One Model, Any CSP: Graph Neural Networks as Fast Global Search Heuristics for Constraint Satisfaction

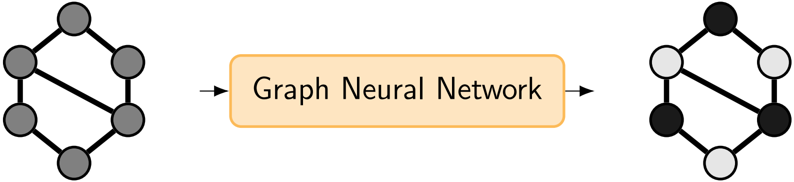
Jan Tönshoff, Jakob Lindner, Berke Kisin, Martin Theisen, Martin Grohe

RWTH Aachen

toenshoff@informatik.rwth-aachen.de

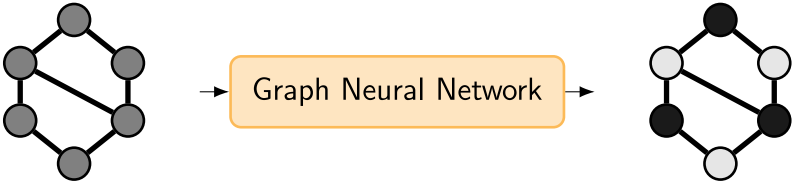
Neural Combinatorial Optimization

Learn heuristics for combinatorial optimization with Graph Neural Networks:



Neural Combinatorial Optimization

Learn heuristics for combinatorial optimization with Graph Neural Networks:

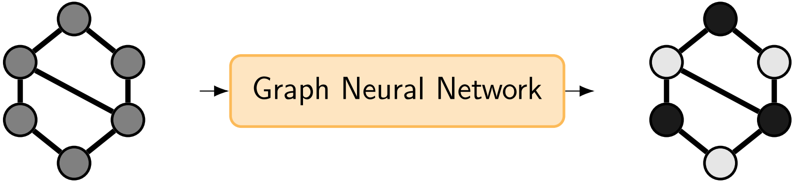


Pros

- Learn novel algorithms from scratch
- Data-driven fine-tuning

Neural Combinatorial Optimization

Learn heuristics for combinatorial optimization with Graph Neural Networks:



Pros

- Learn novel algorithms from scratch
- Data-driven fine-tuning

Cons

- Computationally expensive
- Problem specific approaches

Constraint Satisfaction Problems

Constraint Satisfaction Problems

$\mathcal{I} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$

- Variables $\mathcal{X} = \{X_1, \ldots, X_n\}$
- Domains $\mathcal{D} = \{\mathcal{D}(X_1), \ldots, \mathcal{D}(X_n)\}$
- Constraints $\mathcal{C} = \{C_1, \ldots, C_m\}$

Constraint Satisfaction Problems

$\mathcal{I} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$

- Variables $\mathcal{X} = \{X_1, \dots, X_n\}$
- Domains $\mathcal{D} = \{\mathcal{D}(X_1), \dots, \mathcal{D}(X_n)\}$
- Constraints $\mathcal{C} = \{C_1, \dots, C_m\}$

Variable assignment α: $\alpha(X) \in \mathcal{D}(X)$

Constraint Satisfaction Problems

$\mathcal{I} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$

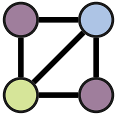
- Variables $\mathcal{X} = \{X_1, \dots, X_n\}$
- Domains $\mathcal{D} = \{\mathcal{D}(X_1), \dots, \mathcal{D}(X_n)\}$
- Constraints $\mathcal{C} = \{C_1, \dots, C_m\}$

Variable assignment α: $\alpha(X) \in \mathcal{D}(X)$

Boolean SAT:

$$f = (X_1 \vee \neg X_2) \wedge X_3$$

Graph Coloring:



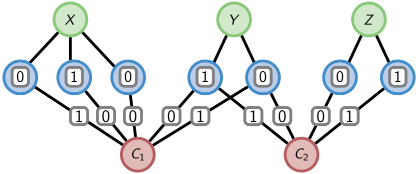
ANYCSP

Are **N**eural Networks great heuristics? **Y**es, for **CSP**s!

- Design unified graph representation and GNN architecture for all CSPs
- Optimize global search heuristic with reinforcement learning

Graph Representation

$$G(\mathcal{I}, \alpha)$$



Learnable Heuristics

$$\pi_\theta$$

$$G(\mathcal{I}, \alpha^{(t)}) \longrightarrow \boxed{\text{GNN } \pi_\theta} \longrightarrow \alpha^{(t+1)}$$

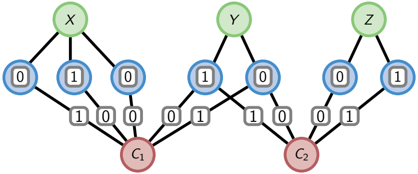
ANYCSP

Are **N**eural Networks great heuristics? **Y**es, for **CSP**s!

- Design unified graph representation and GNN architecture for all CSPs
- Optimize global search heuristic with reinforcement learning

Graph Representation

$G(\mathcal{I}, \alpha)$



Learnable Heuristics

π_θ

$$(X_1 \lor \neg X_2) \land X_3 \rightarrow \boxed{\text{GNN } \pi_\theta} \rightarrow \begin{array}{l} X_1 = 1 \\ X_2 = 0 \\ X_3 = 1 \end{array}$$

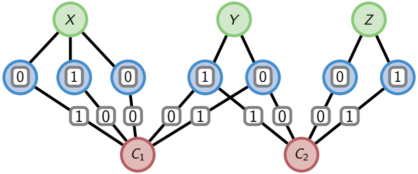
ANYCSP

Are **N**eural Networks great heuristics? **Y**es, for **CSP**s!

- Design unified graph representation and GNN architecture for all CSPs
- Optimize global search heuristic with reinforcement learning

Graph Representation

$G(\mathcal{I}, \alpha)$



Learnable Heuristics

π_θ

Constraint Value Graph

CSP Instance $\mathcal{I}$:

$\mathcal{X} = \{X, Y, Z\}$

$D_X = \{1, 2, 3\}$
$D_Y = \{1, 2\}$
$D_Z = \{1, 2\}$

$C_1 : X \leq Y$
$C_2 : Y \neq Z$

Assignment $\alpha = (2, 1, 2)$

Constraint Value Graph

CSP Instance $\mathcal{I}$:

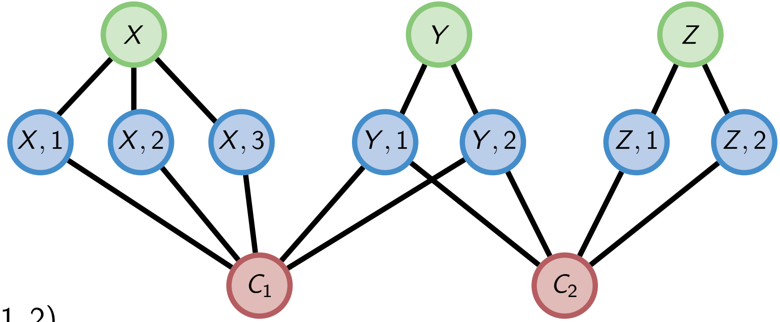
$\mathcal{X} = \{X, Y, Z\}$

$D_X = \{1, 2, 3\}$
$D_Y = \{1, 2\}$
$D_Z = \{1, 2\}$

$C_1 : X \leq Y$
$C_2 : Y \neq Z$

Assignment $\alpha = (2, 1, 2)$

$G(\mathcal{I}, \alpha) = (V, E, L_{\mathcal{D}}, L_{\mathcal{C}})$



Constraint Value Graph

CSP Instance $\mathcal{I}$:

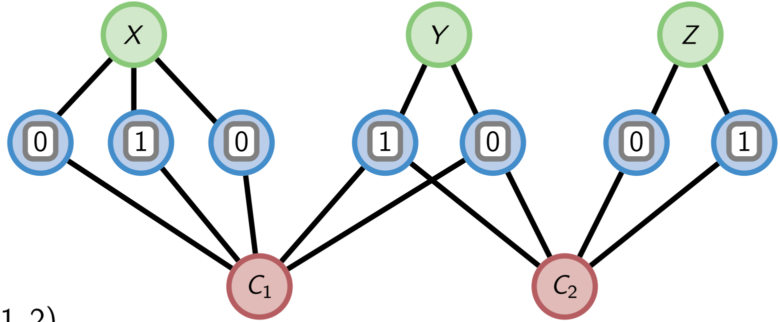
$\mathcal{X} = \{X, Y, Z\}$

$D_X = \{1, 2, 3\}$
$D_Y = \{1, 2\}$
$D_Z = \{1, 2\}$

$C_1 : X \leq Y$
$C_2 : Y \neq Z$

Assignment $\alpha = (2, 1, 2)$

$G(\mathcal{I}, \alpha) = (V, E, L_{\mathcal{D}}, L_{\mathcal{C}})$



Constraint Value Graph

CSP Instance $\mathcal{I}$:

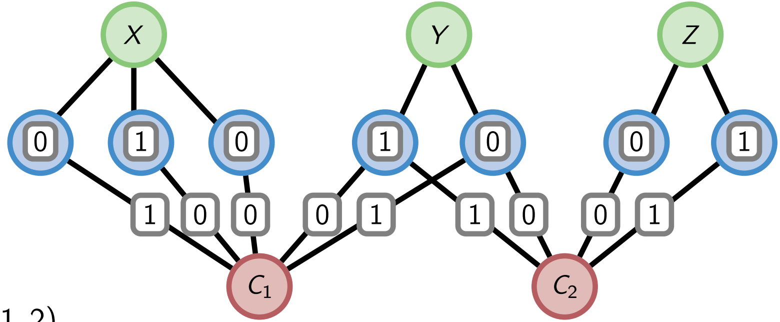
$\mathcal{X} = \{X, Y, Z\}$

$D_X = \{1, 2, 3\}$
$D_Y = \{1, 2\}$
$D_Z = \{1, 2\}$

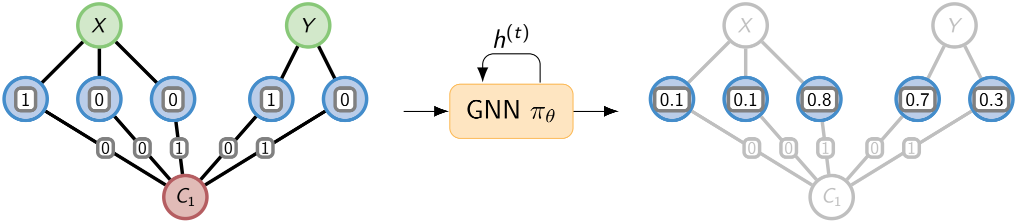
$C_1 : X \leq Y$
$C_2 : Y \neq Z$

Assignment $\alpha = (2, 1, 2)$

$G(\mathcal{I}, \alpha) = (V, E, L_{\mathcal{D}}, L_{\mathcal{C}})$



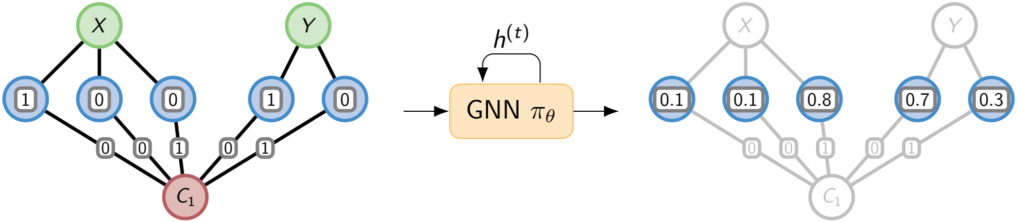
Policy GNN



Our GNN π_θ is a trainable stochastic global search policy:

- Input: $G(\mathcal{I}, \alpha^{(t)})$, recurrent states $h^{(t)}$
- Output: Soft assignment $\varphi^{(t+1)}$

Policy GNN



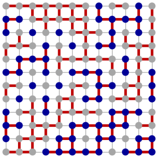
Our GNN π_θ is a trainable stochastic global search policy:

- Input: $G(\mathcal{I}, \alpha^{(t)})$, recurrent states $h^{(t)}$
- Output: Soft assignment $\varphi^{(t+1)}$
- Next assignment: $\alpha^{(t+1)} \sim \varphi^{(t+1)}$

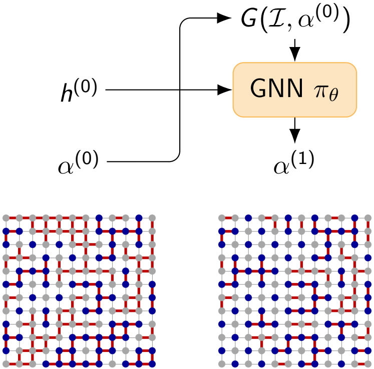
Stochastic Global Search

$h^{(0)}$

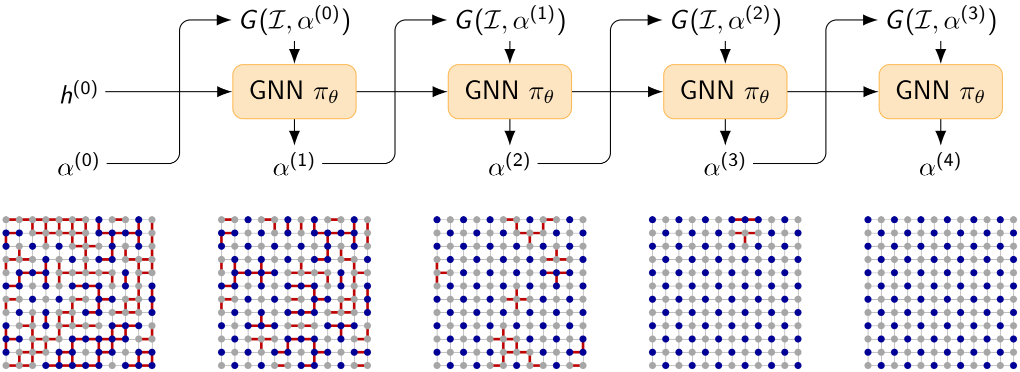
$\alpha^{(0)}$



Stochastic Global Search



Stochastic Global Search



Training

Given a training distribution Ω we optimize π_θ with respect to the reinforcement learning objective

$$\theta^* = \arg\max_{\theta} \mathop{\mathbf{E}}_{\substack{\mathcal{I}\sim\Omega \\ \boldsymbol{\alpha}\sim\pi_\theta(\mathcal{I})}} \left[\sum_{t=1}^{T} \gamma^{t-1} r^{(t)} \right]$$

Training

Given a training distribution Ω we optimize π_θ with respect to the reinforcement learning objective

$$\theta^* = \arg\max_{\theta} \mathop{\mathbf{E}}_{\substack{\mathcal{I}\sim\Omega \\ \boldsymbol{\alpha}\sim\pi_\theta(\mathcal{I})}} \left[\sum_{t=1}^{T} \gamma^{t-1} r^{(t)} \right]$$

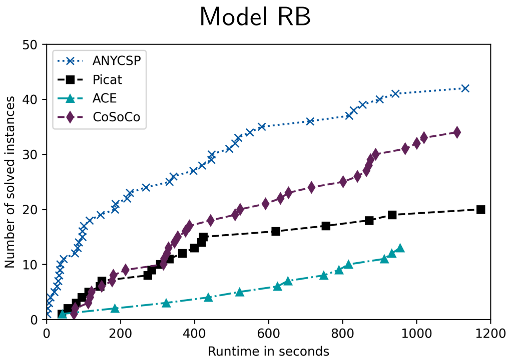
Quality in iteration t:

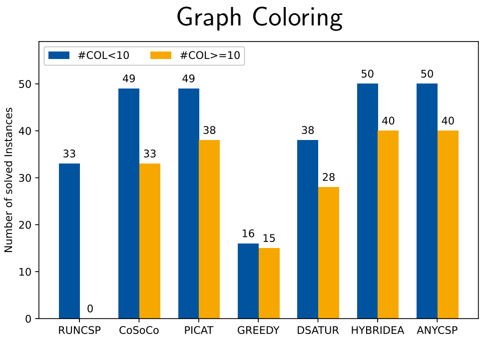
$$Q_{\mathcal{I}}(\alpha) := \frac{|\{C \in \mathcal{C} \ : \ \alpha \models C\}|}{|\mathcal{C}|}$$

Reward in iteration t:

$$r^{(t)} := \max\left\{0,\ Q_{\mathcal{I}}(\alpha^{(t)}) - \max_{0 \leq t' < t} Q_{\mathcal{I}}(\alpha^{(t')})\right\},$$

Experiments





SAT

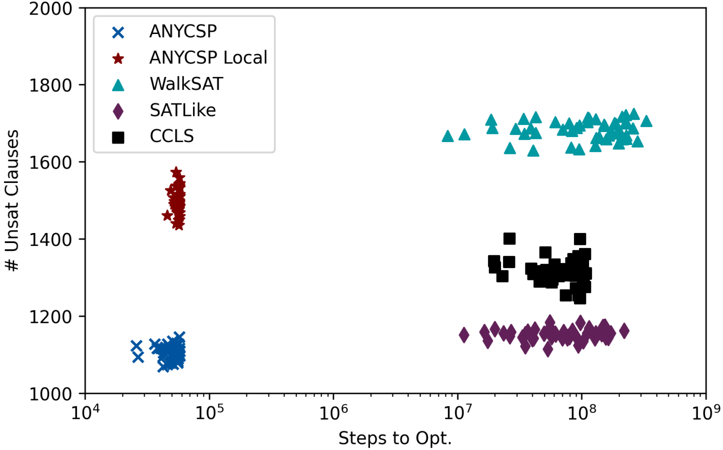
Instances: Random 3SAT Instances from SATLIB.

Metric: Number of satisfied instances.

Method	SL50	SL100	SL150	SL200	SL250
RLSAT	**100**	87	67	27	12
PDP	93	79	72	57	61
WalkSAT	**100**	**100**	97	93	87
ProbSAT	**100**	**100**	97	87	92
ANYCSP	**100**	**100**	**100**	**97**	**99**

MAX-SAT

Instances: 50 5-CNF formulas with 10K variables and 300K clauses.



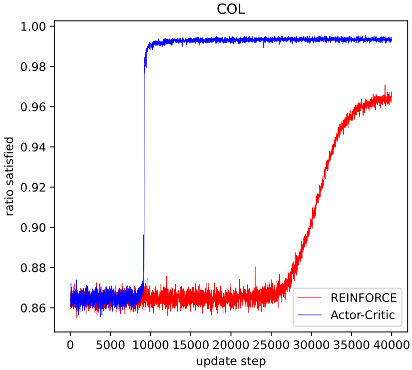
Further Modifications

Changes

- Actor-Critic architecture
- Geometric reward
- Entropy regularization

Results

- earlier, faster, and more robust learning
- better in distribution and on decision problems
- less stable



Conclusion

ANYCSP:

- Constraint Value Graphs: A generic and compact representation for CSPs
- Reinforcement learning applied to exponential action spaces

Conclusion

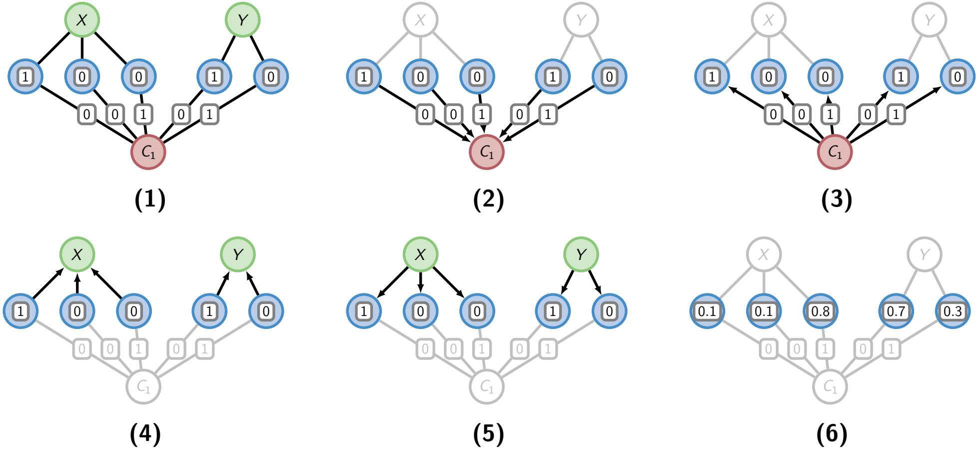
ANYCSP:

- Constraint Value Graphs: A generic and compact representation for CSPs
- Reinforcement learning applied to exponential action spaces

Empirical Observations:

- CSP heuristics can be obtained purely through data-driven training
- GNNs parameterize a powerful and versatile class of global search heuristics

π_θ: Message Passing Scheme



MaxCut

Instances: (Unweighted) GSet graphs

Metric: Mean absolute deviation from best known cut value.

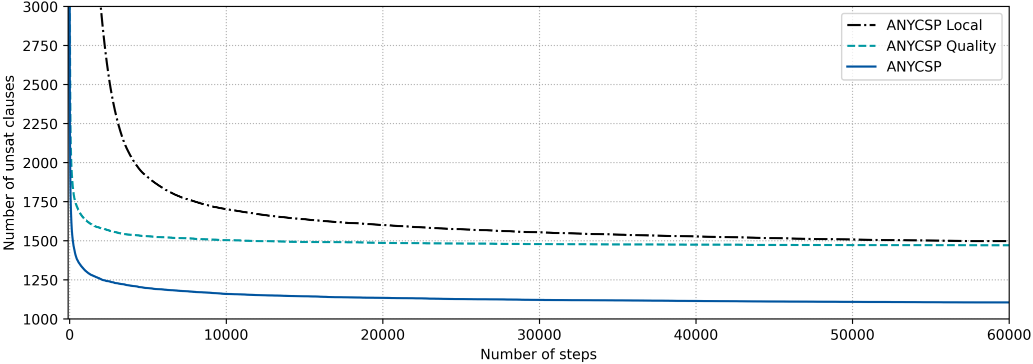
Method	$\|V\|$=800	$\|V\|$=1K	$\|V\|$=2K	$|V|\geq$3K
Greedy	411.44	359.11	737.00	774.25
SDP	245.44	229.22	-	-
RUNCSP	185.89	156.56	357.33	401.00
ECO-DQN	65.11	54.67	157.00	428.25
ECORD	8.67	8.78	39.22	187.75
ANYCSP	**1.22**	**2.44**	**13.11**	**51.63**

Cross-Comparison

Training Distribution Ω vs Test CSPs:

Ω	RB50	$\text{COL}_{<10}$	Gset800	SL250	MAX-5-CNF
$\Omega_{\textbf{RB}}$	**42**	**50**	655.56	98	6192.18
$\Omega_{\textbf{COL}}$	15	**50**	868.22	96	5076.16
$\Omega_{\textbf{MCUT}}$	0	0	**1.22**	0	9048.64
$\Omega_{\textbf{3SAT}}$	0	19	1213.11	**99**	5001.72
$\Omega_{\textbf{MSAT}}$	0	15	1217.67	66	**1103.14**

Ablation


3000
2750
2500
2250
2000
1750
1500
1250
1000
Number of unsat clauses
0
10000
20000
30000
40000
50000
60000
Number of steps
ANYCSP Local
ANYCSP Quality
ANYCSP

REINFORCE

Given a training distribution Ω we optimize π_θ with respect to the reinforcement learning objective

$$\theta^* = \arg\max_{\theta} \mathop{\mathbf{E}}_{\substack{\mathcal{I}\sim\Omega \\ \boldsymbol{\alpha}\sim\pi_\theta(\mathcal{I})}} \left[\sum_{t=1}^{T} \gamma^{t-1} r^{(t)} \right]$$

using the gradient estimation given by REINFORCE (Williams, 1992)

$$-\frac{1}{T} \sum_{t=1}^{T} G_t \frac{\nabla \pi_\theta(\alpha^{(t)} \mid \alpha^{(t-1)}, h^{(t-1)}, \mathcal{I})}{\pi_\theta(\alpha^{(t)} \mid \alpha^{(t-1)}, h^{(t-1)}, \mathcal{I})}$$

where G_t is the empirical gain

$$G_t = \sum_{k=t}^{T} \gamma^{k-t} r^{(k)}$$

Critic

A **critic** $\mathfrak{c}$ learns to estimate gain G_t from recurrent state $h^{(t)}$, which speeds up learning through a baseline (replacing G_t in policy gradient):

$$A_t = G_t - \mathfrak{c}(h^{(t)})$$

and temporal difference learning (replacing G_t everywhere):

$$G_t^{(n)} = \gamma^{n+1}\mathfrak{c}(h^{(t+n+1)}) + \sum_{k=t}^{t+n} \gamma^{k-t} r^{(k)},$$

$$G_t(\lambda) = \lambda^{T+1} G_t + (1-\lambda) \sum_{n=0}^{T} \lambda^n G_t^{(\min(n,T-t))}$$

Modified Reward

Geometric reward including quality and improvement:

$$r_g^{(t)} = \sqrt{(Q_{\mathcal{I}}(\alpha^{(t)}) - Q_{\mathcal{I}}(\alpha^{(0)})) \cdot r^{(t)}}$$

Entropy regularization incentivizes exploration:

$$r_e^{(t)} = r_g^{(t)} - \sigma \log \mathbf{P}(\alpha^{(t)} \mid \varphi_\theta^{(t)})$$

References I

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

Jan Tönshoff, Berke Kisin, Jakob Lindner, and Martin Grohe. One model, any csp: Graph neural networks as fast global search heuristics for constraint satisfaction. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 4280–4288, 8 2023. doi: 10.24963/ijcai.2023/476. URL `https://doi.org/10.24963/ijcai.2023/476`.

Jan Tönshoff, Martin Ritzert, Hinrikus Wolf, and Martin Grohe. Graph neural networks for maximum constraint satisfaction. *Frontiers in Artificial Intelligence*, 3, 2021. ISSN 2624-8212. doi: 10.3389/frai.2020.580607. URL `https://www.frontiersin.org/article/10.3389/frai.2020.580607`.

References II

Saeed Amizadeh, Sergiy Matusevych, and Markus Weimer. Pdp: A general neural framework for learning constraint satisfaction solvers. *arXiv preprint arXiv:1903.01969*, 2019.

Gilles Audemard, Frédéric Boussemart, Christophe Lecoutre, Cédric Piette, and Olivier Roussel. Xcsp3 and its ecosystem. *Constraints*, 25(1):47–69, 2020.

Thomas D Barrett, Christopher WF Parsonson, and Alexandre Laterre. Learning to solve combinatorial graph partitioning problems via efficient exploration. *arXiv preprint arXiv:2205.14105*, 2022.

Thomas Barrett, William Clements, Jakob Foerster, and Alex Lvovsky. Exploratory combinatorial optimization with reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3243–3250, 2020.

References III

Elias Khalil, Hanjun Dai, Yuyu Zhang, Bistra Dilkina, and Le Song. Learning combinatorial optimization algorithms over graphs. *Advances in neural information processing systems*, 30, 2017.

Michel X Goemans and David P Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6):1115–1145, 1995.

Rhyd Lewis. *A guide to graph colouring*, volume 7. Springer, 2015.

Rhyd Lewis, Jonathan Thompson, Christine Mumford, and Jonathan Gillard. A wide-ranging computational comparison of high-performance graph colouring algorithms. *Computers & Operations Research*, 39(9):1933–1950, 2012.

Yinyu Ye. Gset. https://web.stanford.edu/˜yyye/yyye/Gset/, 2003.

Bart Selman, Henry A Kautz, Bram Cohen, et al. Local search strategies for satisfiability testing. *Cliques, coloring, and satisfiability*, 26:521–532, 1993.

References IV

Selman Kautz. Walksat home page, 2019. URL `https://www.cs.rochester.edu/u/kautz/walksat/`.

Ke Xu and Wei Li. Many hard examples in exact phase transitions. *Science Direct Working Paper No S1574-034X (04)*, pages 70228–8, 2003.

Gilles Audemard, Christophe Lecoutre, and Emmanuel Lonca, editors. *XCSP3 Competition 2022 Proceedings*, XCSP3 Competition, Artois, France, 2022.

Neng-Fa Zhou. An xcsp3 solver in picat. In *XCSP3 Competition 2022 Proceedings*, XCSP3 Competition, pages 79–81, 2022.

Christophe Lecoutre. Ace a generic constraint solver. In *XCSP3 Competition 2022 Proceedings*, XCSP3 Competition, pages 58–59, 2022.

Gilles Audemard. Cosoco 1.12. In *XCSP3 Competition 2018 Proceedings*, XCSP3 Competition, pages 78–79, 2018.

References V

Una Benlic and Jin-Kao Hao. Breakout local search for the max-cutproblem. *Engineering Applications of Artificial Intelligence*, 26(3):1162 – 1173, 2013. ISSN 0952-1976. doi: https://doi.org/10.1016/j.engappai.2012.09.001. URL `http://www.sciencedirect.com/science/article/pii/S0952197612002175`.

Hermish Mehta. Cvx graph algorithms. https://github.com/hermish/cvx-graph-algorithms, 2019.

Chuan Luo, Shaowei Cai, Wei Wu, Zhong Jie, and Kaile Su. Ccls: An efficient local search algorithm for weighted maximum satisfiability. *IEEE Transactions on Computers*, 64(7):1830–1843, 2015. doi: 10.1109/TC.2014.2346196.

Shaowei Cai and Zhendong Lei. Old techniques in new ways: Clause weighting, unit propagation and hybridization for maximum satisfiability. *Artificial Intelligence*, 287: 103354, 2020.

References VI

Philippe Galinier and Jin-Kao Hao. Hybrid evolutionary algorithms for graph coloring. *Journal of combinatorial optimization*, 3(4):379–397, 1999.