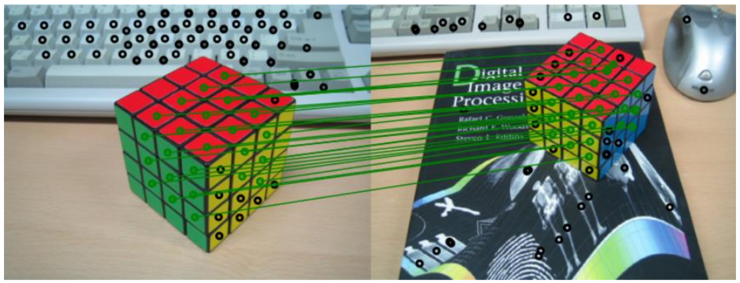
Additional large instances: $|V| > 120$

What datasets are in our benchmark



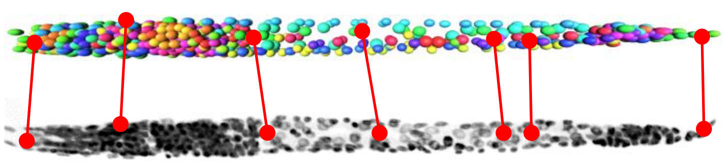
flow

$|V|$ = 48-126
Density: 0.4%
#instances: 6



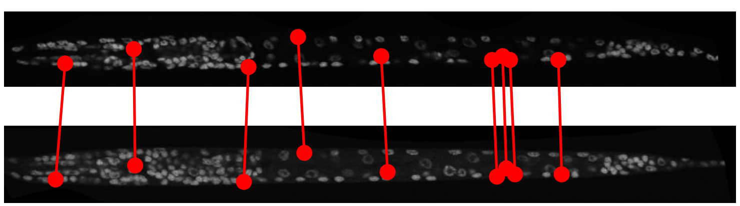
caltech-large

$|V|$ = 36-219
Density: 0.55%
#instances: 9



worms (atlas)

$|V|/|L|$ = 558/1300
Density: 0.00038%
#instances: 30



(worm) pairs

$|V|$ = 511-565
Density: 0.0019%
#instances: 16

Additional large instances: $|V| > 120$

11 datasets, 451 problem instances in total

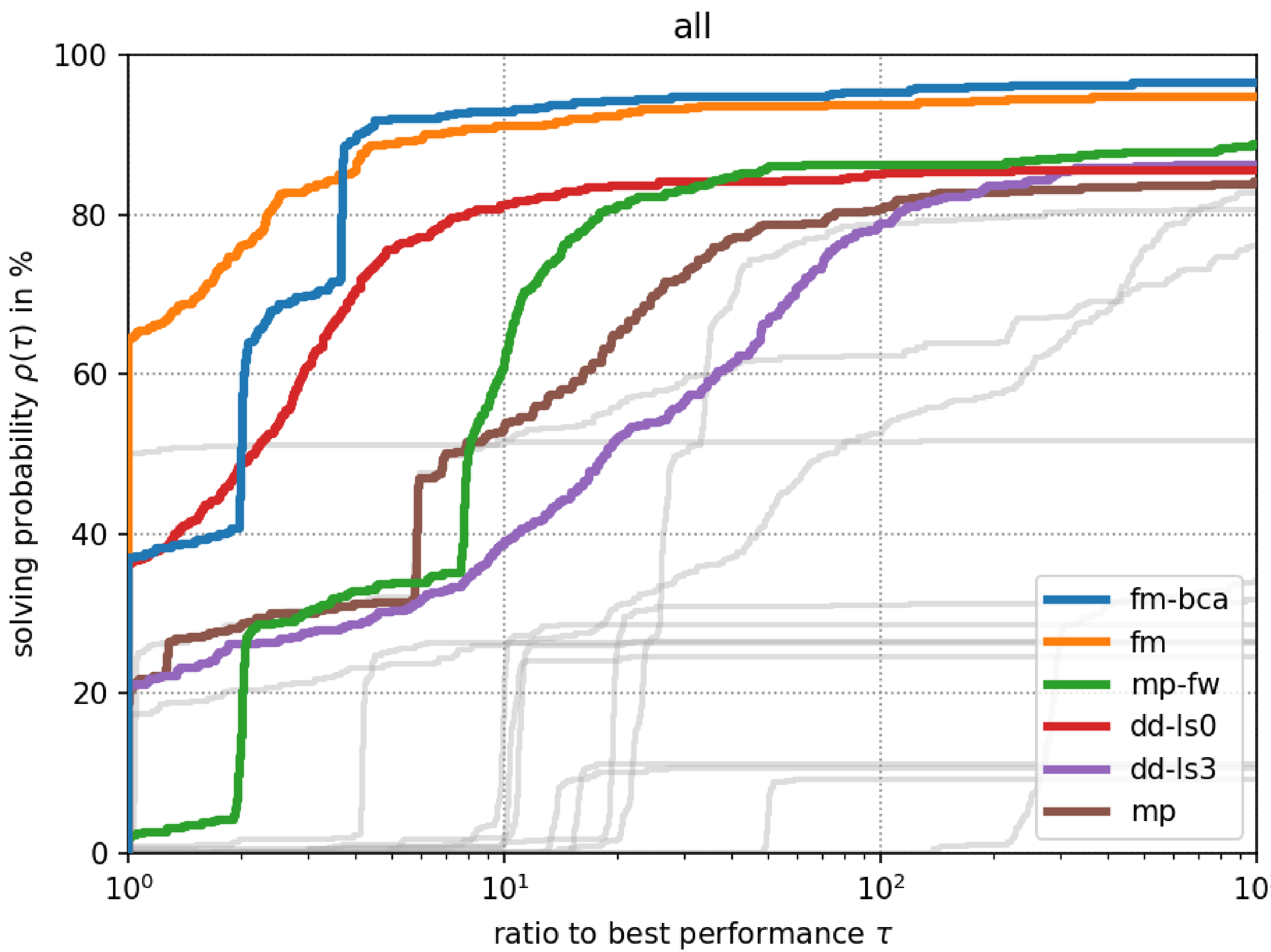
What algorithms we evaluate

	method	IQP	ILP	bijective	non-pos.	0-unary	lineariz.	norm	doubly	spectral	discret.	path fol.	fusion	duality	SGA	BCA	Matlab	C++
Primal methods	fgmd [69]	+		+								+					[68]	
	fm [31]		+										+					[32]
	fw [62]	+					+		+									[56]
	ga [27]	+		+			+		+		+						[20]	
	ipfps [46]	+		+	+		+		+		+						[44]	
	ipfpu [46]	+		+			+		+		+						[44]	
	lsm [33]	+		+	+			+			+						[66]	
	mpm [18]	+		+	+			+			+						[19]	
	pm [65]	+		+	+	+			+		+						[68]	
	rrwm [17]	+		+			+		+		+						[16]	
	smac [21]	+		+	+			+		+	+						[20]	
	sm [43]	+		+	+			+		+	+						[44]	
Duality-based	dd-ls(0/3/4) [59]		+											+	+			[38]
	fm-bca [31]		+										+	+		+		[32]
	hbp [67]		+	+										+		+	[66]	
	mp(-mcf/-fw) [57]		+											+		+		[56]

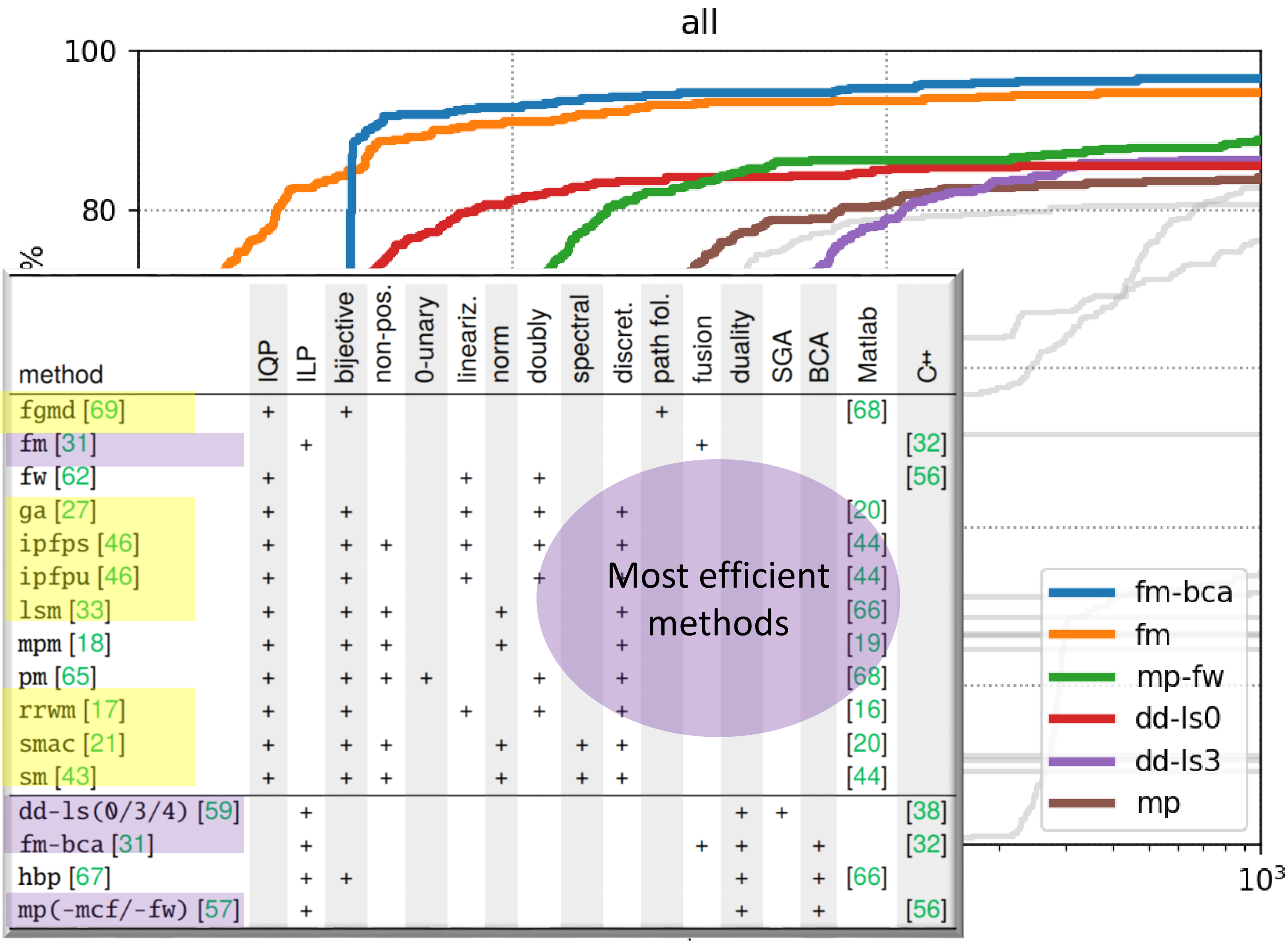
Most popular baselines

20 algorithms

What are the results of evaluation



What are the results of evaluation



Primal methods / Duality-based

method	IQP	ILP	bijective	non-pos.	0-unary	lineariz.	norm	doubly	spectral	discret.	path fol.	fusion	duality	SGA	BCA	Matlab	C++
fgmd [69]	+		+								+					[68]	
fm [31]		+										+					[32]
fw [62]	+					+		+									[56]
ga [27]	+		+			+		+		+						[20]	
ipfps [46]	+		+	+		+		+		+						[44]	
ipfpu [46]	+		+			+		+								[44]	
lsm [33]	+		+	+			+			+						[66]	
mpm [18]	+		+	+			+			+						[19]	
pm [65]	+		+	+	+			+		+						[68]	
rrwm [17]	+		+			+		+		+						[16]	
smac [21]	+		+	+			+		+	+						[20]	
sm [43]	+		+	+			+		+	+						[44]	
dd-ls(0/3/4) [59]		+											+	+			[38]
fm-bca [31]		+										+	+		+		[32]
hbp [67]		+	+										+		+	[66]	
mp(-mcf/-fw) [57]		+											+		+		[56]

Most efficient methods

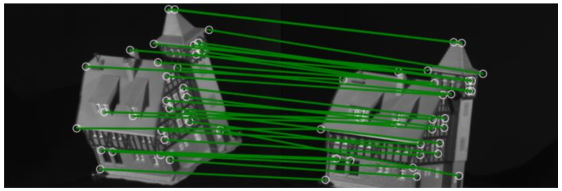
Evaluation peculiarities

- t = 1, 10, 100, 300 sec.
- Methods iteratively output current results.

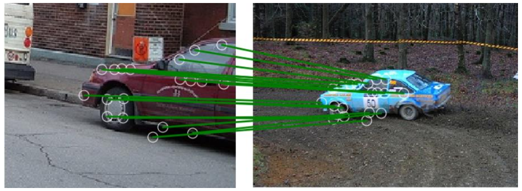
- Evaluation criteria: # optima (opt), *E* (obj), dual bound, accuracy (acc)

- Optimal solutions obtained with Gurobi or *E*=dual bound:
- Optimum found: 416 instances
- Optimum unknown: 35 instances

Most popular datasets



hotel | house-dense | house-sparse



cars

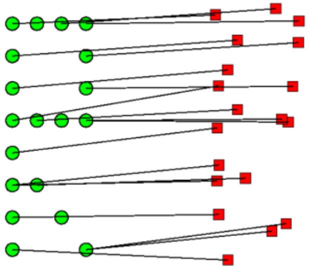
motor

	hotel		house-dense		house-sparse		cars		motor	
	opt %	**acc %**	**opt %**	**acc %**	**opt %**	**acc %**	**opt %**	**acc %**	**opt %**	**acc %**
fm-bca	**100**	**100**	**100**	**100**	**100**	**100**	93	**92**	**100**	**97**
dd-ls0	**100**	**100**	**100**	**100**	**100**	**100**	**97**	91	**100**	**97**
fgmd	96	98	77	89	**100**	**100**	83	89	85	**97**

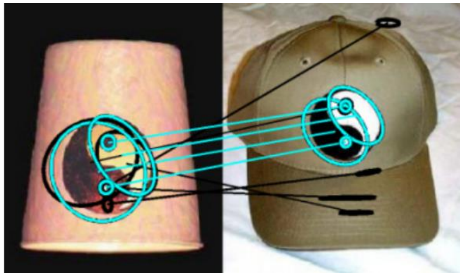
Max. run-time: 1 sec, fgmd – 300 sec

All solved in < 1 sec!

Small datasets



opengm

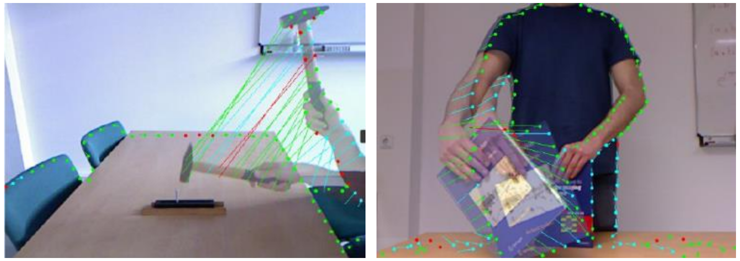


caltech-small

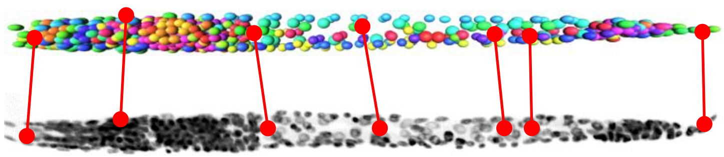
	opengm		caltech-small		
	opt %	**obj**	**opt %**	**obj**	**acc %**
fm	**100**	**-171**	**57**	**-9040**	62
fm-bca	**100**	**-171**	43	-8943	62
fgmd	75	-166	43	-	-
ipfps	0	-95	19	-8983	**67**

Max. run-time: 10 sec, fgmd – 300 sec

Large datasets



flow



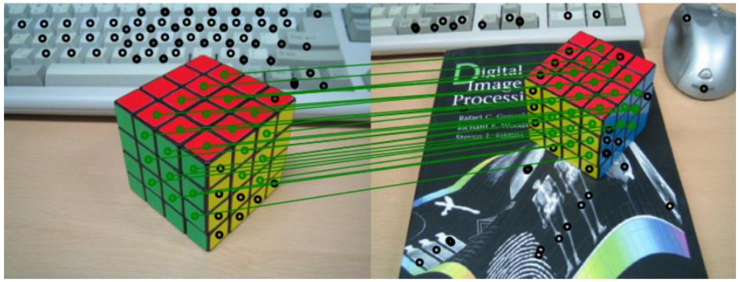
Worms, $|V|/|L|$ = 558/1300

	flow		worms	
	opt %	**obj**	**opt %**	**acc %**
fm	**100**	**-2840**	**93**	**89**
fm-bca	**100**	**-2840**	**93**	**89**
ga	0	-2469	0	0

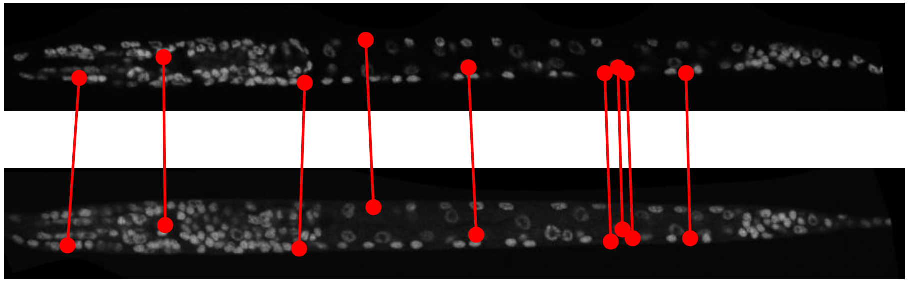
Max. run-time: 10 sec, ga – 300 sec

33 out of 36 instances optimally solved in < 1 sec!

Large datasets



caltech-large



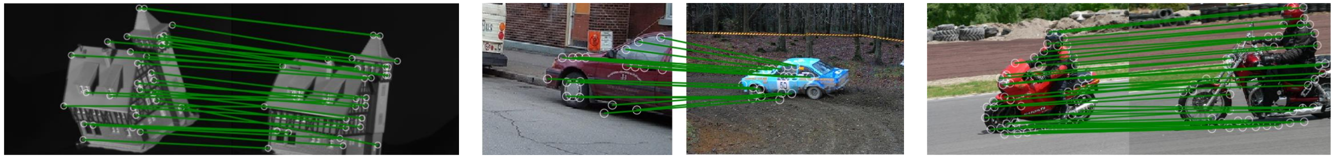
pairs

	caltech-large		pairs	
	obj	**acc %**	**obj**	**acc %**
fm-bca	-34039	51	**-65913**	**58**
fm	**-34117**	51.6	-65625	**54**
ipfpu	**-34216**	**52**	-35666	7

Max. run-time: 10 sec

Empirical conclusions

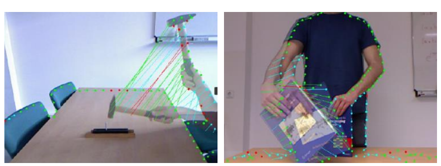
- Popular datasets are insufficient to show efficiency of new algorithms

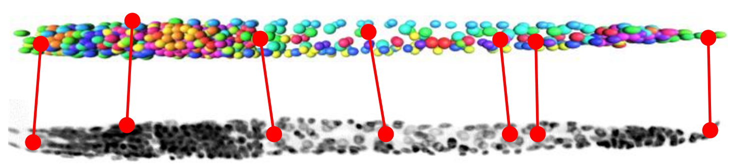


- Popular algorithms are insufficient as baselines

	flow		worms	
	opt %	**obj**	**opt %**	**acc %**
ga	0	-2469	0	0

- The most efficient methods are duality-based ones (e.g. fm-bca,dd-ls0,mp-fw) equipped with powerful primal heuristics
- Even large problem instances with |V|>200-500 can often be solved in < 1 sec





Ready to deep-learn in large scale?

Design of the best performing methods

Design of the best performing methods

1. Based on integer linear program (**ILP**) **representation**
2. **Optimize Lagrange dual** and simplify the cost structure of the problem
3. Run a **strong primal heuristic** on the simplified costs

- *Fusion moves* (fm, fm-bca) — L. Hutschenreiter et al. 2021, Fusion moves for graph matching
- *Message passing* (mp-XXX) — P. Swoboda et al. 2017, A study of Lagrangean decompositions and dual ascent solvers for graph matching
- *Dual decomposition*(dd-lsX) — L. Torresani et al. 2013, A dual decomposition approach to feature correspondence

Strong primal heuristic

3. Strong primal heuristic: Fusion moves

1. Generate proposal solutions
2. Fuse/merge/recombine/crossover it with the current one
3. Update the current solution
4. Goto 1.

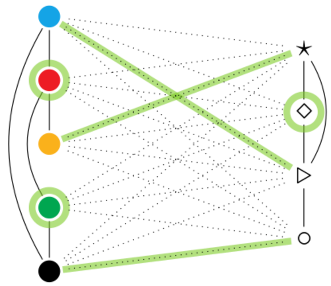
- ❑ *Fusion moves* (fm, fm-bca)

L. Hutschenreiter et al. 2021, Fusion moves for graph matching

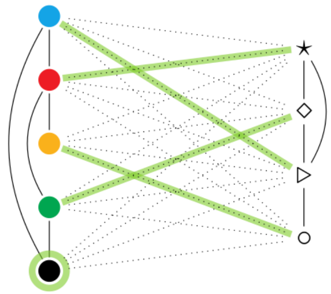
Fusion operation

3. Strong primal heuristic: 2) Fusion operation

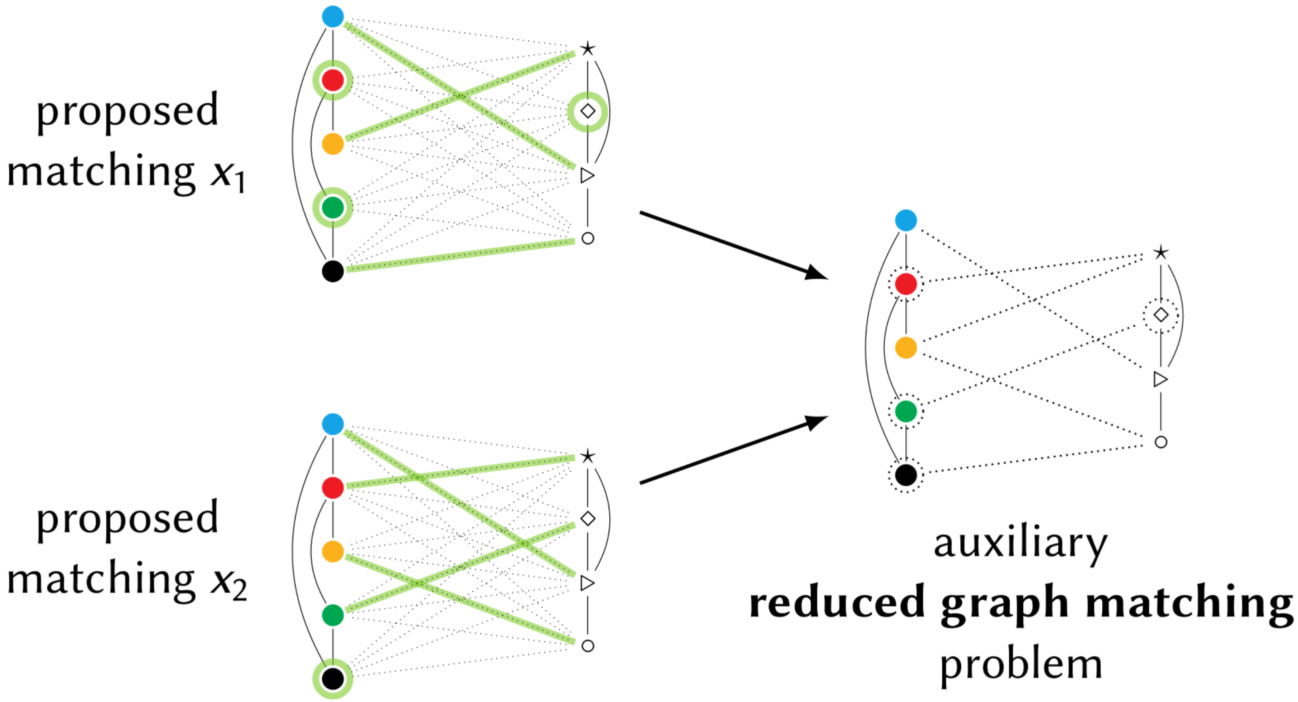
proposed matching x_1



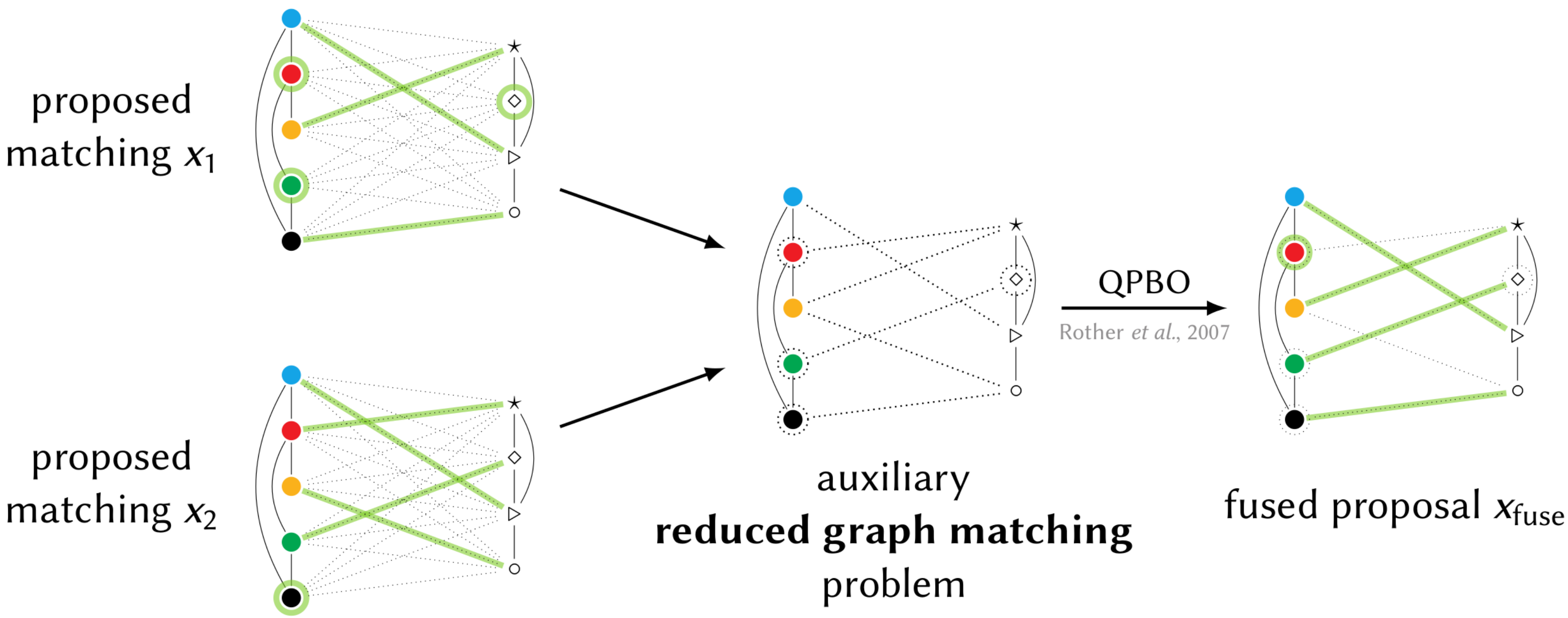
proposed matching x_2



3. Strong primal heuristic: 2) Fusion operation



3. Strong primal heuristic: 2) Fusion operation



$$E(\theta, x_{\text{fuse}}) \leq \min\left[E(\theta, x_1), E(\theta, x_2)\right]$$

Proposal generation

Proposal generation

Desired properties:

- **Diverse**
- **Low objective value**

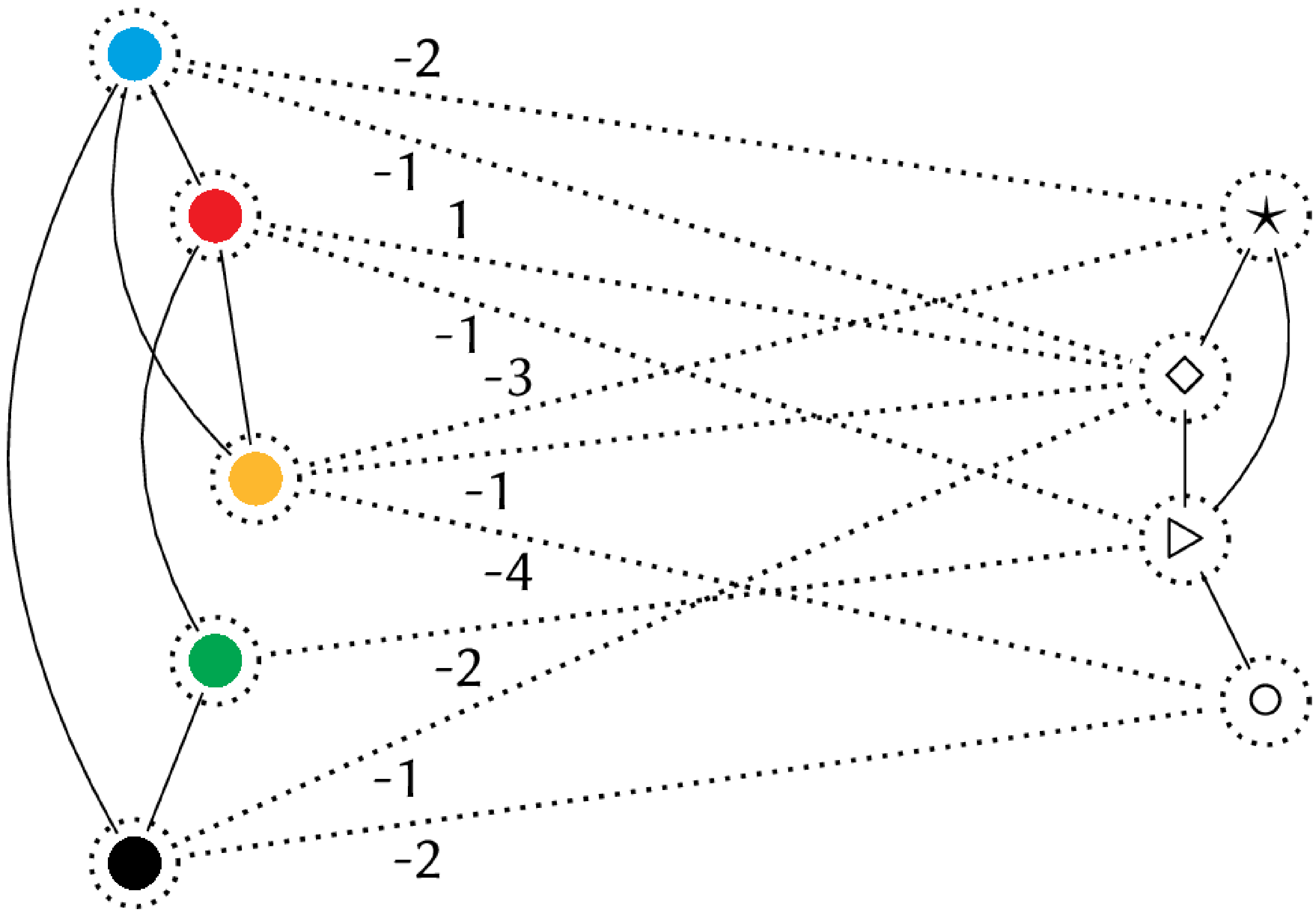
Proposal generation

Desired properties:

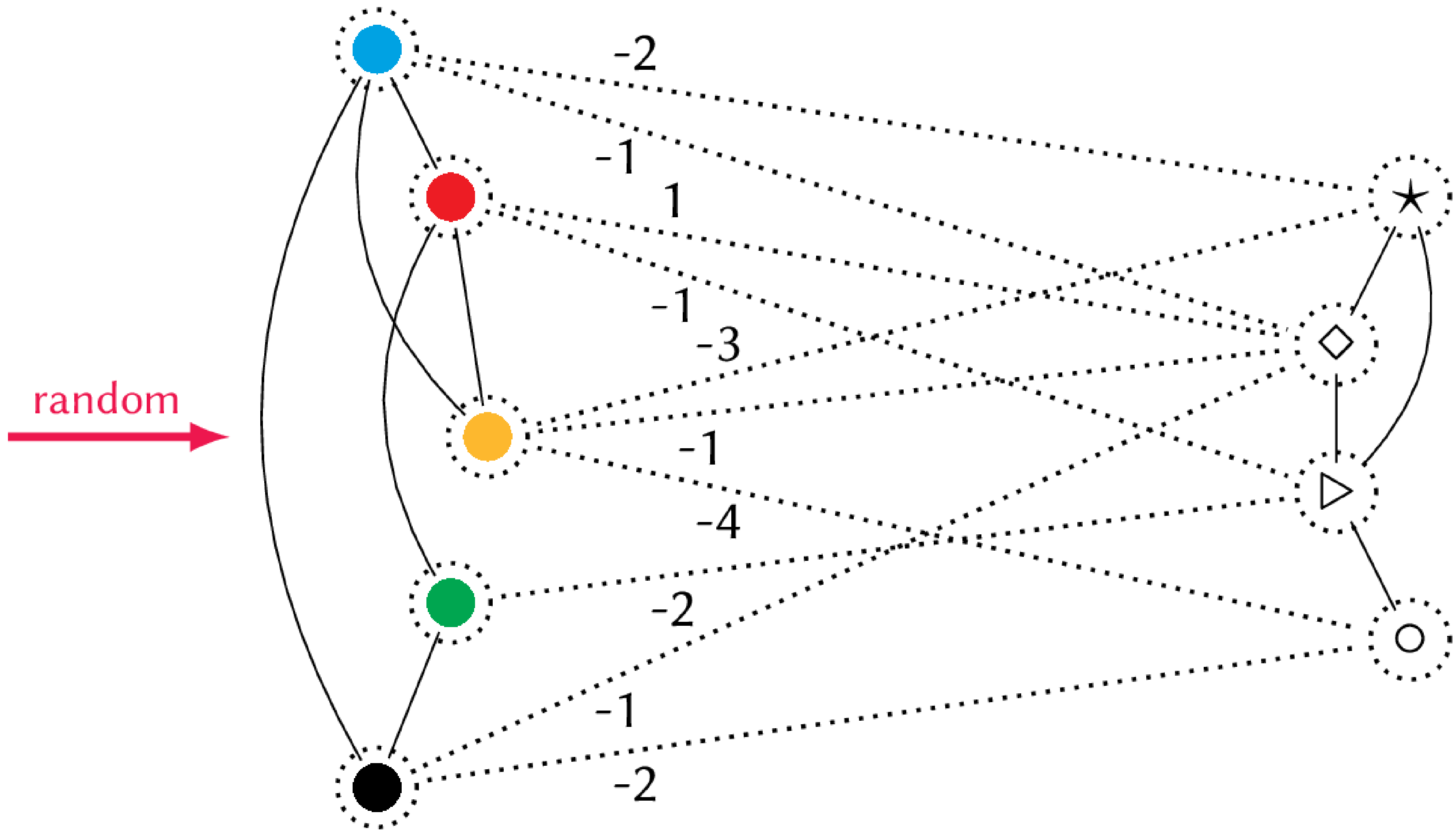
- **Diverse** →
- **Low objective value** →

Randomized greedy heuristic

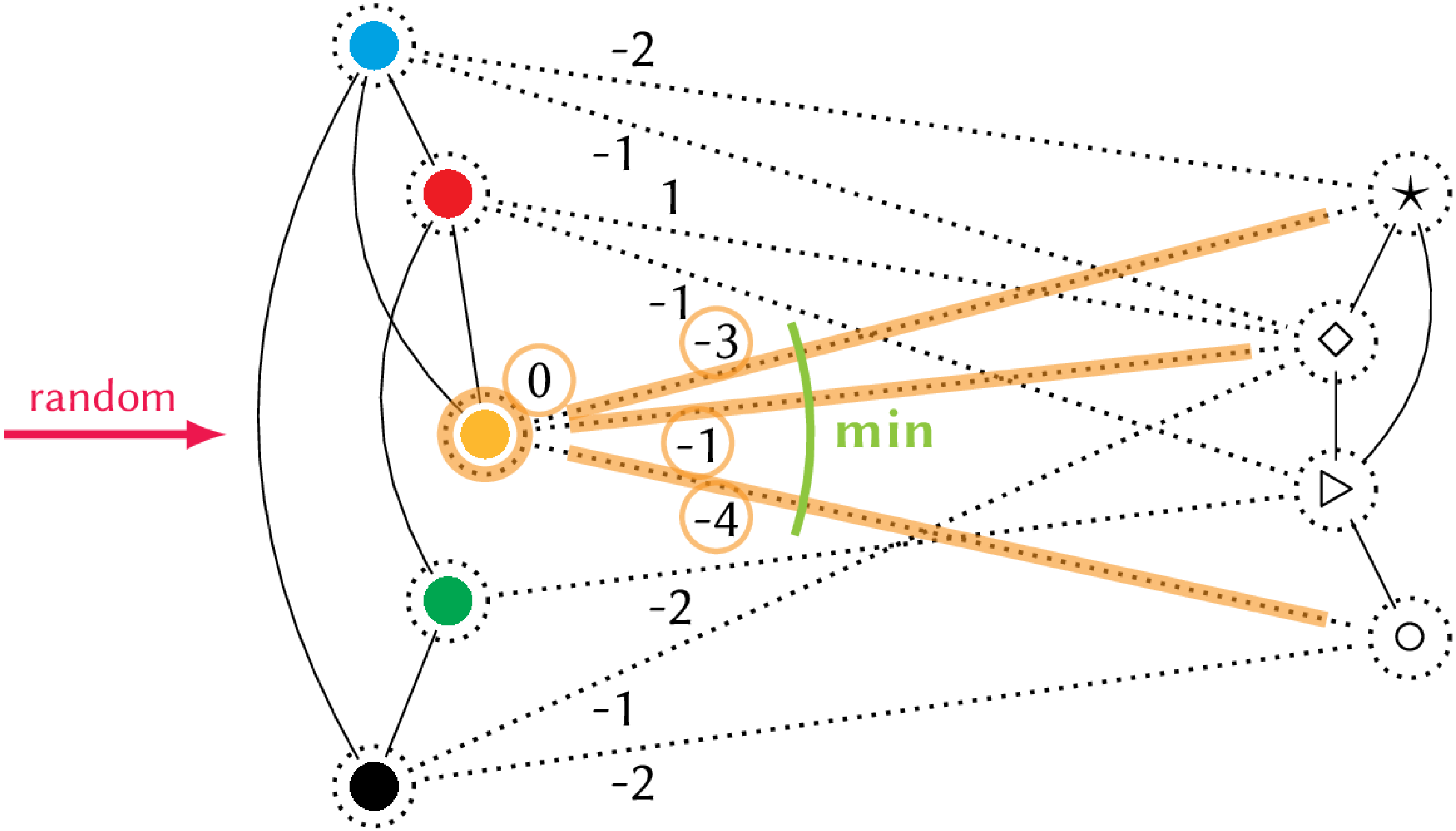
Randomized greedy heuristic



Randomized greedy heuristic

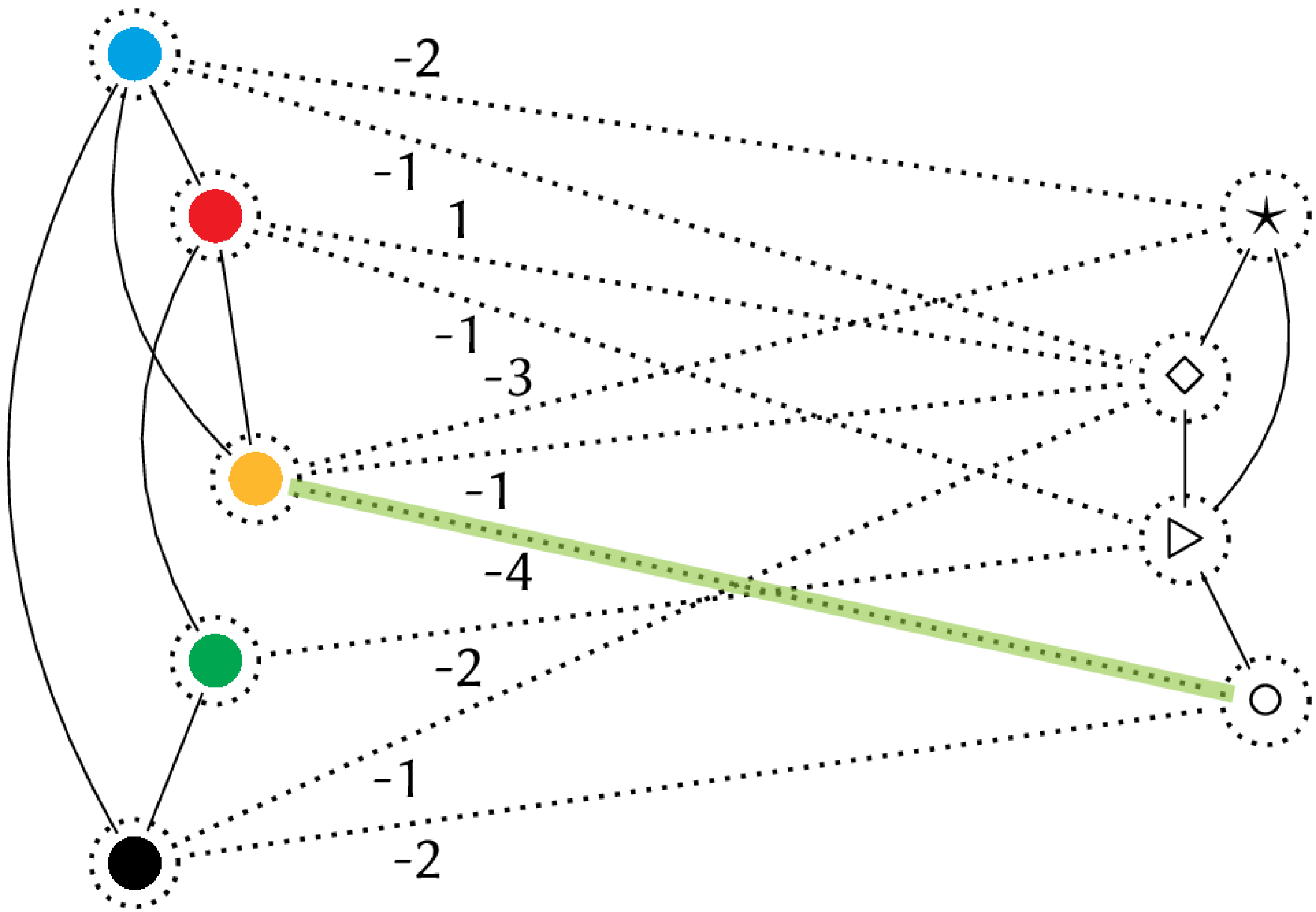


Randomized greedy heuristic

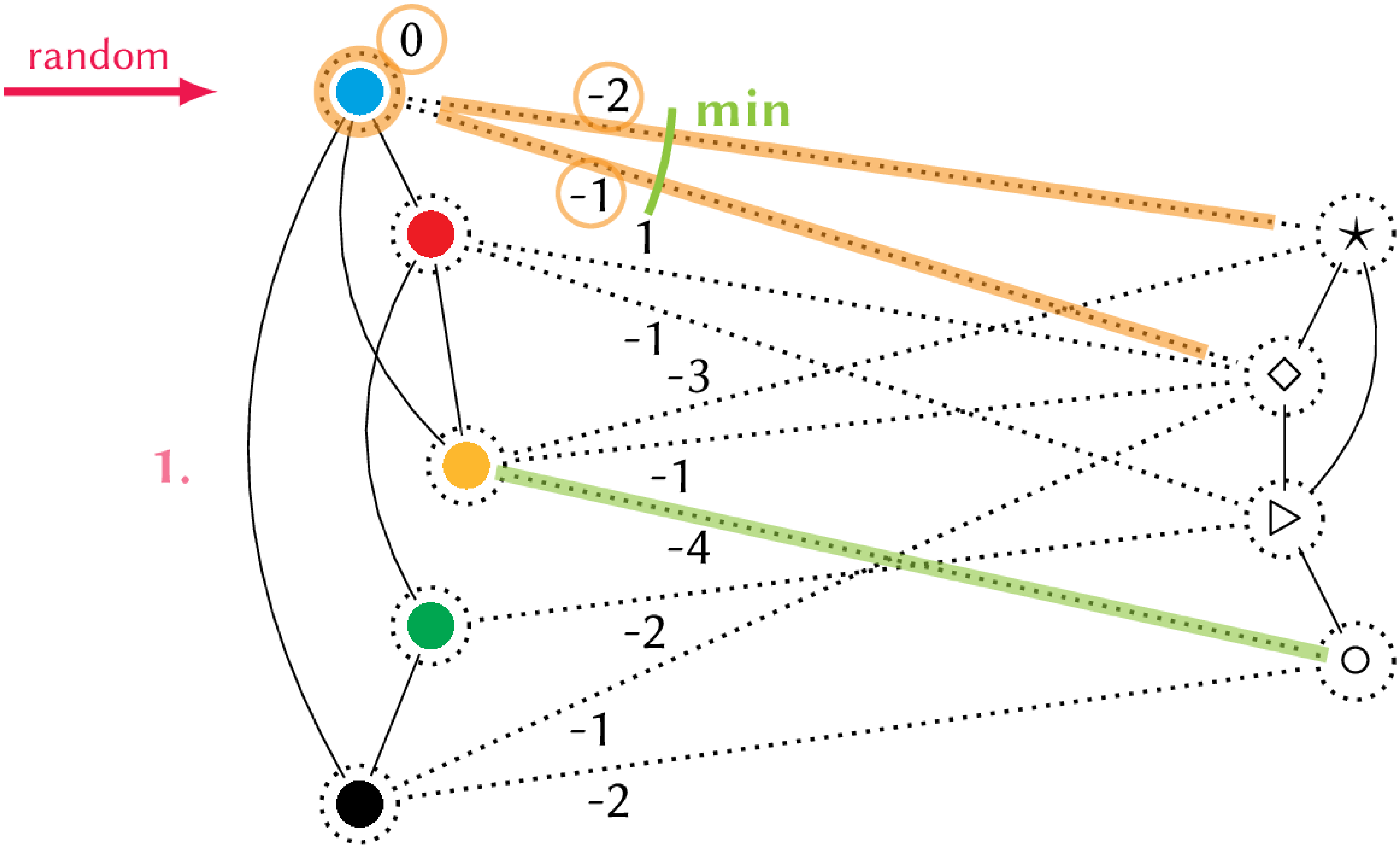


Randomized greedy heuristic

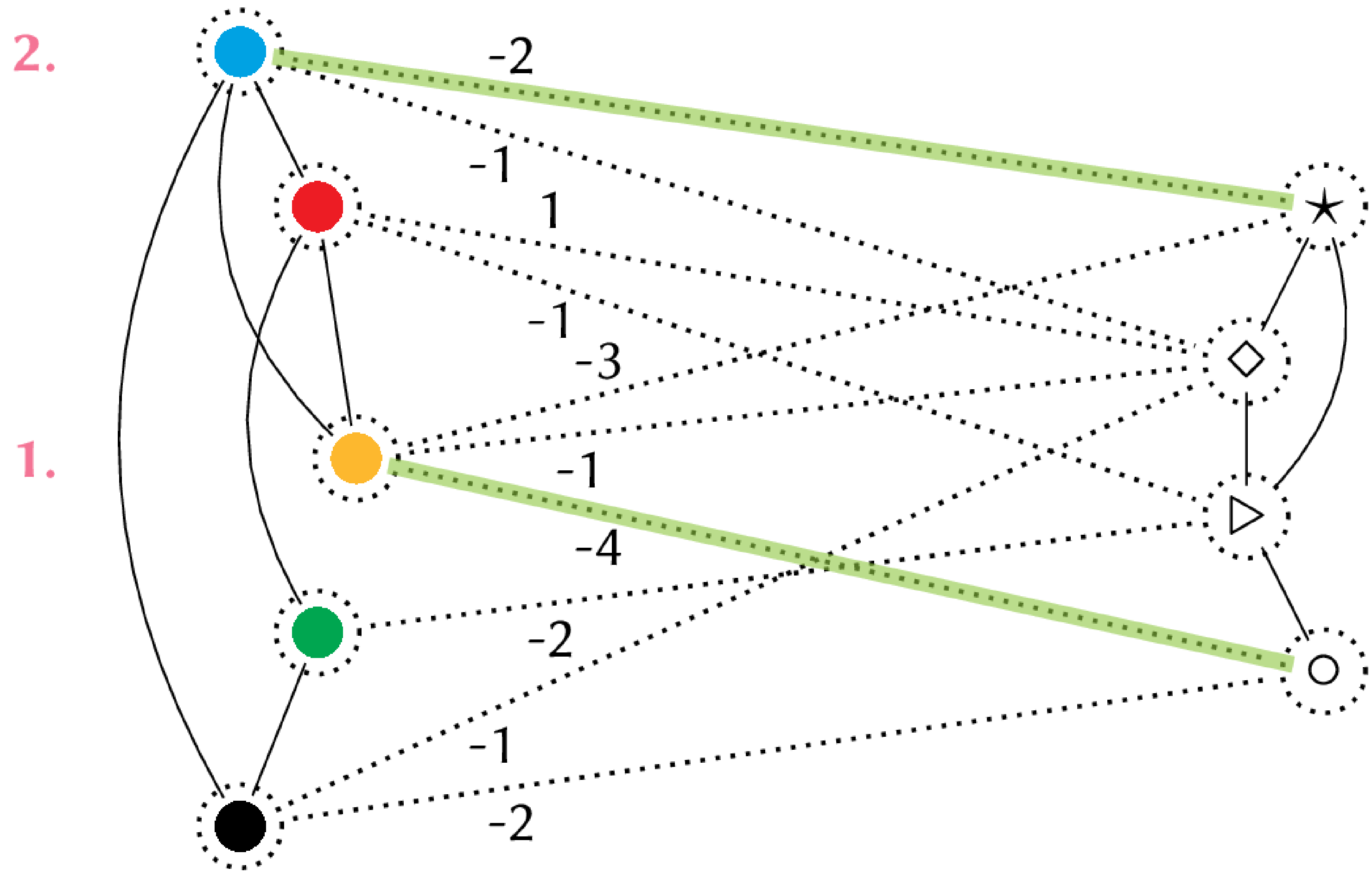
1.



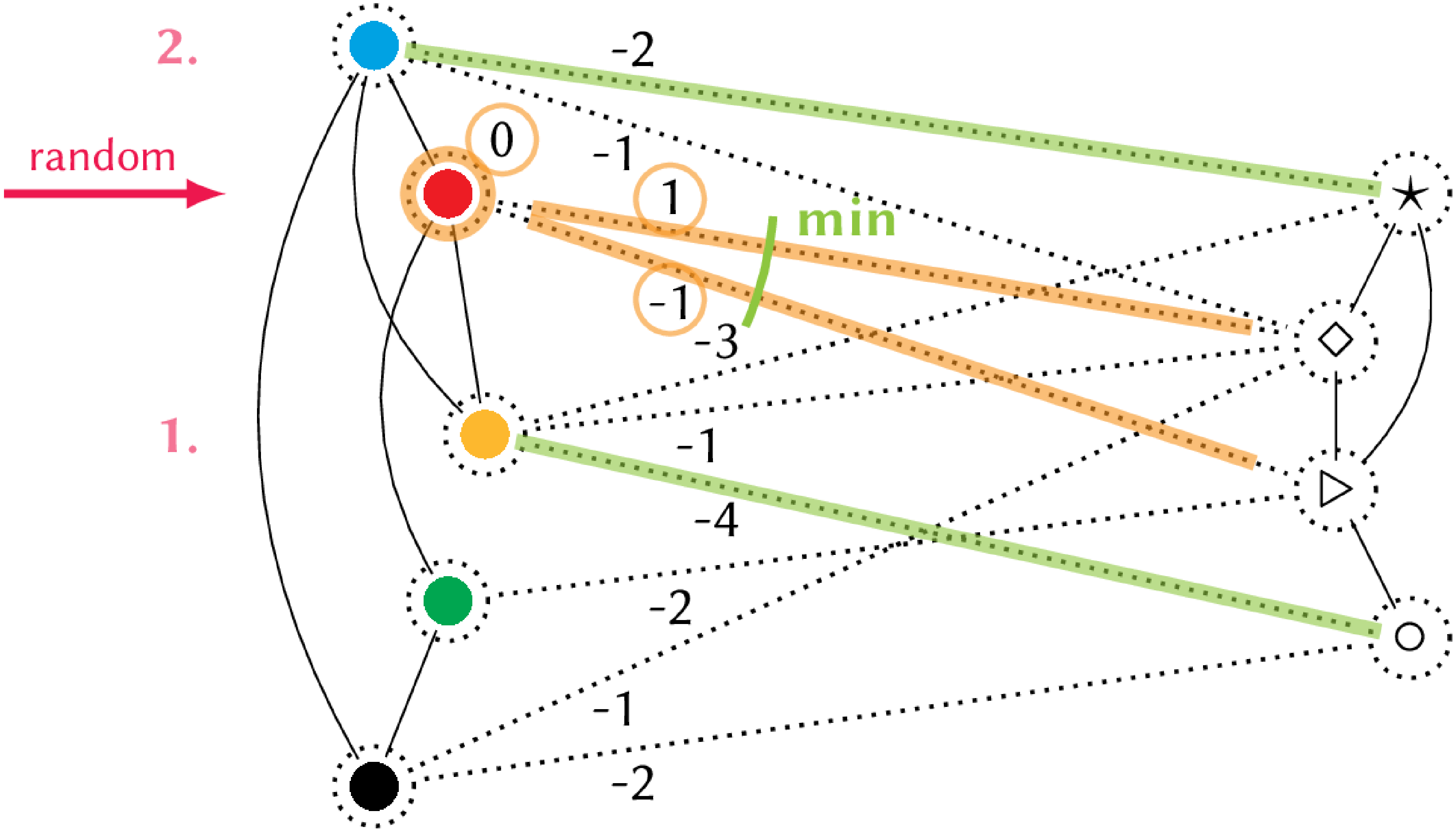
Randomized greedy heuristic



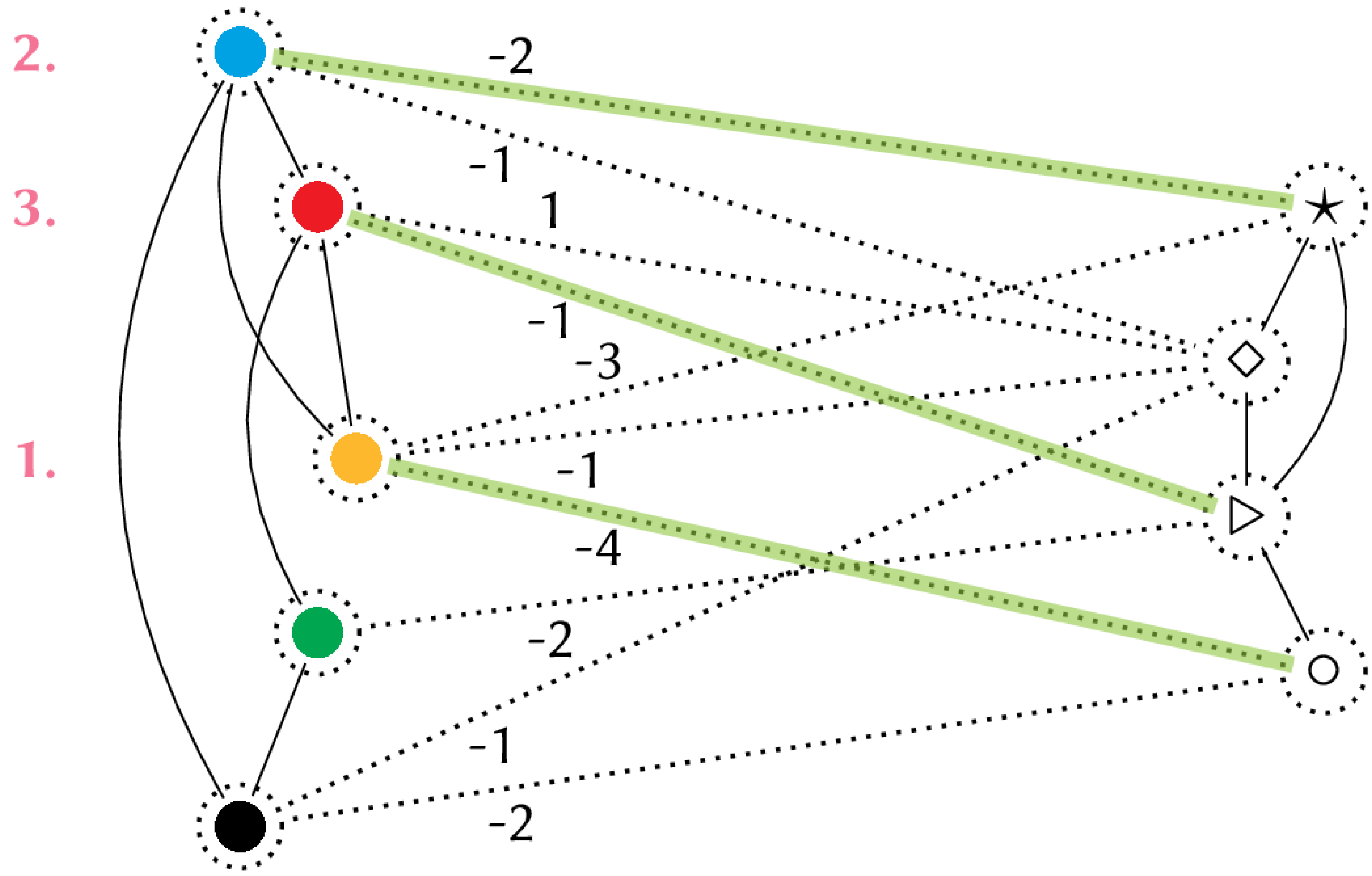
Randomized greedy heuristic



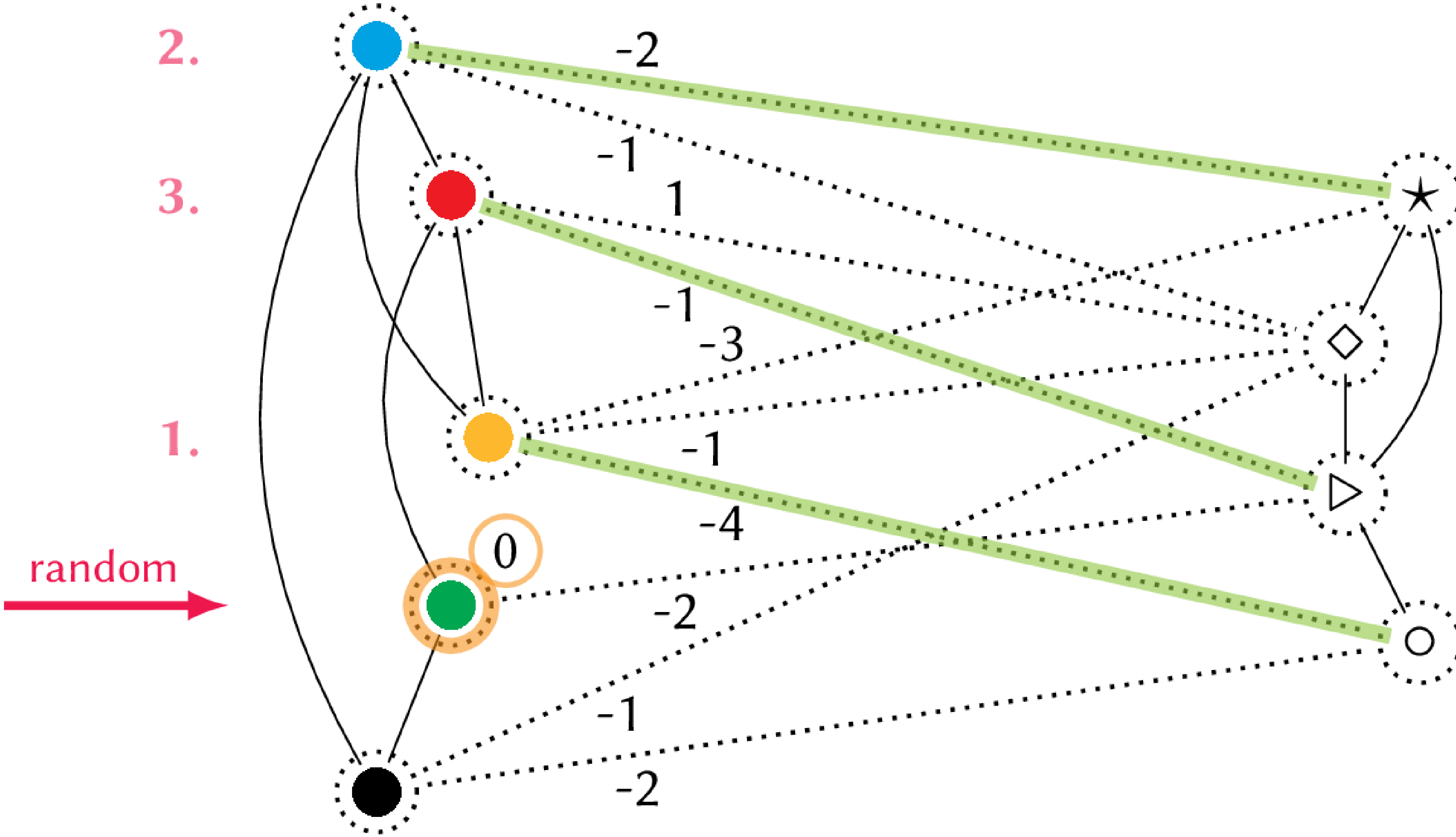
Randomized greedy heuristic



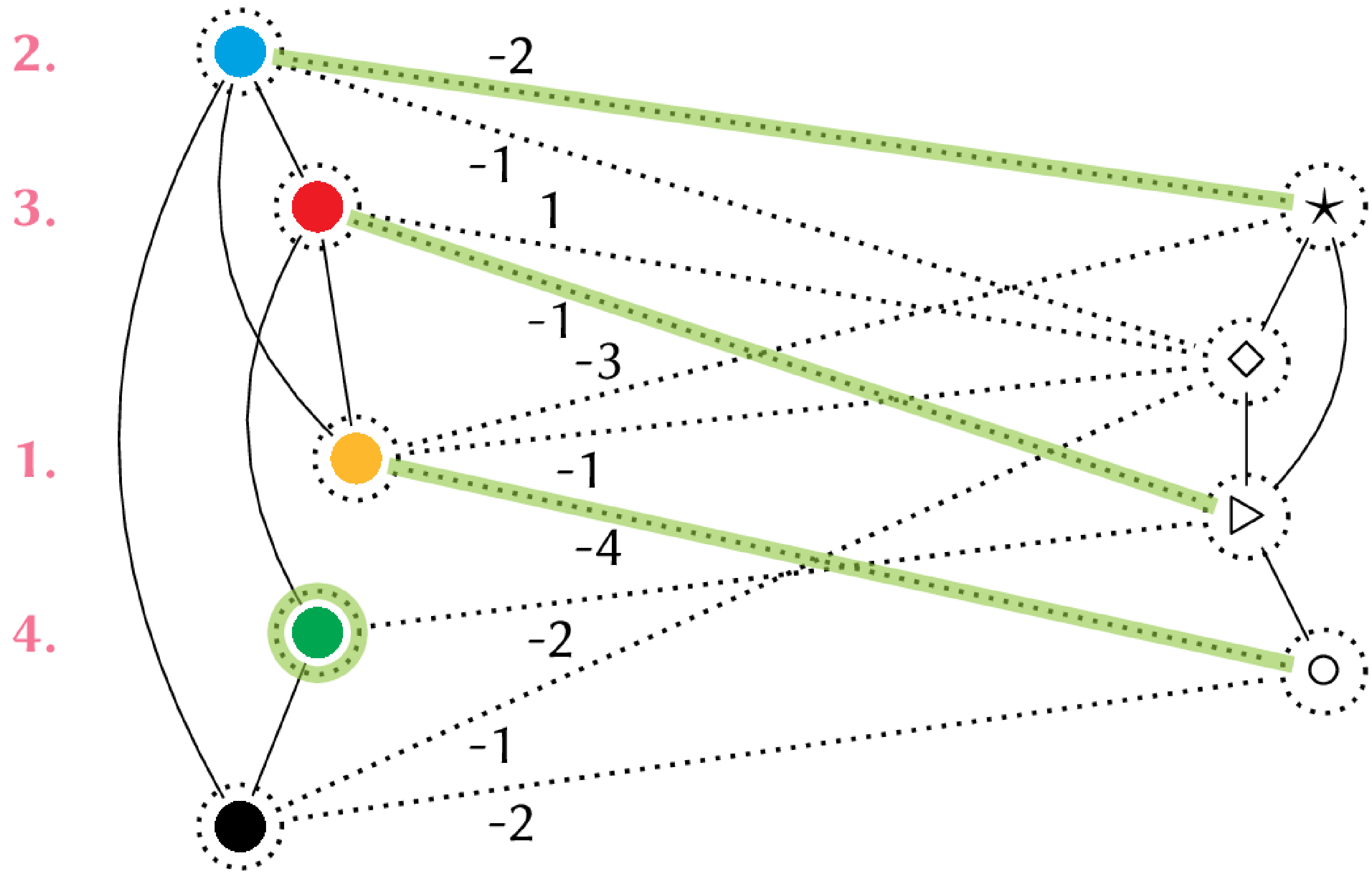
Randomized greedy heuristic



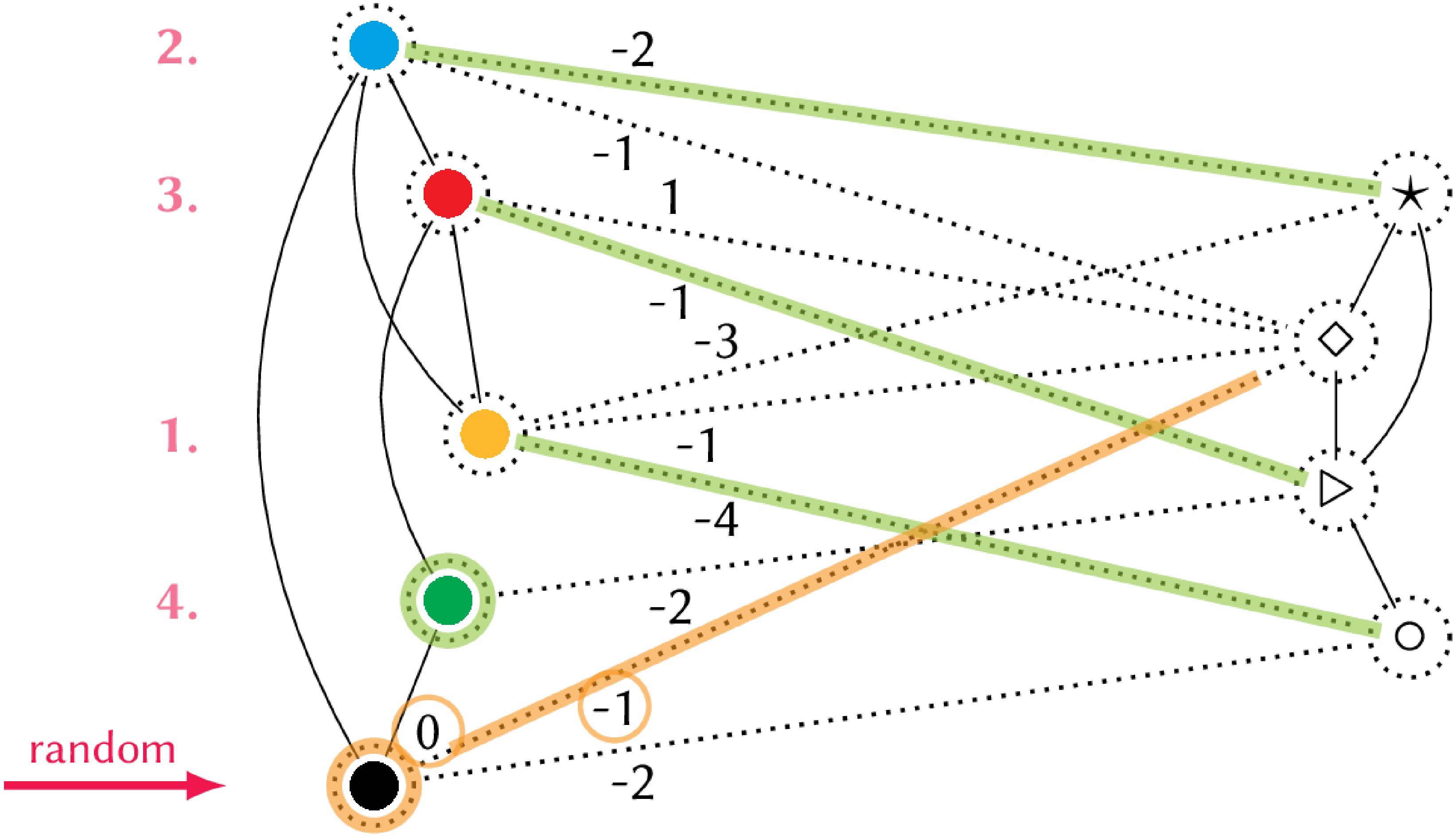
Randomized greedy heuristic



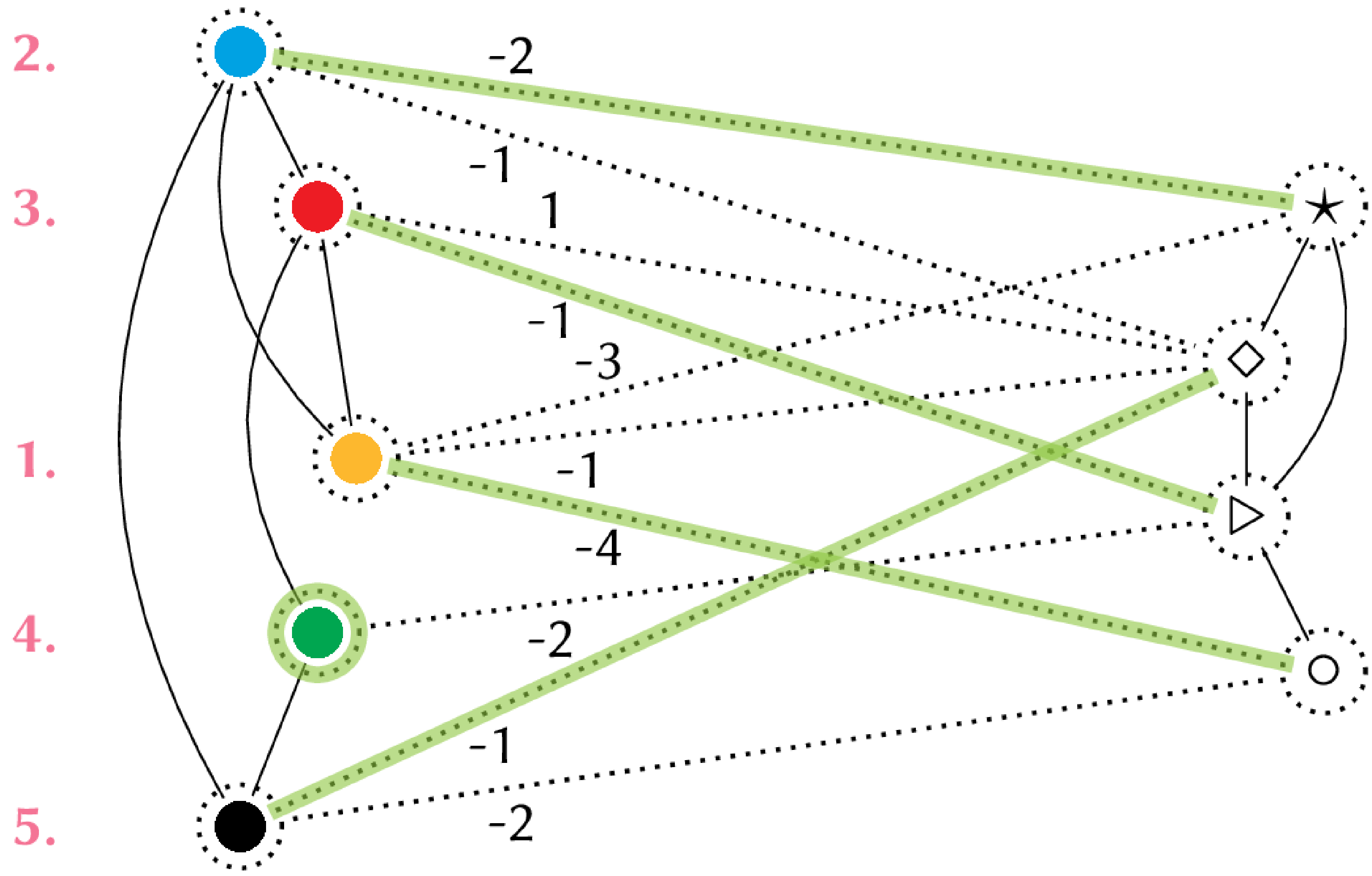
Randomized greedy heuristic



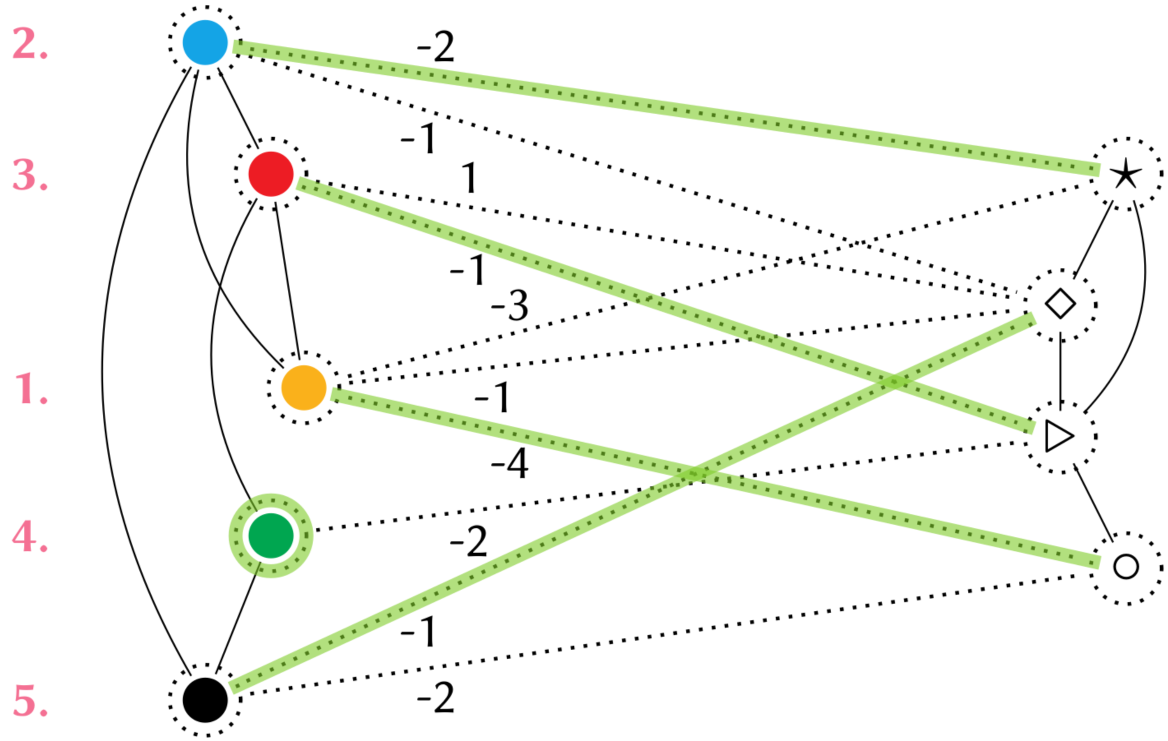
Randomized greedy heuristic

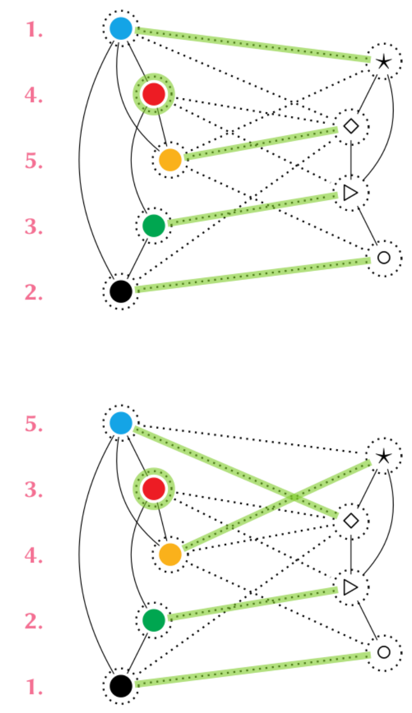


Randomized greedy heuristic



Randomized greedy heuristic





3. Strong primal heuristic: Fusion moves

1. Generate proposal solutions
2. Fuse it with the current one
3. Update the current solution
4. Goto 1.

- *Fusion moves* (fm, fm-bca)

L. Hutschenreiter et al. 2021, Fusion moves for graph matching

Design of the best performing methods

1. Based on integer linear program (**ILP**) **representation**

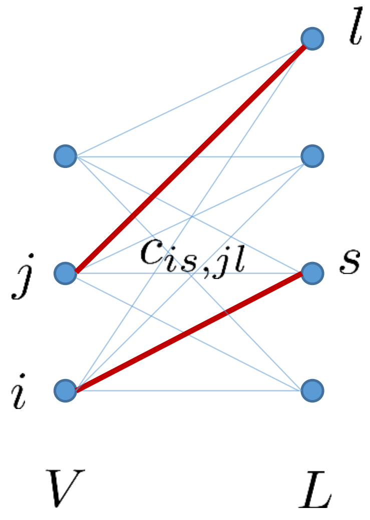
2. **Optimize Lagrange dual** and simplify the cost structure of the problem

3. Run a **strong primal heuristic** on the simplified costs

- *Fusion moves* (fm, fm-bca)

Integer linear program representation

1. ILP representation



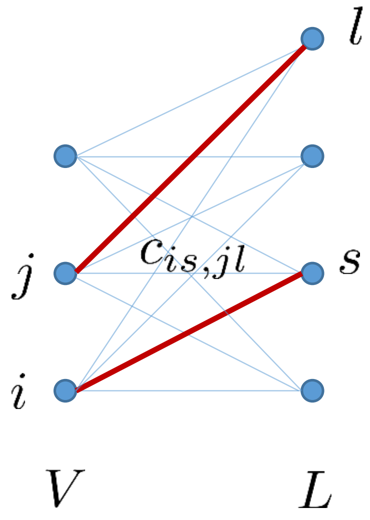
$$\min_{x\in\{0,1\}^{V\times L}} \sum_{i,j\in V}\sum_{s,l\in L} c_{is,jl}x_{is}x_{jl}$$

$$\text{s.t.: } \sum_{s} x_{is} \leq 1 \ \forall i$$

$$\sum_{i} x_{is} \leq 1 \ \forall s$$

Integer **quadratic** program (IQP)

1. ILP representation



$$\min_{x \in \{0,1\}^{V \times L}} \sum_{i,j \in V} \sum_{s,l \in L} c_{is,jl} x_{is} x_{jl}$$

$$\text{s.t.:} \sum_{s} x_{is} \leq 1 \ \forall i$$

$$\sum_{i} x_{is} \leq 1 \ \forall s$$

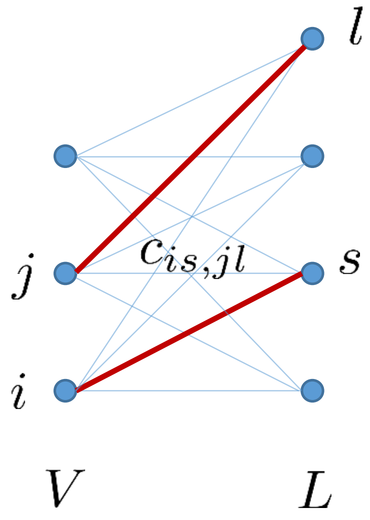
Integer **quadratic** program (IQP)

$$\min_{\substack{x \in \{0,1\}^N \\ Ax \leq b}} \langle c, x \rangle$$

Integer linear program (ILP)

- Well-studied, efficient off-the-shelf solvers (e.g. Gurobi)
- Powerful relaxations/approximative techniques exist

1. ILP representation: Variables lifting



$$\min_{x\in\{0,1\}^N} \sum_{i,j\in V} \sum_{s,l\in L} c_{is,jl} x_{is,jl}$$

$$\text{s.t.:} \sum_s x_{is} \leq 1 \ \forall i$$

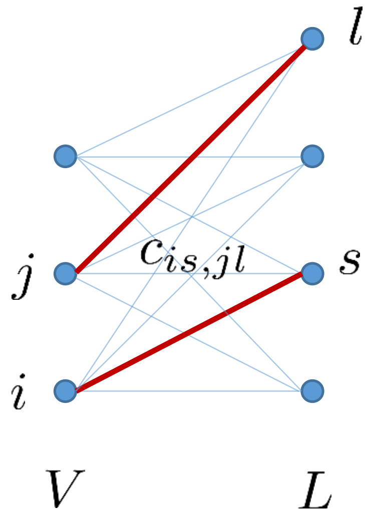
$$\sum_i x_{is} \leq 1 \ \forall s$$

Non-linear constraint: $x_{is,jl} := x_{is} x_{jl}$

$$\min_{\substack{x\in\{0,1\}^N \\ Ax\leq b}} \langle c, x\rangle$$

Integer linear program (ILP)

1. ILP representation: Variables lifting



$$\min_{x \in \{0,1\}^N} \sum_{i,j \in V} \sum_{s,l \in L} c_{is,jl} x_{is,jl}$$

$$\text{s.t.: } \sum_s x_{is} \leq 1 \ \forall i$$

$$\sum_i x_{is} \leq 1 \ \forall s$$

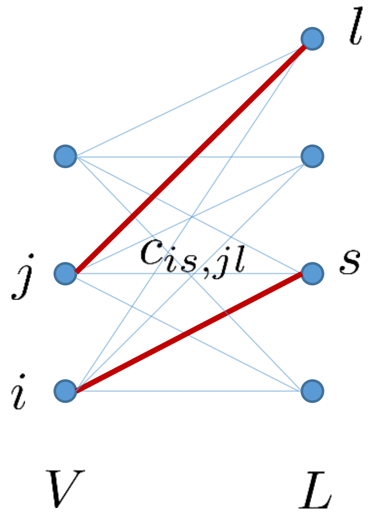
Non-linear constraint: $x_{is,jl} := x_{is} x_{jl}$

$$\min_{\substack{x \in \{0,1\}^N \\ Ax \leq b}} \langle c, x \rangle$$

Integer linear program (ILP)

Lifted variables / problem

1. ILP representation: Variables lifting



$$\min_{x \in \{0,1\}^N} \sum_{i,j \in V} \sum_{s,l \in L} c_{is,jl} x_{is,jl}$$

$$\text{s.t.:} \sum_{s} x_{is} \leq 1 \ \forall i$$

$$\sum_{i} x_{is} \leq 1 \ \forall s$$

Non-linear constraint: $x_{is,jl} := x_{is} x_{jl}$

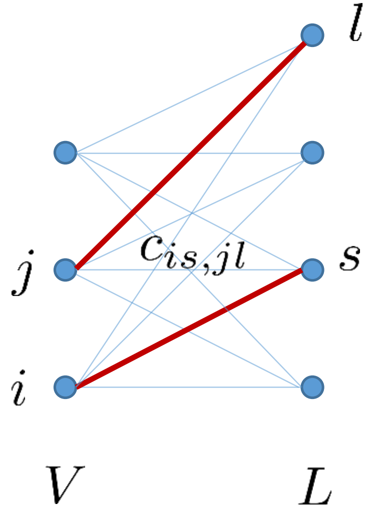
$$\min_{\substack{x \in \{0,1\}^N \\ Ax \leq b}} \langle c, x \rangle$$

Integer linear program (ILP)

Should hold only for binary variables:

x_{is}	1	1	0	0
x_{jl}	1	0	1	0
$x_{is,jl}$	1	0	0	0

1. ILP representation: Variables lifting



$$\min_{x\in\{0,1\}^N} \sum_{i,j\in V} \sum_{s,l\in L} c_{is,jl} x_{is,jl}$$

$$\text{s.t.:} \sum_s x_{is} \leq 1 \ \forall i$$

$$\sum_i x_{is} \leq 1 \ \forall s$$

Non-linear constraint: $x_{is,jl} := x_{is} x_{jl}$

$$\min_{\substack{x\in\{0,1\}^N \\ Ax\leq b}} \langle c, x\rangle$$

Integer linear program (ILP)

Should hold only for binary variables:

x_{is}	1	1	0	0
x_{jl}	1	0	1	0
$x_{is,jl}$	1	0	0	0

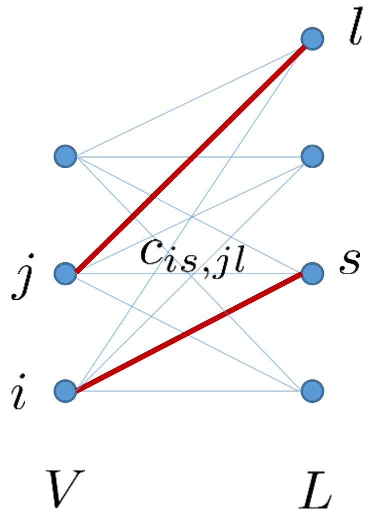
Linear constraints sufficient for binary variables:

$$x_{iji'j'} \leq x_{ij}$$

$$x_{iji'j'} \leq x_{i'j'}$$

$$x_{iji'j'} \geq x_{ij} + x_{i'j'} - 1$$

1. ILP representation: Variables lifting



$$\min_{\substack{x\in\{0,1\}^N \\ Ax\leq b}} \langle c, x\rangle$$

Integer linear
program (ILP)

$$\min_{x\in\{0,1\}^N} \sum_{i,j\in V}\sum_{s,l\in L} c_{is,jl}x_{is,jl}$$

$$\text{s.t.: } \sum_s x_{is} \leq 1 \;\forall i$$

$$\sum_i x_{is} \leq 1 \;\forall s$$

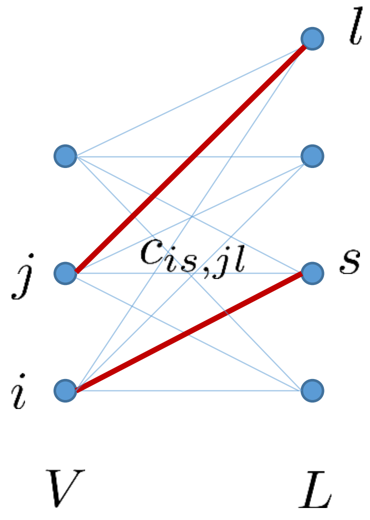
$$x_{iji'j'} \leq x_{ij}$$

$$x_{iji'j'} \leq x_{i'j'}$$

$$x_{iji'j'} \geq x_{ij} + x_{i'j'} - 1$$

Integer linear
program (ILP)

1. ILP representation: Variables lifting



$$\min_{x \in \{0,1\}^N} \sum_{i,j \in V} \sum_{s,l \in L} c_{is,jl} x_{is,jl}$$

$$\text{s.t.:} \sum_s x_{is} \le 1 \ \forall i$$

$$\sum_i x_{is} \le 1 \ \forall s$$

$$x_{iji'j'} \le x_{ij}$$

$$x_{iji'j'} \le x_{i'j'}$$

$$x_{iji'j'} \ge x_{ij} + x_{i'j'} - 1$$

$$\min_{\substack{x \in \{0,1\}^N \\ Ax \le b}} \langle c, x \rangle$$

Integer linear program (ILP)

Integer linear program (ILP)

Different ILP representations/lifted constraints exist

2. Lagrange relaxation: Definition and meaning

2. Lagrange relaxation: Definition and meaning

$$\min_{\substack{x \in \{0,1\}^N \\ Ax=0}} \langle c, x \rangle$$

2. Lagrange relaxation: Definition and meaning

$$\min_{\substack{x\in\{0,1\}^N \\ Ax=0}} \langle c,x\rangle \quad \geq \max_{\lambda} \min_{x\in\{0,1\}^N} \langle c,x\rangle + \langle \lambda, Ax\rangle$$

2. Lagrange relaxation: Definition and meaning

$$\min_{\substack{x\in\{0,1\}^N \\ Ax=0}} \langle c, x\rangle \geq \max_{\lambda} \min_{x\in\{0,1\}^N} \langle c, x\rangle + \langle \lambda, Ax\rangle$$

$$\langle \lambda, Ax\rangle = \langle A^\top \lambda, x\rangle$$

$$= \max_{\lambda} \min_{x\in[0,1]^N} \langle c + A^\top \lambda, x\rangle$$

2. Lagrange relaxation: Definition and meaning

$$\min_{\substack{x\in\{0,1\}^N \\ Ax=0}} \langle c, x\rangle \geq \max_{\lambda} \min_{x\in\{0,1\}^N} \langle c, x\rangle + \langle \lambda, Ax\rangle$$

$$\langle \lambda, Ax\rangle = \langle A^\top \lambda, x\rangle$$

$$= \max_{\lambda} \underbrace{\min_{x\in[0,1]^N} \langle c + A^\top \lambda, x\rangle}_{\text{concave w.r.t. } \lambda}$$

2. Lagrange relaxation: Definition and meaning

$$\min_{\substack{x \in \{0,1\}^N \\ Ax = 0}} \langle c, x \rangle \geq \max_{\lambda} \min_{x \in \{0,1\}^N} \langle c, x \rangle + \langle \lambda, Ax \rangle$$

$$\langle \lambda, Ax \rangle = \langle A^\top \lambda, x \rangle$$

$$= \max_{\lambda} \underbrace{\min_{x \in [0,1]^N} \langle c + A^\top \lambda, x \rangle}_{\text{concave w.r.t. } \lambda}$$

Reduced/reparametrized costs

$$\text{Note: } Ax = 0 \quad \Rightarrow \quad \langle c, x \rangle = \langle \overbrace{c + A^\top \lambda}, x \rangle$$

2. Lagrange relaxation: Definition and meaning

$$\min_{\substack{x\in\{0,1\}^N \\ Ax=0}} \langle c, x\rangle \geq \max_{\lambda} \min_{x\in\{0,1\}^N} \langle c, x\rangle + \langle \lambda, Ax\rangle$$

$$\langle \lambda, Ax\rangle = \langle A^\top \lambda, x\rangle$$

$$= \max_{\lambda} \underbrace{\min_{x\in[0,1]^N} \langle c + A^\top \lambda, x\rangle}_{\text{concave w.r.t. } \lambda}$$

Reduced/reparametrized costs

Note: $Ax = 0 \quad \Rightarrow \quad \langle c, x\rangle = \langle \overbrace{c + A^\top \lambda}, x\rangle$

---

if λ^*-optimal:

$$\min_{x\in[0,1]^N} \langle c + A^\top \lambda^*, x\rangle = \min_{\substack{x\in[0,1]^N \\ Ax=0}} \langle c + A^\top \lambda^*, x\rangle$$

2. Lagrange relaxation: Definition and meaning

$$\min_{\substack{x\in\{0,1\}^N \\ Ax=0}} \langle c, x\rangle \geq \max_{\lambda} \min_{x\in\{0,1\}^N} \langle c, x\rangle + \langle \lambda, Ax\rangle$$

$$= \max_{\lambda} \underbrace{\min_{x\in[0,1]^N} \langle c + A^\top \lambda, x\rangle}_{\text{concave w.r.t. } \lambda}$$

Reduced/reparametrized costs

Note: $Ax = 0 \Rightarrow \langle c, x\rangle = \langle \overbrace{c + A^\top \lambda}, x\rangle$

---

if λ^*-optimal:

$$\min_{x\in[0,1]^N} \langle c + A^\top \lambda^*, x\rangle = \min_{\substack{x\in[0,1]^N \\ Ax=0}} \langle c + A^\top \lambda^*, x\rangle$$

$$(c + A^\top \lambda^*)_i < 0 \Rightarrow x_i^* = 1$$

2. Lagrange relaxation: Definition and meaning

$$\min_{\substack{x\in\{0,1\}^N \\ Ax=0}} \langle c, x\rangle \quad \geq \max_{\lambda} \min_{x\in\{0,1\}^N} \langle c, x\rangle + \langle \lambda, Ax\rangle$$

$$= \max_{\lambda} \underbrace{\min_{x\in[0,1]^N} \langle c + A^\top \lambda, x\rangle}_{\text{concave w.r.t. } \lambda}$$

Reduced/reparametrized costs

$$\text{Note: } Ax = 0 \quad \Rightarrow \quad \langle c, x\rangle = \langle \overbrace{c + A^\top \lambda}, x\rangle$$

---

if λ^*-optimal:

$$\min_{x\in[0,1]^N} \langle c + A^\top \lambda^*, x\rangle = \min_{\substack{x\in[0,1]^N \\ Ax=0}} \langle c + A^\top \lambda^*, x\rangle$$

$$(c + A^\top \lambda^*)_i < 0 \Rightarrow x_i^* = 1$$

$$(c + A^\top \lambda^*)_i > 0 \Rightarrow x_i^* = 0$$

2. Lagrange relaxation: Definition and meaning

$$\min_{\substack{x \in \{0,1\}^N \\ Ax=0}} \langle c, x \rangle \geq \max_{\lambda} \min_{x \in \{0,1\}^N} \langle c, x \rangle + \langle \lambda, Ax \rangle$$

$$= \max_{\lambda} \underbrace{\min_{x \in [0,1]^N} \langle c + A^\top \lambda, x \rangle}_{\text{concave w.r.t. } \lambda}$$

Reduced/reparametrized costs

Note: $Ax = 0 \Rightarrow \langle c, x \rangle = \langle \overbrace{c + A^\top \lambda}, x \rangle$

---

if λ^*-optimal:

$$\min_{x \in [0,1]^N} \langle c + A^\top \lambda^*, x \rangle = \min_{\substack{x \in [0,1]^N \\ Ax=0}} \langle c + A^\top \lambda^*, x \rangle$$

$$(c + A^\top \lambda^*)_i < 0 \Rightarrow x_i^* = 1$$
$$(c + A^\top \lambda^*)_i > 0 \Rightarrow x_i^* = 0$$
$$(c + A^\top \lambda^*)_i = 0 \Rightarrow x_i^* \in [0, 1]$$

2. Lagrange relaxation: Definition and meaning

$$\min_{\substack{x \in \{0,1\}^N \\ Ax=0}} \langle c, x \rangle \geq \max_{\lambda} \min_{x \in \{0,1\}^N} \langle c, x \rangle + \langle \lambda, Ax \rangle$$

$$= \max_{\lambda} \underbrace{\min_{x \in [0,1]^N} \langle c + A^\top \lambda, x \rangle}_{\text{concave w.r.t. } \lambda}$$

Reduced/reparametrized costs

$$\text{Note: } Ax = 0 \quad \Rightarrow \quad \langle c, x \rangle = \overbrace{\langle c + A^\top \lambda, x \rangle}$$

---

if λ^*-optimal:

$$\min_{x \in [0,1]^N} \langle c + A^\top \lambda^*, x \rangle = \min_{\substack{x \in [0,1]^N \\ Ax=0}} \langle c + A^\top \lambda^*, x \rangle$$

$$(c + A^\top \lambda^*)_i < 0 \Rightarrow x_i^* = 1$$

$$(c + A^\top \lambda^*)_i > 0 \Rightarrow x_i^* = 0 \qquad \Longrightarrow \qquad Ax^* = 0$$

$$(c + A^\top \lambda^*)_i = 0 \Rightarrow x_i^* \in [0, 1]$$

2. Lagrange relaxation: Definition and meaning

$$\min_{\substack{x \in \{0,1\}^N \\ Ax=0}} \langle c, x\rangle \geq \max_{\lambda} \min_{x\in\{0,1\}^N} \langle c, x\rangle + \langle \lambda, Ax\rangle$$

$$= \max_{\lambda} \underbrace{\min_{x\in[0,1]^N} \langle c + A^\top \lambda, x\rangle}_{\text{concave w.r.t. } \lambda}$$

Reduced/reparametrized costs

$$\text{Note: } Ax = 0 \quad \Rightarrow \quad \langle c, x\rangle = \langle \overbrace{c + A^\top \lambda}, x\rangle$$

---

if λ^*-optimal:

$$\min_{x\in[0,1]^N} \langle c + A^\top \lambda^*, x\rangle = \min_{\substack{x\in[0,1]^N \\ Ax=0}} \langle c + A^\top \lambda^*, x\rangle$$

$$(c + A^\top \lambda^*)_i < 0 \Rightarrow x_i^* = 1$$
$$(c + A^\top \lambda^*)_i > 0 \Rightarrow x_i^* = 0 \qquad \Longrightarrow \qquad Ax^* = 0$$
$$(c + A^\top \lambda^*)_i = 0 \Rightarrow x_i^* \in [0, 1]$$

---

if $x^* \in \{0, 1\}^N$:

2. Lagrange relaxation: Definition and meaning

$$\min_{\substack{x\in\{0,1\}^N\\ Ax=0}} \langle c, x\rangle \geq \max_{\lambda} \min_{x\in\{0,1\}^N} \langle c, x\rangle + \langle \lambda, Ax\rangle$$

$$= \max_{\lambda} \underbrace{\min_{x\in[0,1]^N} \langle c + A^\top \lambda, x\rangle}_{\text{concave w.r.t. } \lambda}$$

Reduced/reparametrized costs

$$\text{Note: } Ax = 0 \quad \Rightarrow \quad \langle c, x\rangle = \langle \overbrace{c + A^\top \lambda}, x\rangle$$

---

if λ^*-optimal:

$$\min_{x\in[0,1]^N} \langle c + A^\top \lambda^*, x\rangle = \min_{\substack{x\in[0,1]^N\\ Ax=0}} \langle c + A^\top \lambda^*, x\rangle$$

$$(c + A^\top \lambda^*)_i < 0 \Rightarrow x_i^* = 1$$

$$(c + A^\top \lambda^*)_i > 0 \Rightarrow x_i^* = 0 \qquad \Longrightarrow \qquad Ax^* = 0$$

$$(c + A^\top \lambda^*)_i = 0 \Rightarrow x_i^* \in [0, 1]$$

$$\text{if } x^* \in \{0,1\}^N: \quad x^* \in \arg \min_{\substack{x\in\{0,1\}^N\\ Ax=0}} \langle c + A^\top \lambda^*, x\rangle$$

2. Lagrange relaxation: Definition and meaning

$$\min_{\substack{x\in\{0,1\}^N\\ Ax=0}} \langle c, x\rangle \geq \max_{\lambda} \min_{x\in\{0,1\}^N} \langle c, x\rangle + \langle \lambda, Ax\rangle$$

$$= \max_{\lambda} \underbrace{\min_{x\in[0,1]^N} \langle c + A^\top\lambda, x\rangle}_{\text{concave w.r.t. } \lambda}$$

Reduced/reparametrized costs

Note: $Ax = 0 \Rightarrow \langle c, x\rangle = \langle \overbrace{c + A^\top\lambda}, x\rangle$

---

if λ^*-optimal:

$$\min_{x\in[0,1]^N} \langle c + A^\top\lambda^*, x\rangle = \min_{\substack{x\in[0,1]^N\\ Ax=0}} \langle c + A^\top\lambda^*, x\rangle$$

$(c + A^\top\lambda^*)_i < 0 \Rightarrow x_i^* = 1$

$(c + A^\top\lambda^*)_i > 0 \Rightarrow x_i^* = 0 \quad \Longrightarrow \quad Ax^* = 0$

$(c + A^\top\lambda^*)_i = 0 \Rightarrow x_i^* \in [0, 1]$

---

if $x^* \in \{0,1\}^N$: $\quad x^* \in \arg \min_{\substack{x\in\{0,1\}^N\\ Ax=0}} \langle c + A^\top\lambda^*, x\rangle = \arg \min_{x\in\{0,1\}^N} \langle c + A^\top\lambda^*, x\rangle$ Simple!

2. Lagrange relaxation: Definition and meaning

$$\min_{\substack{x\in\{0,1\}^N\\ Ax=0}} \langle c, x\rangle \geq \max_{\lambda} \min_{x\in\{0,1\}^N} \langle c, x\rangle + \langle \lambda, Ax\rangle$$

$$= \max_{\lambda} \underbrace{\min_{x\in[0,1]^N} \langle c + A^\top \lambda, x\rangle}_{\text{concave w.r.t. } \lambda}$$

Reduced/reparametrized costs

$$\text{Note: } Ax = 0 \quad \Rightarrow \quad \langle c, x\rangle = \overbrace{\langle c + A^\top \lambda, x\rangle}$$

---

if λ^*-optimal:

$$\min_{x\in[0,1]^N} \langle c + A^\top \lambda^*, x\rangle = \min_{\substack{x\in[0,1]^N\\ Ax=0}} \langle c + A^\top \lambda^*, x\rangle$$

$$(c + A^\top \lambda^*)_i < 0 \Rightarrow x_i^* = 1$$

$$(c + A^\top \lambda^*)_i > 0 \Rightarrow x_i^* = 0 \qquad \Longrightarrow \qquad Ax^* = 0$$

$$(c + A^\top \lambda^*)_i = 0 \Rightarrow x_i^* \in [0, 1]$$

---

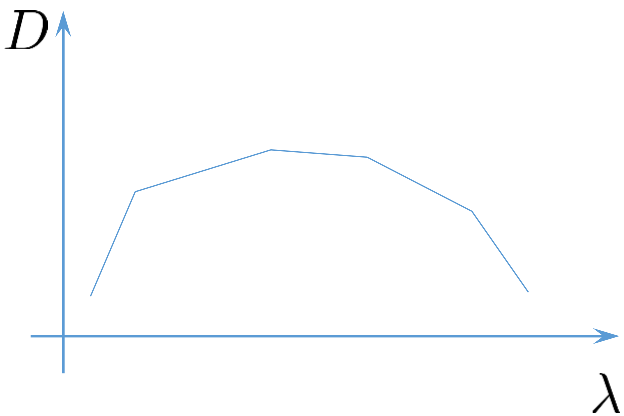
if $x^* \in \{0,1\}^N$: $\quad x^* \in \arg \min_{\substack{x\in\{0,1\}^N\\ Ax=0}} \langle c + A^\top \lambda^*, x\rangle = \arg \min_{x\in\{0,1\}^N} \langle c + A^\top \lambda^*, x\rangle$ Simple!

Reduced costs simplify the problem!

2. Lagrange relaxation: How to solve

$$\min_{\substack{x\in\{0,1\}^N \\ Ax=0}} \langle c, x\rangle \geq \max_{\lambda} \min_{x\in\{0,1\}^N} \langle c, x\rangle + \langle \lambda, Ax\rangle$$

$$= \max_{\lambda} \underbrace{\min_{x\in[0,1]^N} \langle c + A^\top \lambda, x\rangle}_{D,\ \text{concave w.r.t. } \lambda}$$

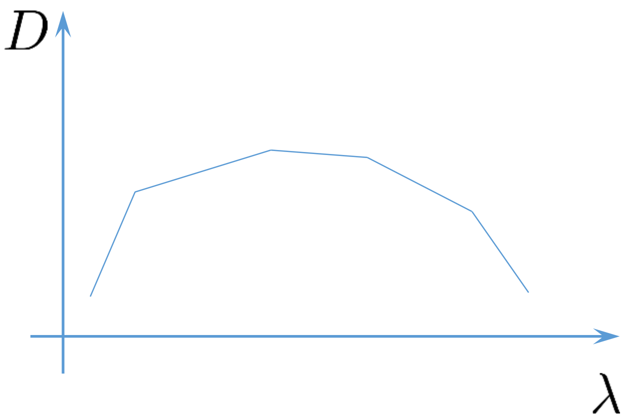


concave, piece-wise linear, large-scale

2. Lagrange relaxation: How to solve

$$\min_{\substack{x\in\{0,1\}^N \\ Ax=0}} \langle c, x\rangle \geq \max_{\lambda} \min_{x\in\{0,1\}^N} \langle c, x\rangle + \langle \lambda, Ax\rangle$$

$$= \max_{\lambda} \underbrace{\min_{x\in[0,1]^N} \langle c + A^\top \lambda, x\rangle}_{\text{D, concave}}$$



concave, piece-wise linear, large-scale

Large-scale => first-order methods:

- **Sub-gradient** [Shor 197X]
  - dd-lsX *Dual decomposition* [Torresani et al., 2013]
- Bundle [Kiwiel, Lemarechal 198X]
- Proximal [e.g. Parikh, Boyd 2013]
- **Block-coordinate ascent (BCA)** [196X]
  - fm-bca *Fusion Moves + BCA* [Hutschenreiter et al., 2021]
  - mp-XXX *Message passing* [Swoboda et al. 2017]
- Smoothing + BCA/accelerated gradient [Nesterov 200X]

Putting all together

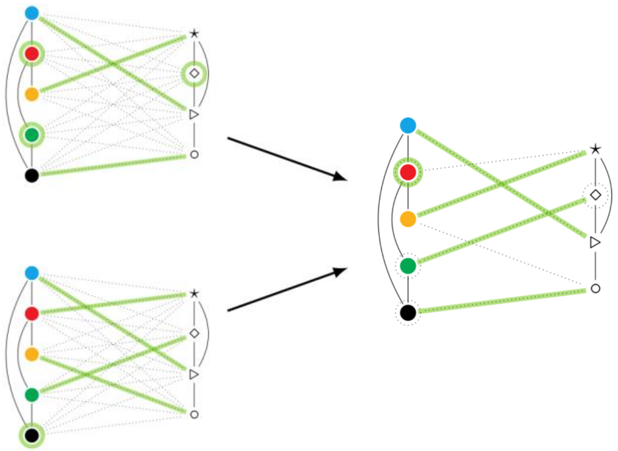
1. Consider **ILP representation**

$$x_{is,jl} := x_{is} x_{jl}$$

2. Optimize Lagrange dual and
simplify the cost structure of the problem

$$\min_{\substack{x \in \{0,1\}^N \\ Ax=0}} \langle c + A^\top \lambda^*, x \rangle$$

3. Run a **strong primal heuristic** on the simplified costs



What expects us in future?