Fusion Moves for Graph Matching

Lisa Hutschenreiter

Stefan Haller

Lorenz Feineis

Carsten Rother

Dagmar Kainmüller

Bogdan Savchynskyy

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

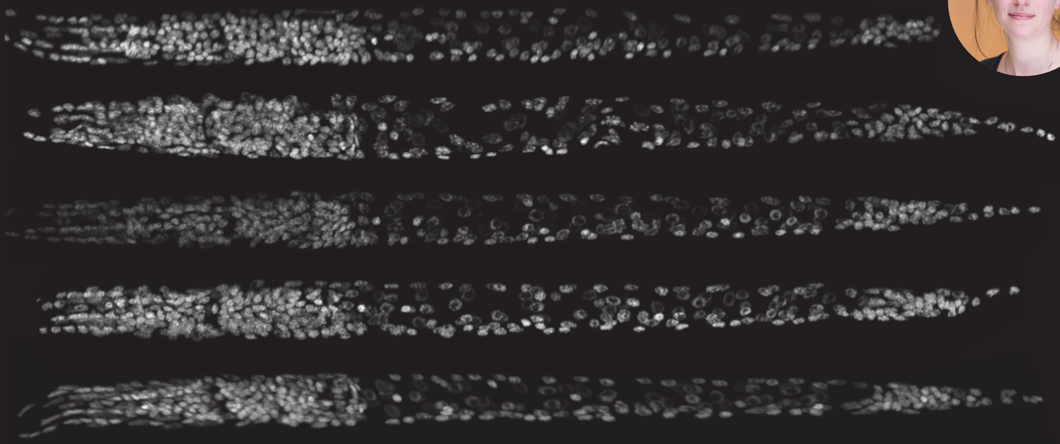
Computer Vision and Learning Lab

HiDSS 4HEALTH HELMHOLTZ INFORMATION & DATA SCIENCE SCHOOL FOR HEALTH

erc European Research Council Established by the European Commission

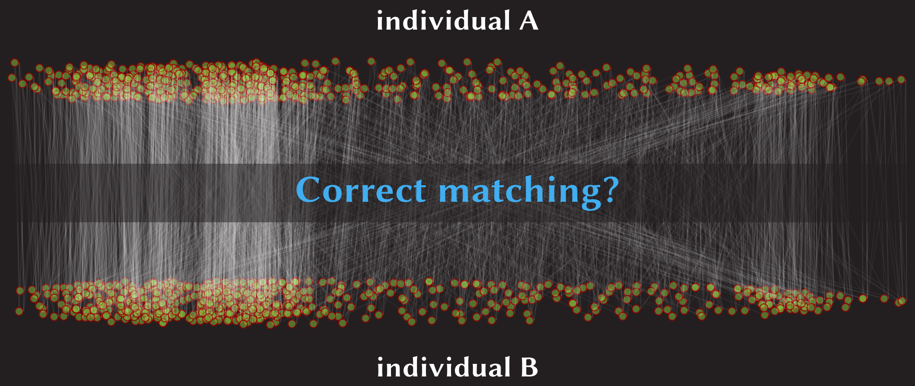
V. Ljosa, K. Sokolnicki, A. Carpenter: **Annotated high-throughput microscopy image sets for validation**. *Nature methods, 2012*

3d images of *Caenorhabtidis elegans*

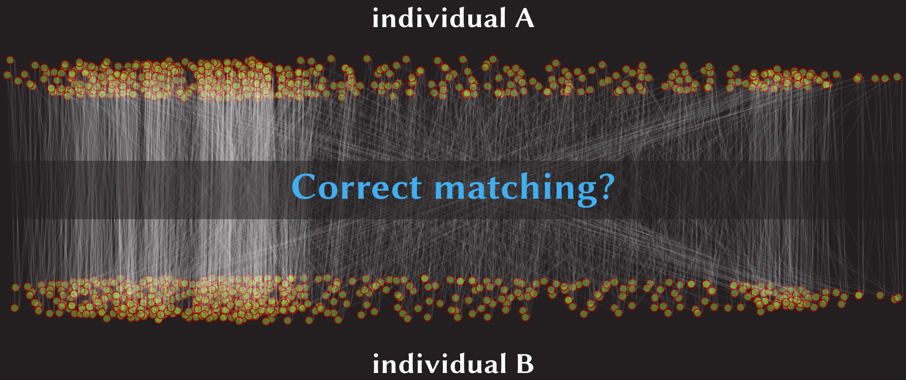


F. Long, H. Peng, X. Liu, S. Kim, and E. Myers: **A 3d digital atlas of C. elegans and its application to single-cell analyses**. *Nature methods, 2009*

C. elegans matching



C. elegans matching



15 minutes per matching

about **300 individuals** → about **45 000 matching problems**

1.3 years total computation time

Graph matching

graph 1 with nodes

$\mathcal{V}_1 = \{ \bullet, \bullet, \bullet, \bullet, \bullet \}$

graph 2 with nodes

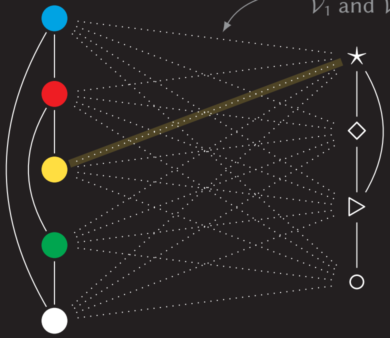
$\mathcal{V}_2 = \{ \star, \diamond, \triangleright, \circ \}$

Graph matching

all possible assignments between $\mathcal{V}_1$ and $\mathcal{V}_2$, e. g. ● $\mapsto$ $\star$

graph 1 with nodes

$\mathcal{V}_1 = \{$ ●, ●, ●, ●, ● $\}$



graph 2 with nodes

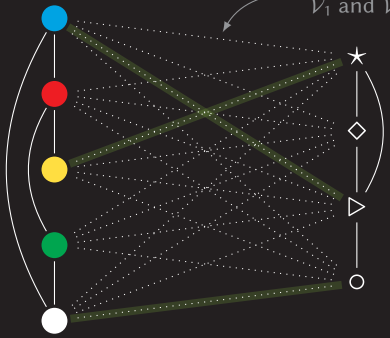
$\mathcal{V}_2 = \{ \star, \diamond, \triangleright, \circ \}$

Graph matching

all possible assignments between $\mathcal{V}_1$ and $\mathcal{V}_2$, e. g. 🟡 $\mapsto \star$

graph 1 with nodes

$\mathcal{V}_1 = \{$ 🔵, 🔴, 🟡, 🟢, ⚪ $\}$



graph 2 with nodes

$\mathcal{V}_2 = \{ \star, \diamond, \triangleright, \circ \}$

every node is assigned to **at most one** other node, e. g.

$x = \{$ 🔵 $\mapsto \triangleright$, 🟡 $\mapsto \star$, ⚪ $\mapsto \circ \}$

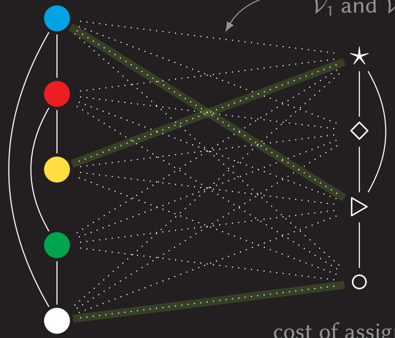
$$\min_{\text{feasible } x} \left(\sum_{\text{ass} \in x} \theta(\text{ass}) + \sum_{\substack{\text{ass}_1, \text{ass}_2 \in x \\ \text{ass}_1 \neq \text{ass}_2}} \theta(\text{ass}_1, \text{ass}_2) \right)$$

Graph matching

all possible assignments between $\mathcal{V}_1$ and $\mathcal{V}_2$, e. g. $\bullet \mapsto \star$

graph 1 with nodes

$$\mathcal{V}_1 = \{\, \bullet, \bullet, \bullet, \bullet, \bullet \,\}$$



graph 2 with nodes

$$\mathcal{V}_2 = \{\, \star, \diamond, \triangleright, \circ \,\}$$

every node is assigned to **at most one** other node, e. g.

$$x = \{\, \bullet \mapsto \triangleright, \bullet \mapsto \star, \bullet \mapsto \circ \,\}$$

cost of assignments, e. g. $\theta(\bullet \mapsto \star) = -2$

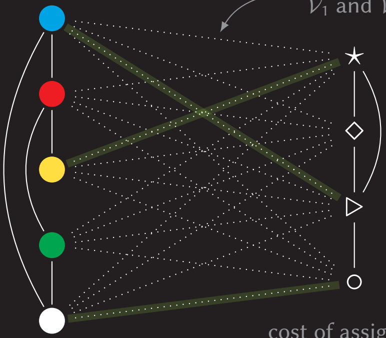
$$\min_{\text{feasible } x} \left(\sum_{\text{ass} \in x} \theta(\text{ass}) + \sum_{\substack{\text{ass}_1, \text{ass}_2 \in x \\ \text{ass}_1 \neq \text{ass}_2}} \theta(\text{ass}_1, \text{ass}_2) \right)$$

Graph matching

all possible assignments between $\mathcal{V}_1$ and $\mathcal{V}_2$, e. g. 🟡 $\mapsto \star$

graph 1 with nodes

$\mathcal{V}_1 = \{$ 🔵, 🔴, 🟡, 🟢, ⚪ $\}$



graph 2 with nodes

$\mathcal{V}_2 = \{ \star, \diamond, \triangleright, \circ \}$

every node is assigned to **at most one** other node, e. g.

$x = \{$ 🔵 $\mapsto \triangleright$, 🟡 $\mapsto \star$, ⚪ $\mapsto \circ \}$

cost of assignments, e. g.

$\theta($ 🟡 $\mapsto \star) = -2$

$$\min_{\text{feasible } x} \left(\sum_{\text{ass} \in x} \theta(\text{ass}) + \sum_{\substack{\text{ass}_1, \text{ass}_2 \in x \\ \text{ass}_1 \neq \text{ass}_2}} \theta(\text{ass}_1, \text{ass}_2) \right)$$

cost of assignment pairs, e. g.

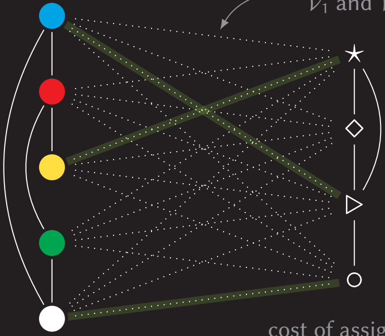
$\theta($ 🔵 $\mapsto \triangleright$, ⚪ $\mapsto \circ) = -1$

Graph matching

graph 1 with nodes

$\mathcal{V}_1 = \{ ●, ●, ●, ●, ● \}$

all possible assignments between $\mathcal{V}_1$ and $\mathcal{V}_2$, e. g. $● \mapsto \star$



graph 2 with nodes

$\mathcal{V}_2 = \{ \star, \diamond, \triangleright, \circ \}$

every node is assigned to **at most one** other node, e. g.

$x = \{ ● \mapsto \triangleright, ● \mapsto \star, ● \mapsto \circ \}$

cost of assignments, e. g.

$\theta(● \mapsto \star) = -2$

$$\min_{\text{feasible } x} \left(\sum_{ass \in x} \theta(ass) + \sum_{\substack{ass_1, ass_2 \in x \\ ass_1 \neq ass_2}} \theta(ass_1, ass_2) \right)$$

energy $E(\theta, x)$

cost of assignment pairs, e. g.

$\theta(● \mapsto \triangleright, ● \mapsto \circ) = -1$

Graph matching in Computer Vision

Primal heuristics

treat graph matching as
continuous quadratic program

- integer projected fixed point method
  Leordeanu *et al.*: **An integer projected fixed point method for graph matching and MAP inference**. *NeurIPS '09*
- graduated assignment
  Gold, Rangarajan: **A graduated assignment algorithm for graph matching**. *PAMI '96*
- spectral relaxation-based
  Leordeanu, Hebert: **A spectral technique for correspondence problems using pairwise constraints**. *ICCV '05*
- path-following procedures
  Bernard *et al.*: **DS*: Tighter lifting-free convex relaxations for quadratic matching problems**. *CVPR '18*

⋮

Graph matching in Computer Vision

Primal heuristics	**Dual methods**
treat graph matching as **continuous** quadratic program	simplify cost structure $\theta \to \hat{\theta}$ by reparametrization, $E(\theta, x) = E(\hat{\theta}, x)$

Primal heuristics

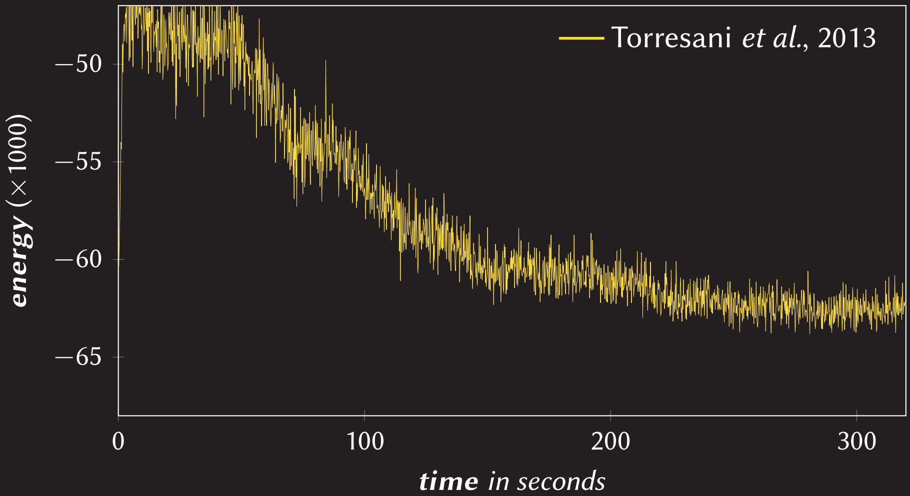
- integer projected fixed point method
  Leordeanu *et al.*: **An integer projected fixed point method for graph matching and MAP inference**. *NeurIPS '09*
- graduated assignment
  Gold, Rangarajan: **A graduated assignment algorithm for graph matching**. *PAMI '96*
- spectral relaxation-based
  Leordeanu, Hebert: **A spectral technique for correspondence problems using pairwise constraints**. *ICCV '05*
- path-following procedures
  Bernard *et al.*: **DS*: Tighter lifting-free convex relaxations for quadratic matching problems**. *CVPR '18*

⋮

Dual methods

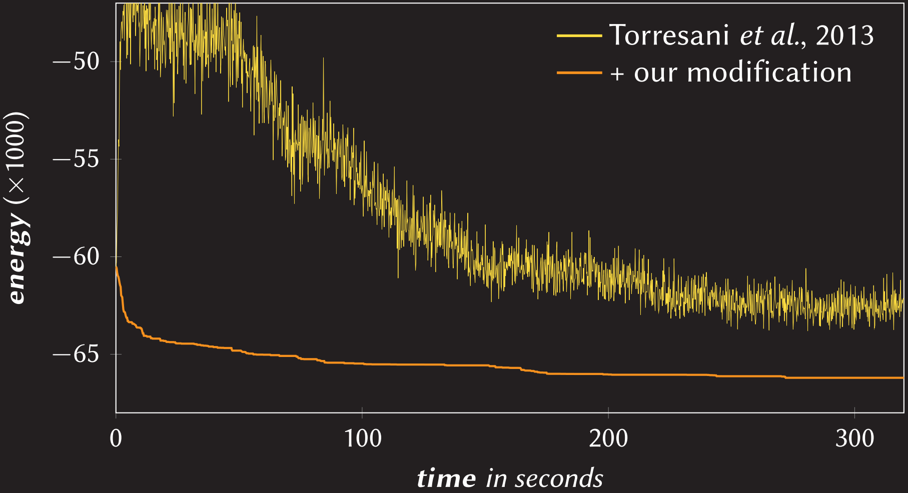
- subgradient-based
  Torresani *et al.*: **A dual decomposition approach to feature correspondence**. *PAMI '13*
- block-coordinate ascent
  Zhang *et al.*: **Pairwise matching through max-weight bipartite belief propagation**. *CVPR '16*
  Swoboda *et al.*: **A study of Lagrangean decompositions and dual ascent solvers for graph matching**. *CVPR '17*

Dual decomposition approach for *C. elegans* matching



L. Torresani, V. Kolmogorov, C. Rother: **A dual decomposition approach to feature correspondence**. *PAMI '13*

Dual decomposition approach for *C. elegans* matching



L. Torresani, V. Kolmogorov, C. Rother: **A dual decomposition approach to feature correspondence**. *PAMI '13*

What we propose for graph matching

graph matching solver

dual updates: $\theta \rightarrow \hat{\theta}$

- by block-coordinate ascent, similar to Swoboda *et al.*, 2017
- modified for speed

+ proposal generation method

+ fusion moves

- inspired by Lempitsky *et al.*, 2010

P. Swoboda, C. Rother, H. Abu Alhaija, D. Kainmüller, B. Savchynskyy: **A study of Lagrangean decompositions and dual ascent solvers for graph matching**. *CVPR '17*

V. Lempitsky, C. Rother, S. Roth, A. Blake: **Fusion moves for Markov random field optimization**. *PAMI '10*

What we propose for graph matching

graph matching solver

dual updates: $\theta \to \hat{\theta}$

- by block-coordinate ascent, similar to Swoboda *et al.*, 2017
- modified for speed

+ proposal generation method

+ fusion moves

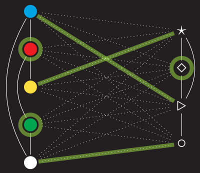
- inspired by Lempitsky *et al.*, 2010

P. Swoboda, C. Rother, H. Abu Alhaija, D. Kainmüller, B. Savchynskyy: **A study of Lagrangean decompositions and dual ascent solvers for graph matching**. *CVPR '17*

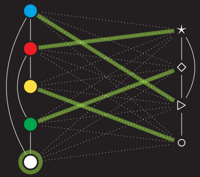
V. Lempitsky, C. Rother, S. Roth, A. Blake: **Fusion moves for Markov random field optimization**. *PAMI '10*

Fusion moves

proposed
matching x_1



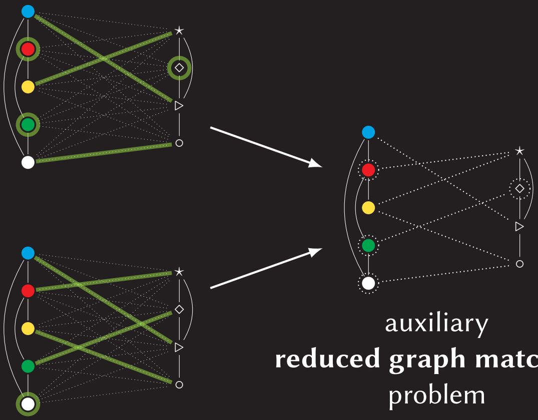
proposed
matching x_2



Fusion moves

proposed
matching x_1

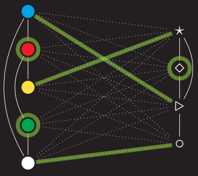
proposed
matching x_2



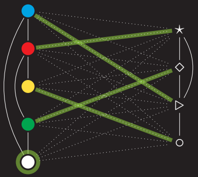
auxiliary
reduced graph matching
problem

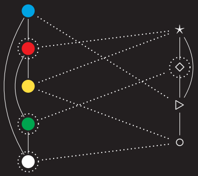
Fusion moves

proposed matching x_1



proposed matching x_2

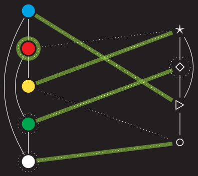




auxiliary **reduced graph matching** problem

QPBO

Rother *et al.*, 2007



fused proposal x_{fuse}

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

C. Rother, V. Kolmogorov, V. Lempitsky, M. Szummer: **Optimizing binary MRFs via extended roof duality**. *CVPR '07*

How to generate proposals for fusion moves?

How to generate proposals for fusion moves?

Desired properties:

- **high quality**

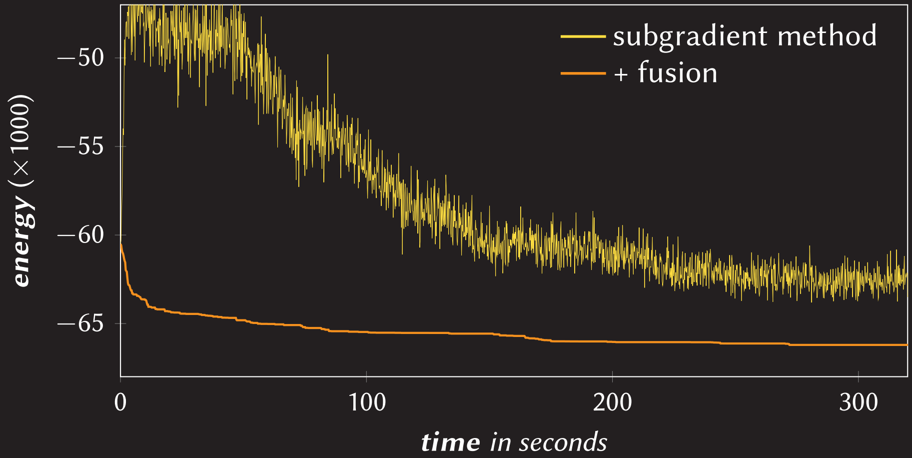
How to generate proposals for fusion moves?

Desired properties:

- **high quality**
- **diverse**

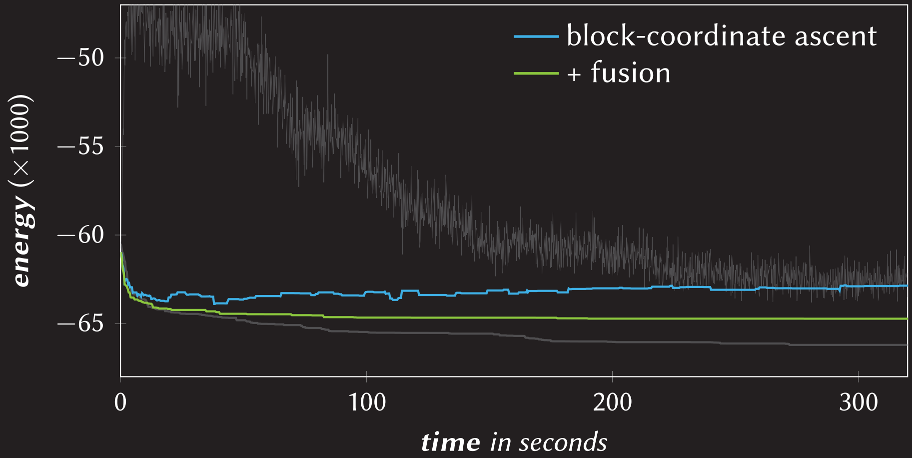
Methods for generation

Variant 1: dual solver + primitive primal heuristic + **fusion moves**



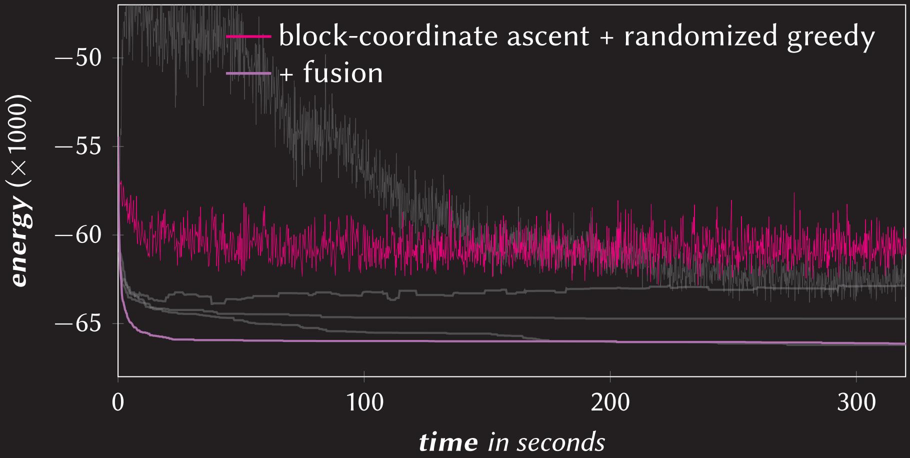
Methods for generation

Variant 1: dual solver + primitive primal heuristic + **fusion moves**

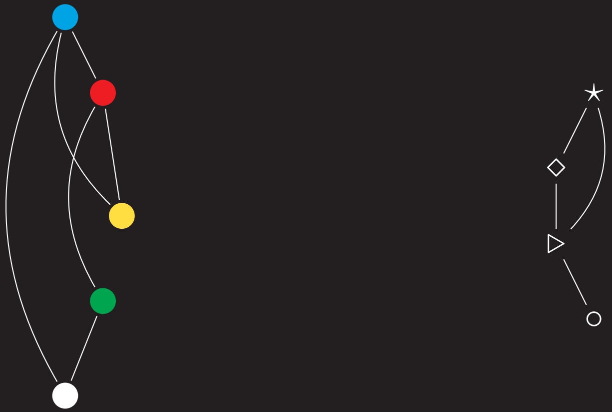


Methods for generation

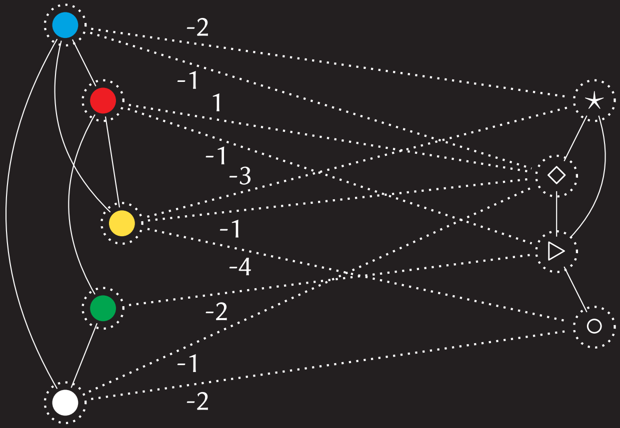
Variant 2: dual solver + **randomized greedy** + **fusion moves**



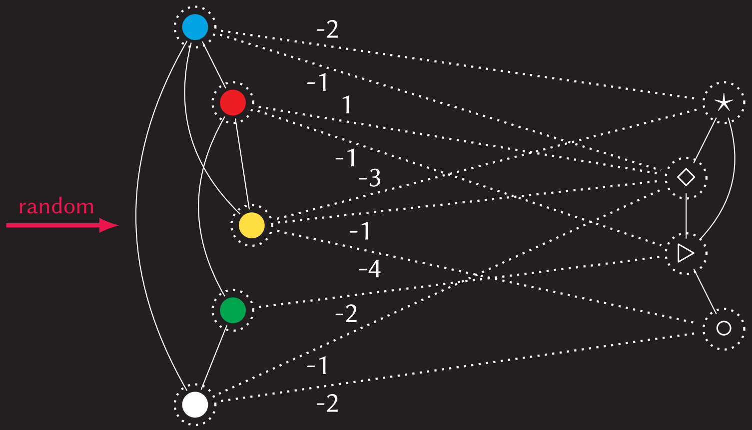
Randomized greedy heuristic



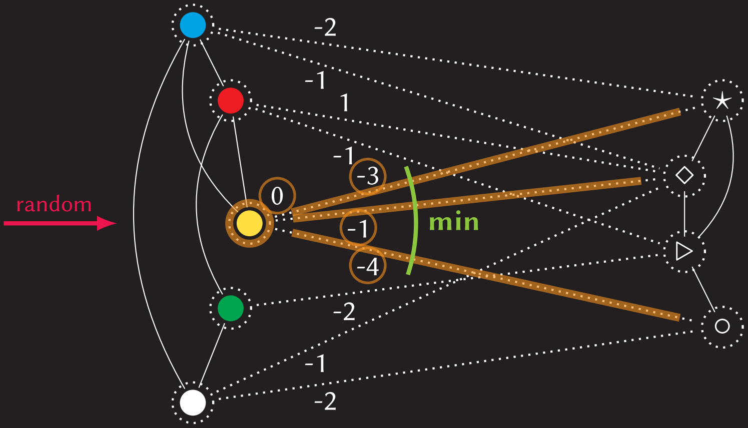
Randomized greedy heuristic



Randomized greedy heuristic

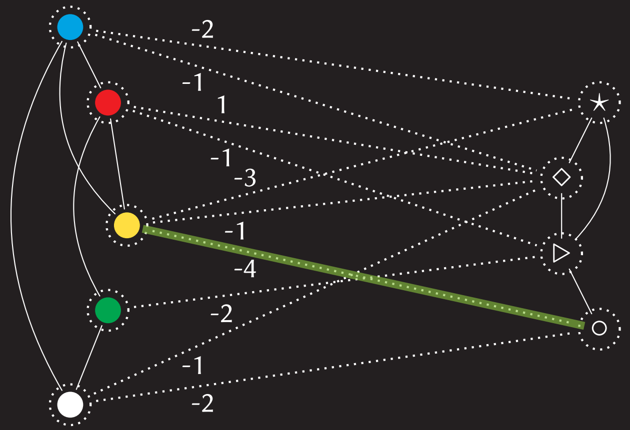


Randomized greedy heuristic

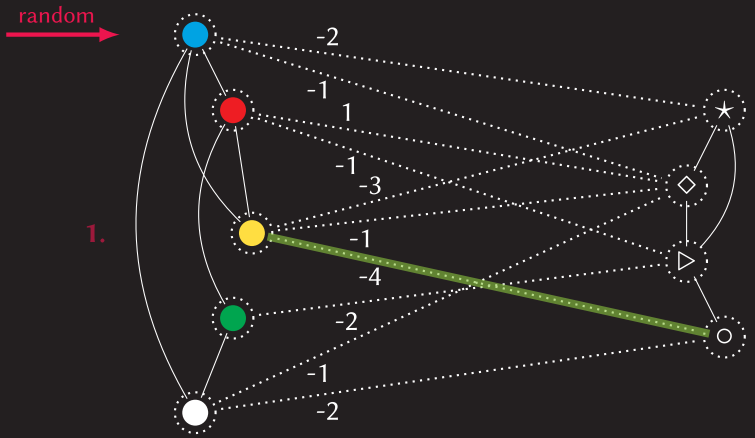


Randomized greedy heuristic

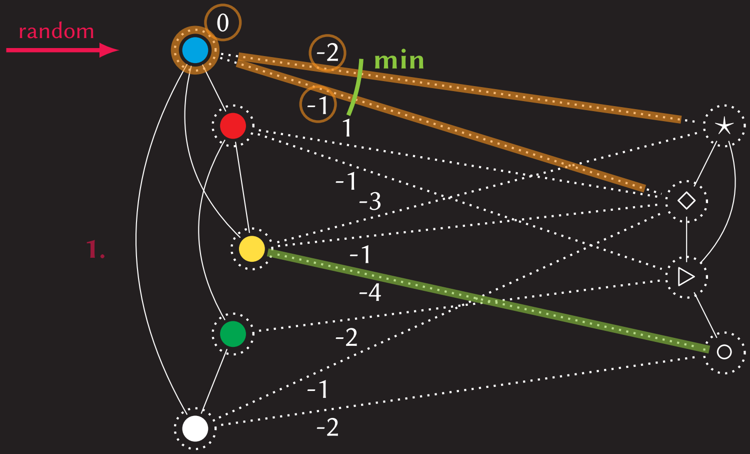
1.



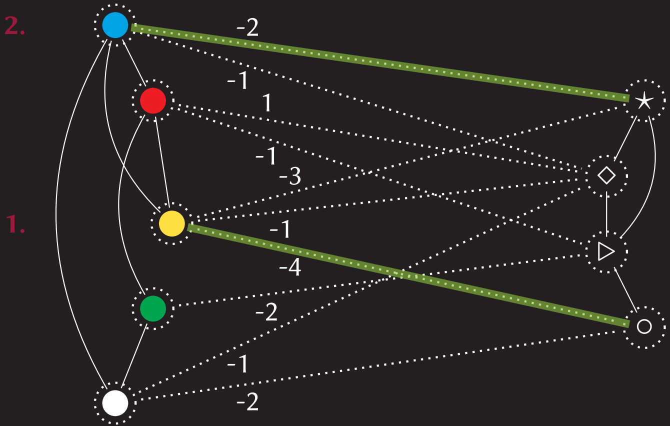
Randomized greedy heuristic



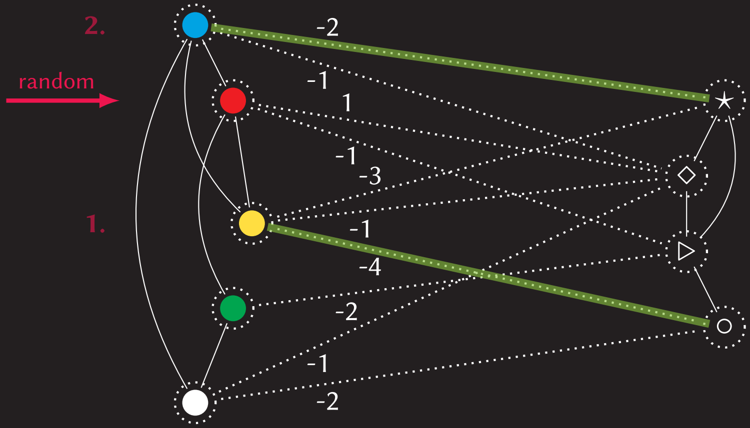
Randomized greedy heuristic



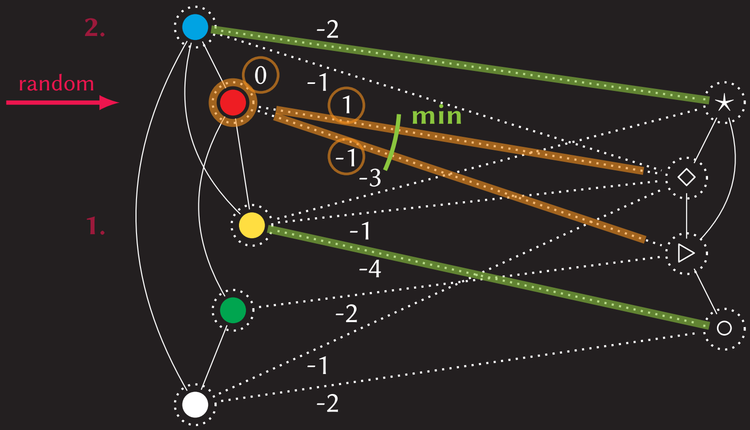
Randomized greedy heuristic



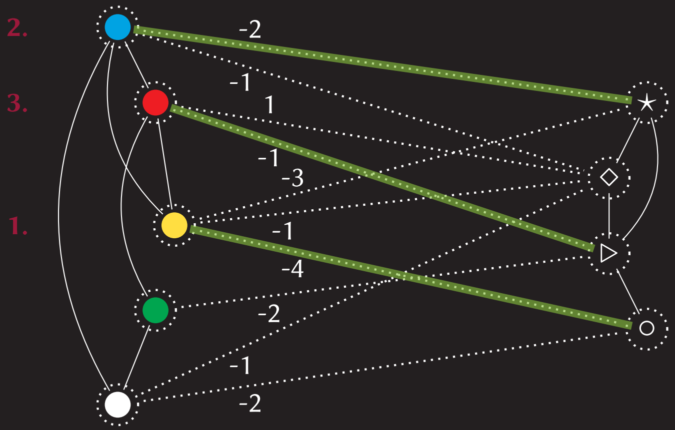
Randomized greedy heuristic



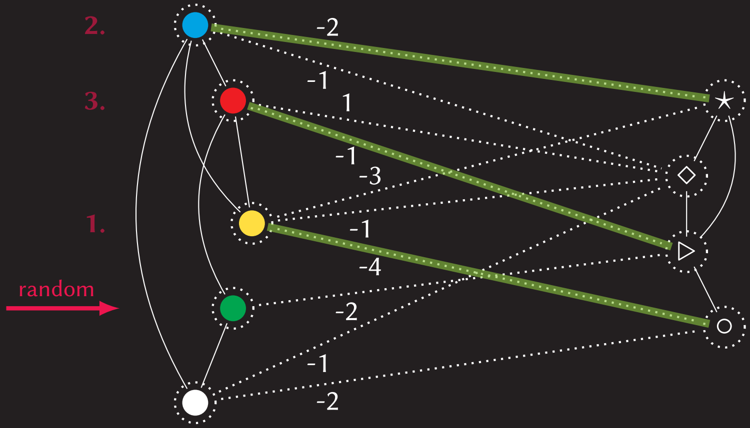
Randomized greedy heuristic



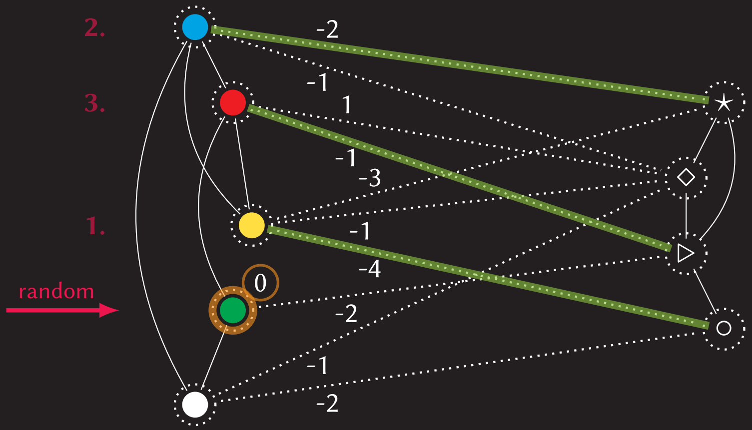
Randomized greedy heuristic



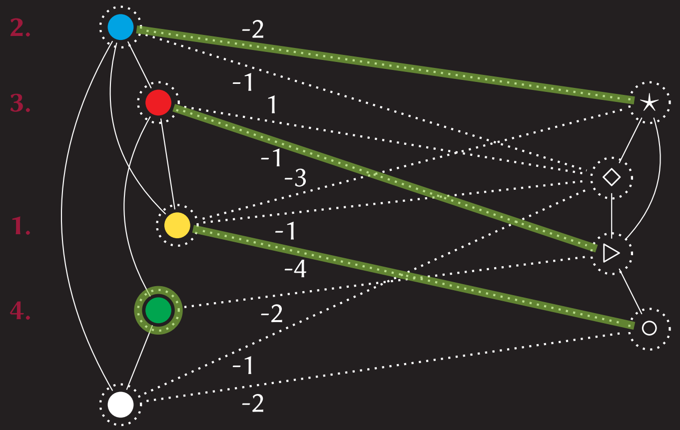
Randomized greedy heuristic



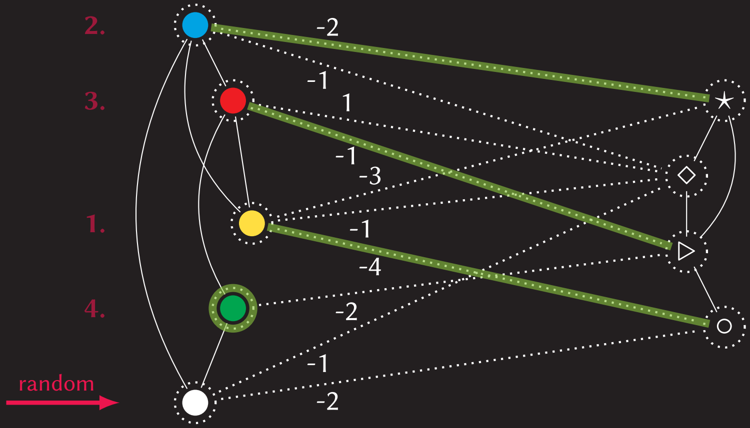
Randomized greedy heuristic



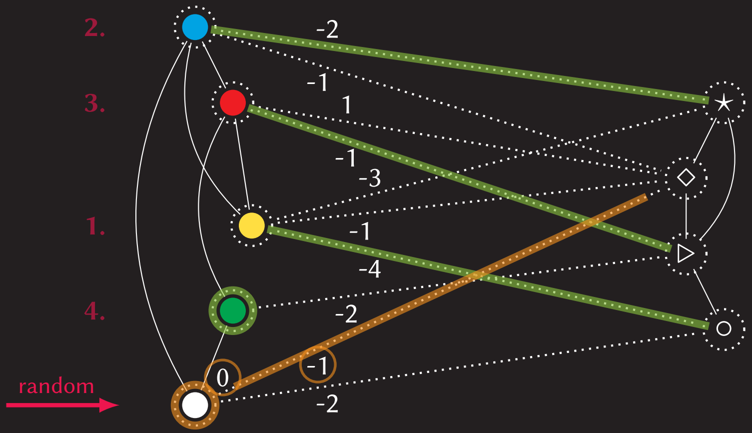
Randomized greedy heuristic



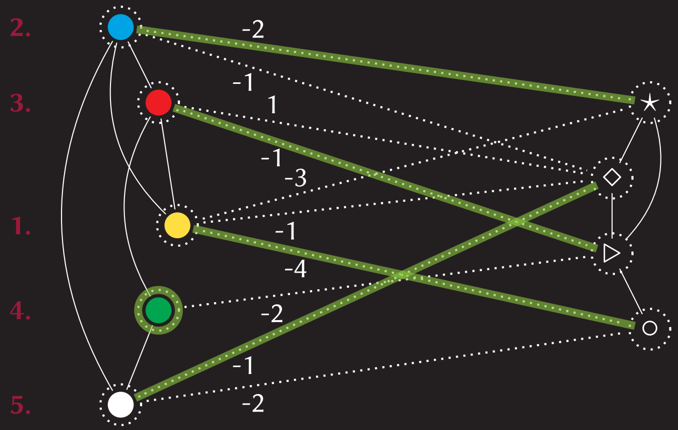
Randomized greedy heuristic



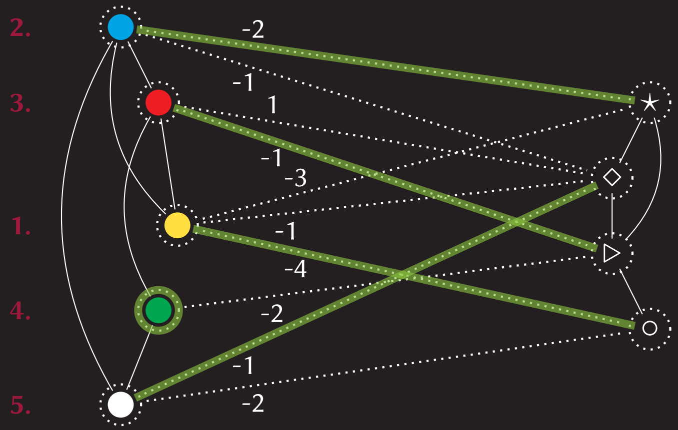
Randomized greedy heuristic

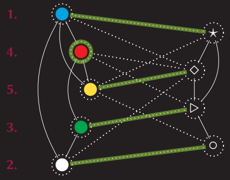


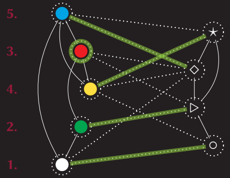
Randomized greedy heuristic



Randomized greedy heuristic







Performance on benchmark datasets

Percentage of instances solved to optimality; average time needed

	HBP [Zha16]	AMP [Swo17]	DD [Tor13]	*our*	
`hotel, house`	0.15 *s*	0.2 *s*	0.02 *s*	0.01 *s*	timeout: 1 *s*
`motor, car`	0.13 *s*	0.08 *s*	0.11 *s*	0.01 *s*	timeout: 1 *s*
`opengm`	2.71 *s*	–	1.08 *s*	0.004 *s*	timeout: 10 *s*
`flow`	–	0.13 *s*	1.66 *s*	0.06 *s*	timeout: 10 *s*
`worms`	–	6.45 *s*	–	0.39 *s*	timeout: 10 *s*

[Tor13] Torresani *et al.*: **A dual decomposition approach to feature correspondence**. *PAMI '13*
[Zha16] Zhang *et al.*: **Pairwise matching through max-weight bipartite belief propagation**. *CVPR '16*
[Swo17] Swoboda *et al.*: **A study of Lagrangean decompositions and dual ascent solvers for graph matching**. *CVPR '17*

Performance on benchmark datasets

Percentage of instances solved to optimality; average time needed

	HBP [Zha16]	AMP [Swo17]	DD [Tor13]	*our*	
hotel, house	0.15 *s*	0.2 *s*	0.02 *s*	0.01 *s*	timeout: 1 *s*
motor, car	0.13 *s*	0.08 *s*	0.11 *s*	0.01 *s*	
opengm	2.71 *s*	–	1.08 *s*	0.004 *s*	timeout: 10 *s*
flow	–	0.13 *s*	1.66 *s*	0.06 *s*	
worms	–	6.45 *s*	–	0.39 *s*	

[Tor13] Torresani *et al.*: **A dual decomposition approach to feature correspondence**. *PAMI '13*
[Zha16] Zhang *et al.*: **Pairwise matching through max-weight bipartite belief propagation**. *CVPR '16*
[Swo17] Swoboda *et al.*: **A study of Lagrangean decompositions and dual ascent solvers for graph matching**. *CVPR '17*

Performance on benchmark datasets

Percentage of instances solved to optimality; average time needed

	HBP [Zha16]		AMP [Swo17]		DD [Tor13]		*our*		
hotel, house		0.15 *s*		0.2 *s*		0.02 *s*		0.01 *s*	timeout: 1 *s*
motor, car		0.13 *s*		0.08 *s*		0.11 *s*		0.01 *s*	
opengm		2.71 *s*		–		1.08 *s*		0.004 *s*	timeout: 10 *s*
flow		–		0.13 *s*		1.66 *s*		0.06 *s*	
worms		–		6.45 *s*		–		0.39 *s*	

[Tor13] Torresani *et al.*: **A dual decomposition approach to feature correspondence**. *PAMI '13*
[Zha16] Zhang *et al.*: **Pairwise matching through max-weight bipartite belief propagation**. *CVPR '16*
[Swo17] Swoboda *et al.*: **A study of Lagrangean decompositions and dual ascent solvers for graph matching**. *CVPR '17*

Conclusions

- We propose a **randomized greedy primal heuristic**.
- We show **how to use fusion moves** for graph matching.
- Together with block-coordinate ascent this results in a highly efficient solver **suitable for deep graph matching**.

Lisa Hutschenreiter
Heidelberg University, Germany
lisa.hutschenreiter@iwr.uni-heidelberg.de

ICCV 2021 OCTOBER 11-17 VIRTUAL

Join me online during **Q & A Session 5**:

- Tuesday, October 12, at 4 to 5 pm EDT
- Thursday, October 14, at 9 to 10 am EDT

Code and datasets available via
https://vislearn.github.io/libmpopt/iccv2021/