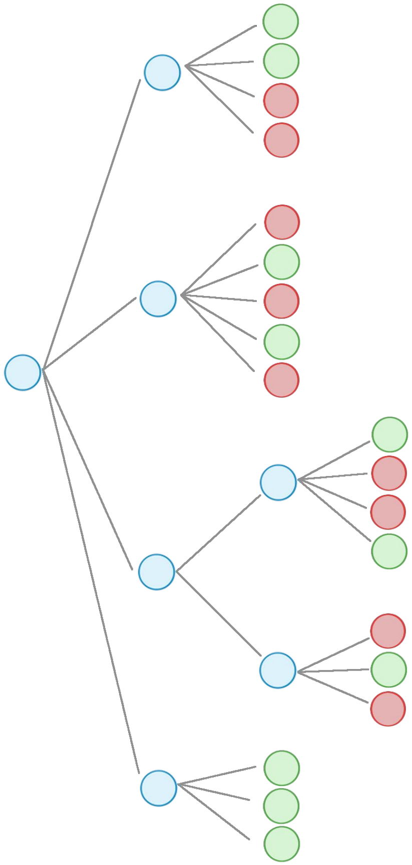
Learning Domain-Specific Heuristics with Graph Convolutional Networks

Matheus Z. Marcon
advisor: Felipe Meneguzzi

matheus.marcon@edu.pucrs.br

Heuristic Planning

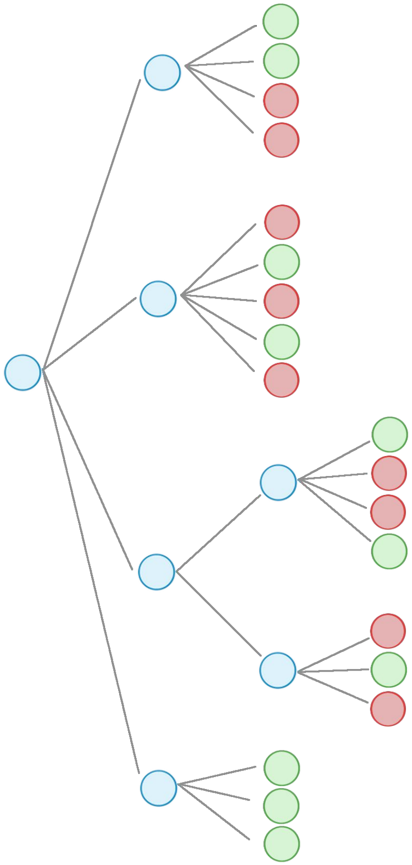
- Planning can become expensive.
- Heuristics focus the search on nodes that seem more promising.
- Might trade optimality, completeness and precision for performance.



Heuristic Planning

- Heuristics need to be **informative.**

- Off-the-shelf heuristics might suffer from poor performance in complex scenarios.



Domain-specific Heuristics

- Domain-independent fail to capture domain singularities.
- Specific design needs expert domain knowledge.

➜ Might be unfeasible for real world problems.

Domain-specific Heuristics

- Domain-independent fail to capture domain singularities.
- Specific design needs expert domain knowledge.

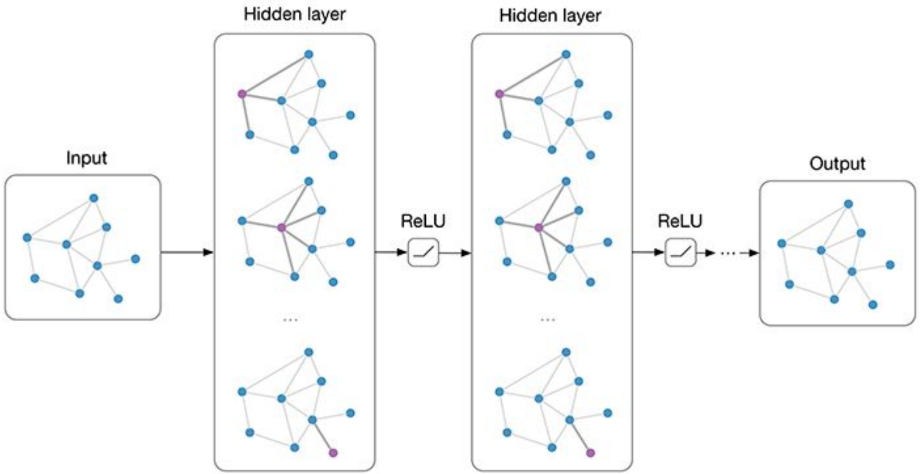
➜ Might be unfeasible for real world problems.

How to find a solution independently of human knowledge?

Proposed Method

Graph Convolutional Networks (GCNs):

- Graph-based model with node-wise heuristic values as output.



Dataset

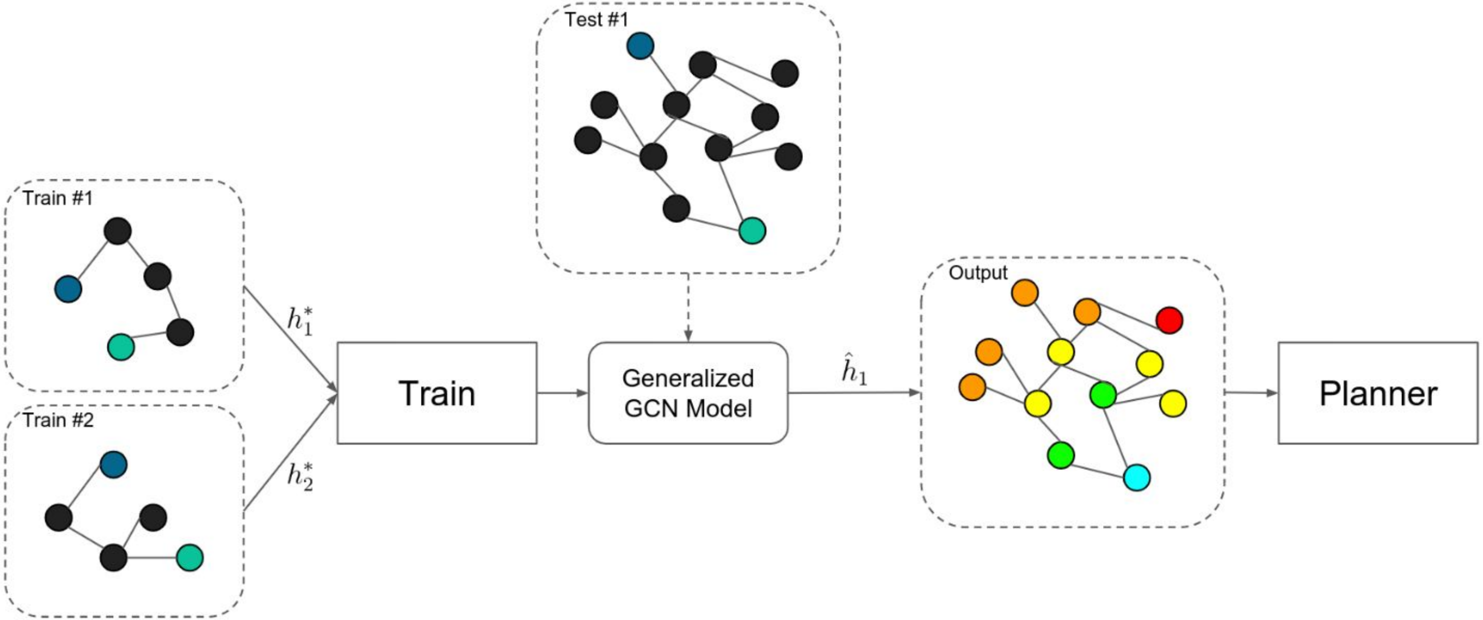
PDDL benchmark domains problem generators.

source: Joerg Hoffman, FF Domain Collection

https://fai.cs.uni-saarland.de/hoffmann/ff-domains.html

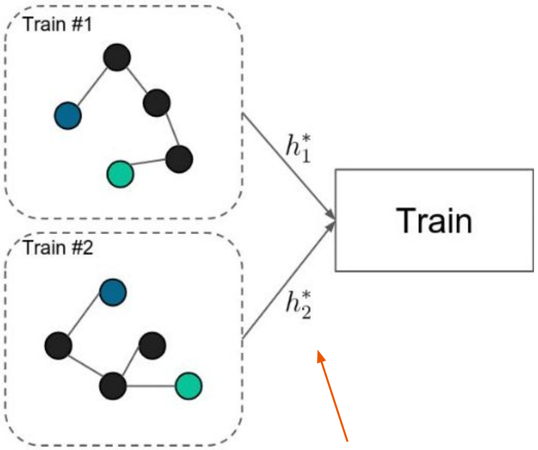
Used domains: Blocksworld-4ops and Logistics

Proposed Method



Proposed Method

n subgraphs for each task



perfect heuristics for each node

Graph building

Expansion level: 0

Graph building

Initial State for given task

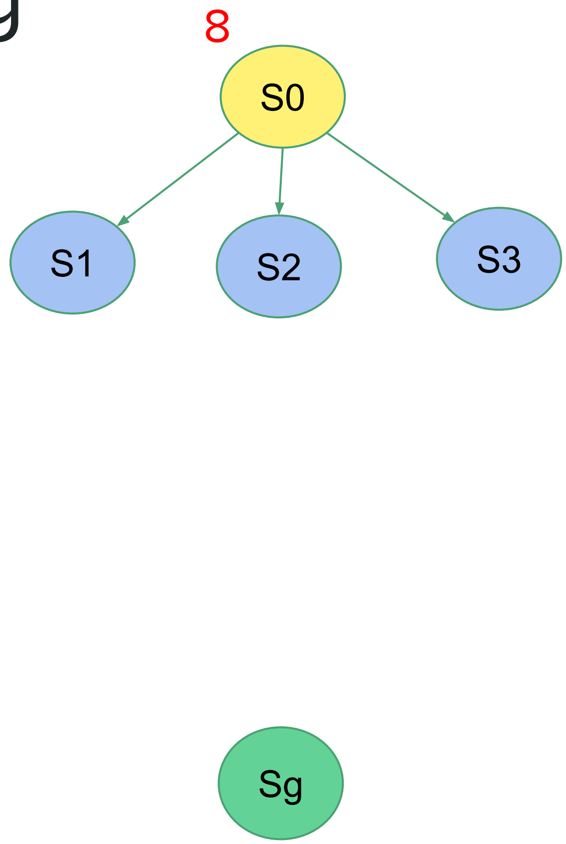
A* search

hFF heuristic

- **- accuracy**
- **+ speed**

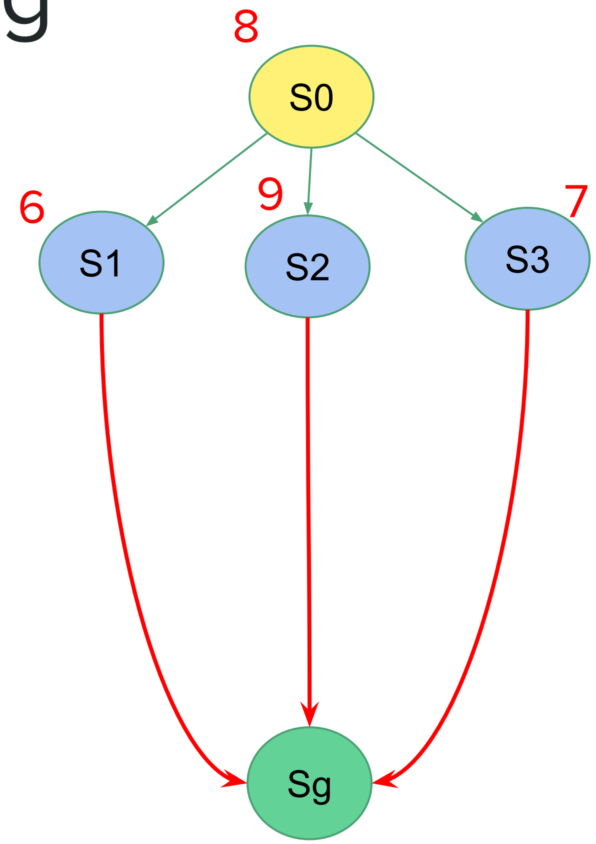
Expansion level: 0

Graph building



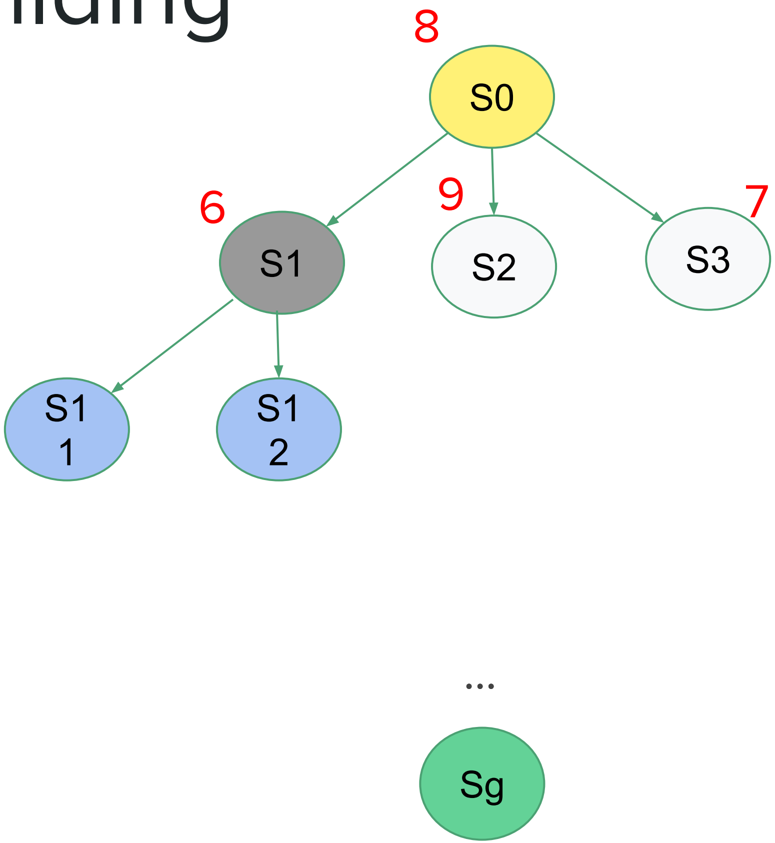
Expansion level: 1

Graph building



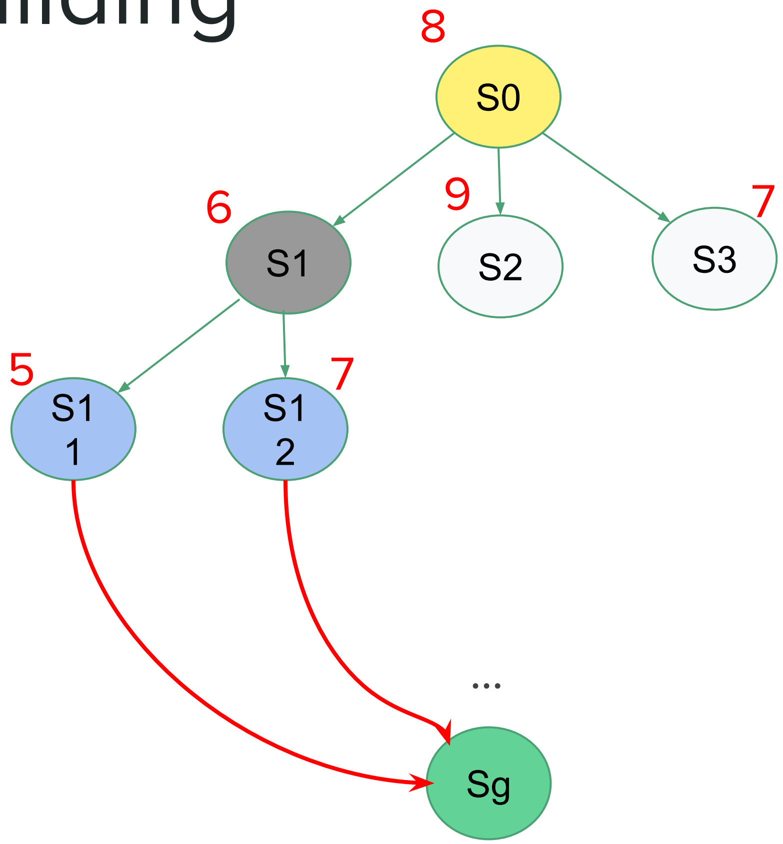
Expansion level: 1

Graph building



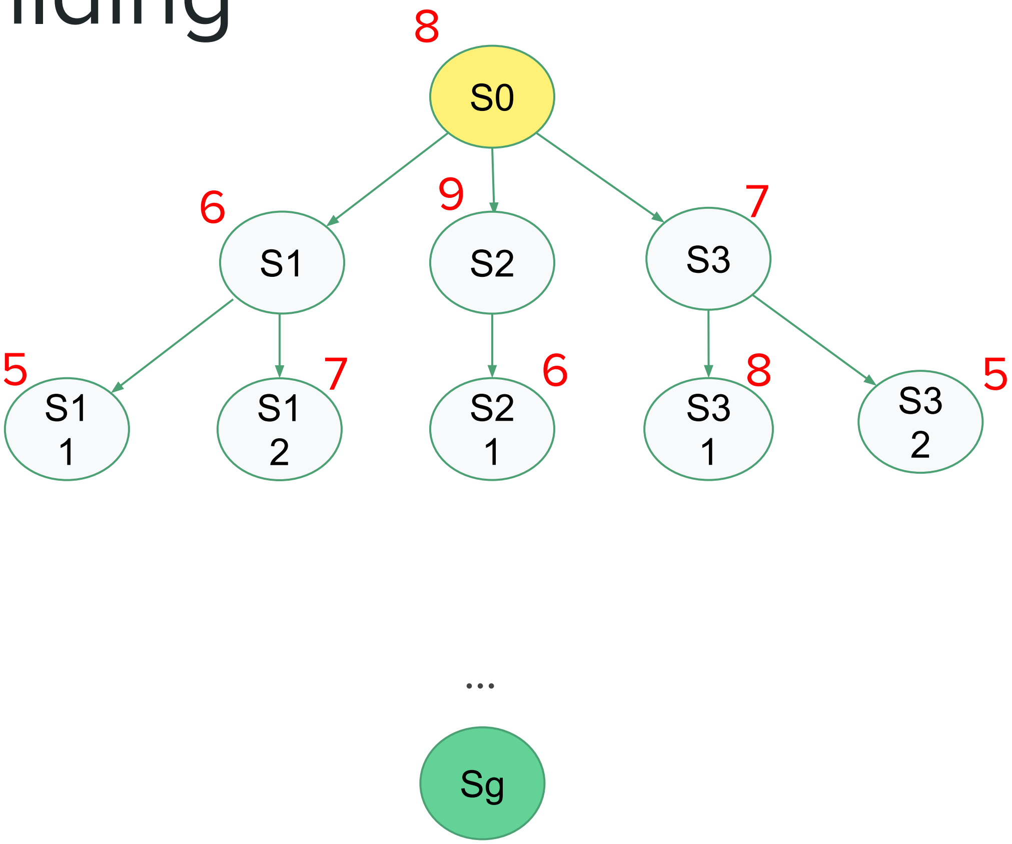
Expansion level: 2

Graph building



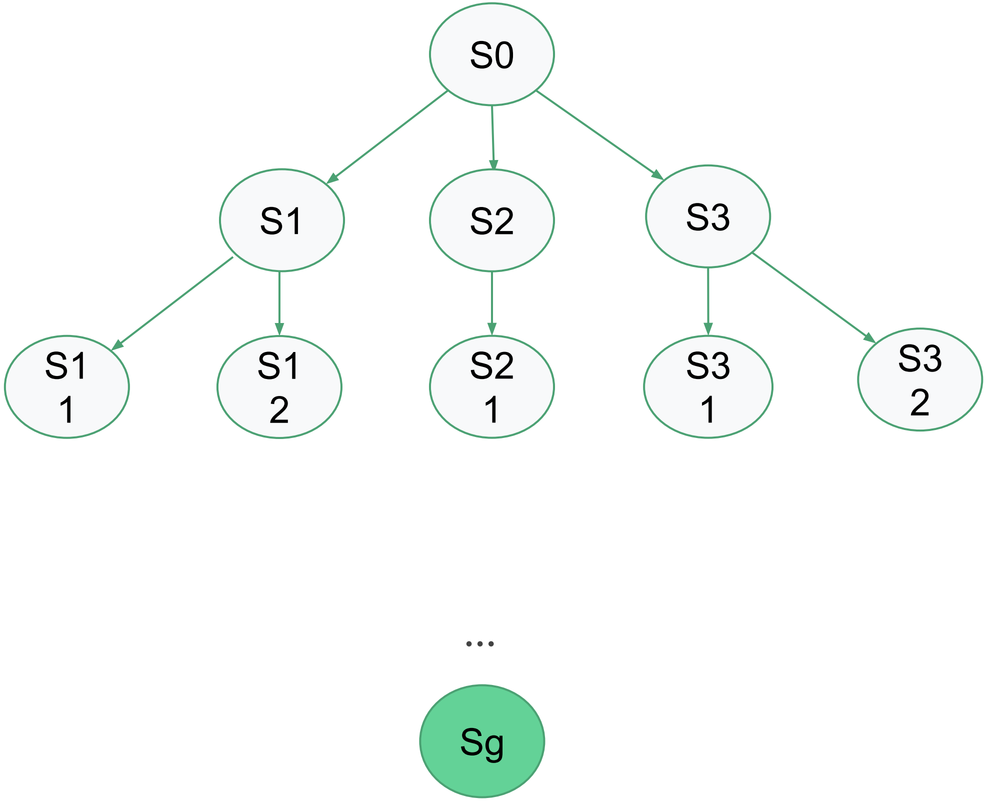
Expansion level: 2

Graph building

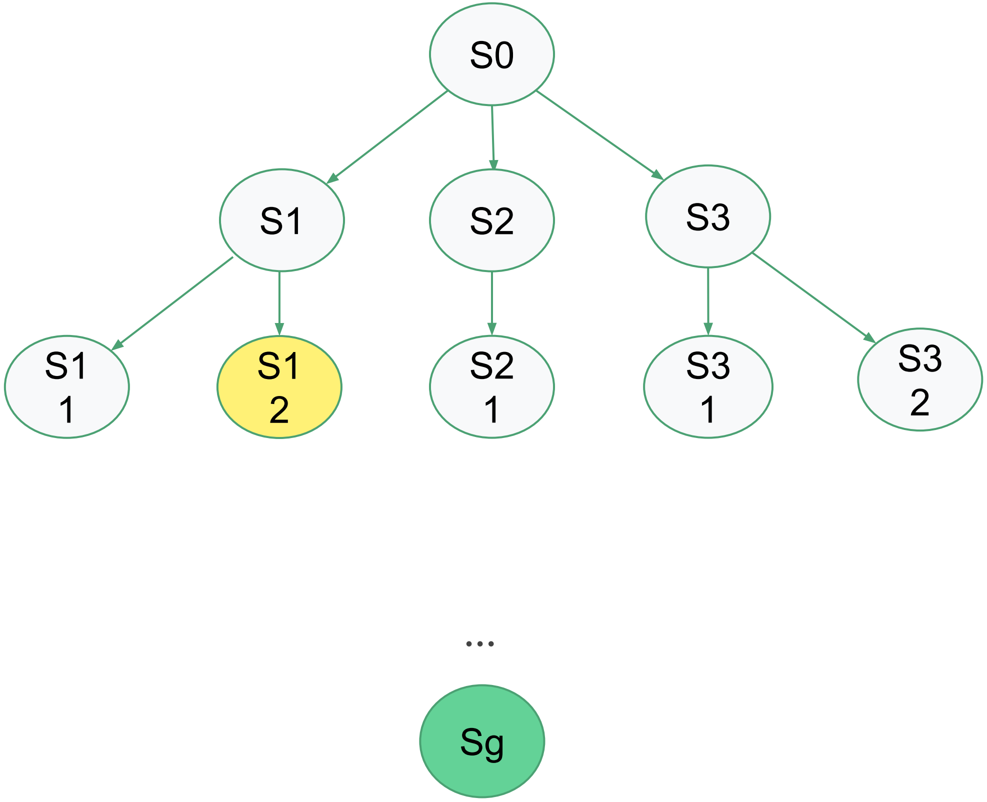


Expansion level: **2**

Graph building



Graph building

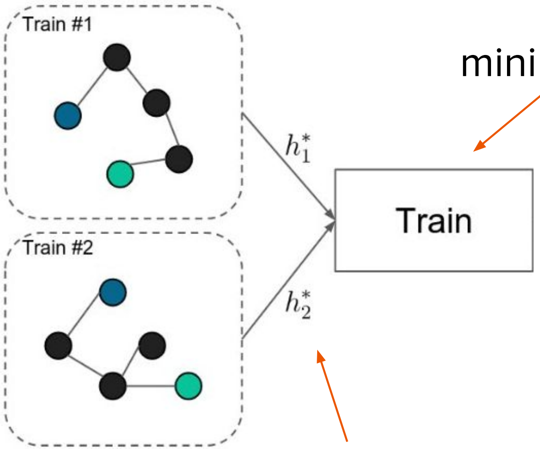


Graph building

Proposed Method

n subgraphs for each task

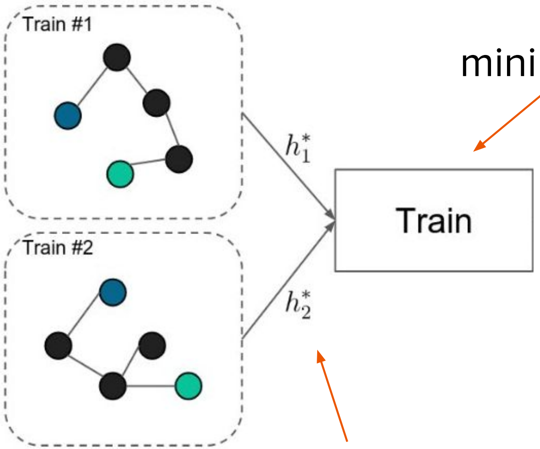
minimize loss



perfect heuristics for each node

Proposed Method

n subgraphs for each task

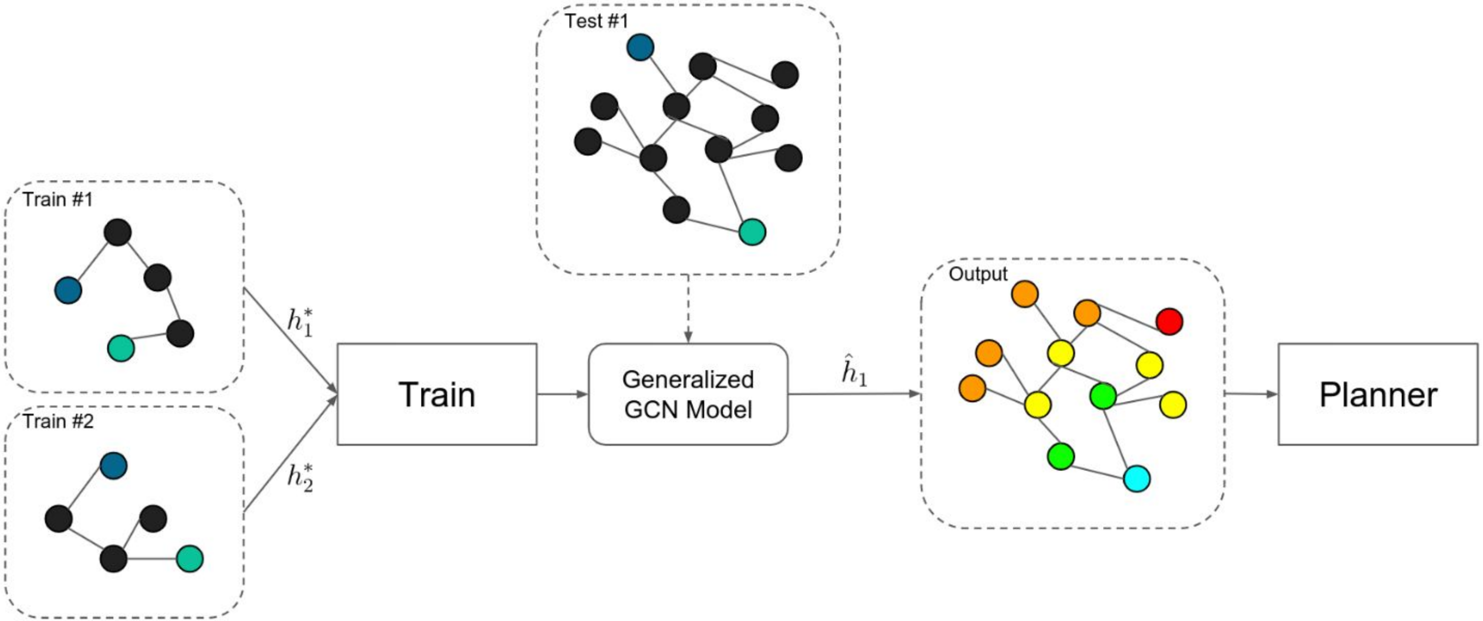


minimize loss

$$L_h(g) = \frac{1}{N}\sum_{i=1}^{N}(\hat{h}_{i,g} - h^*_{i,g})^2$$

perfect heuristics for each node

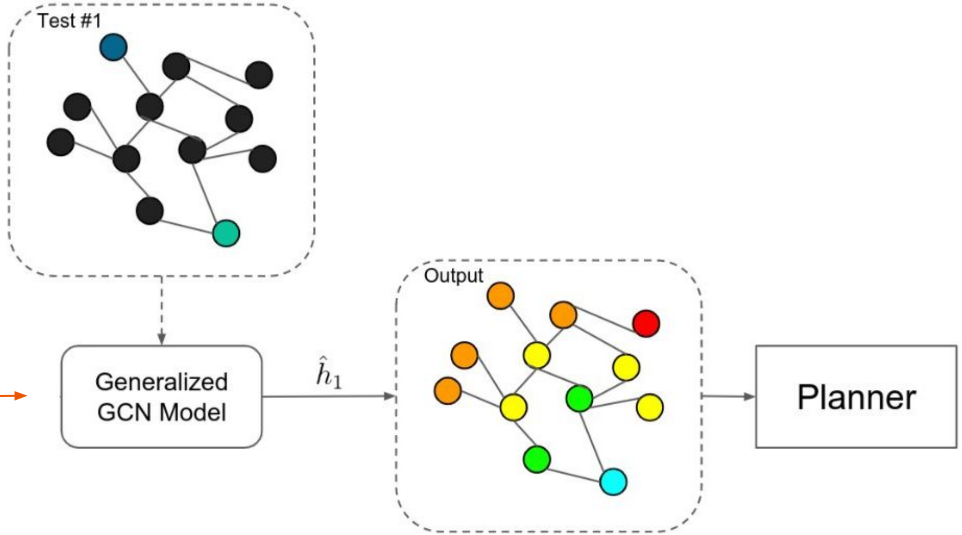
Proposed Method



Proposed Method

test problem instance

trained model

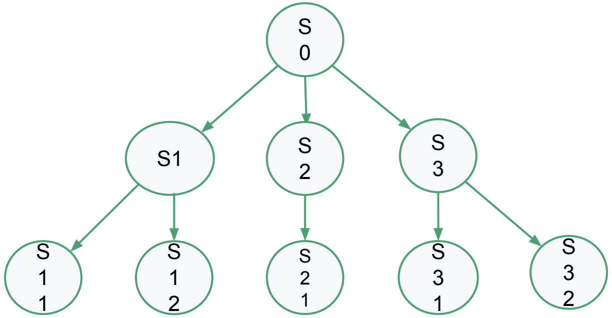


predicted value for each node

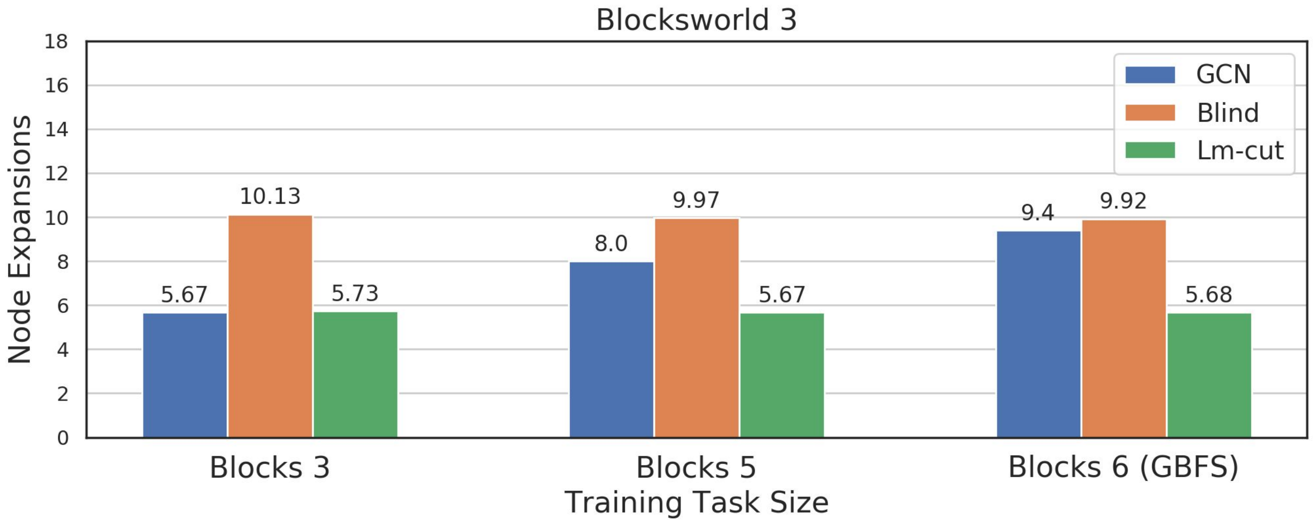
Inference

- Nodes must be expanded before feeding the network.

- Inference happens in constant time.

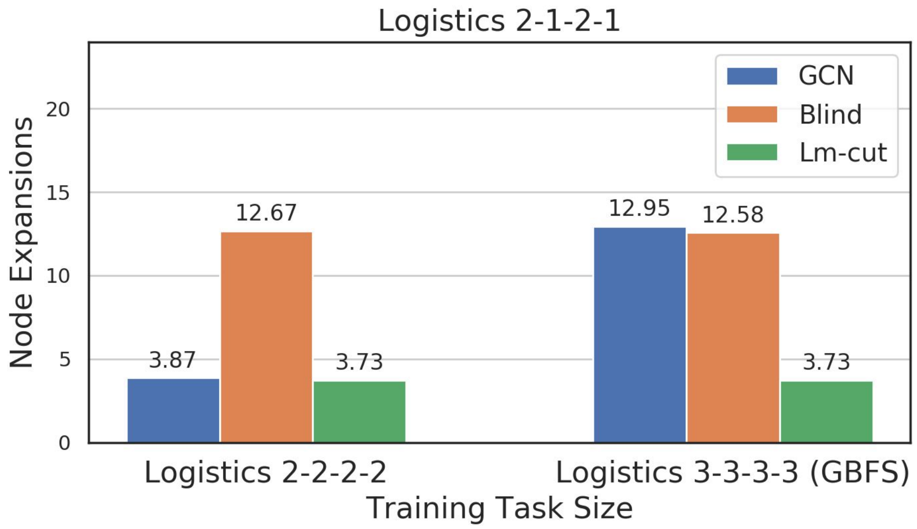


Results



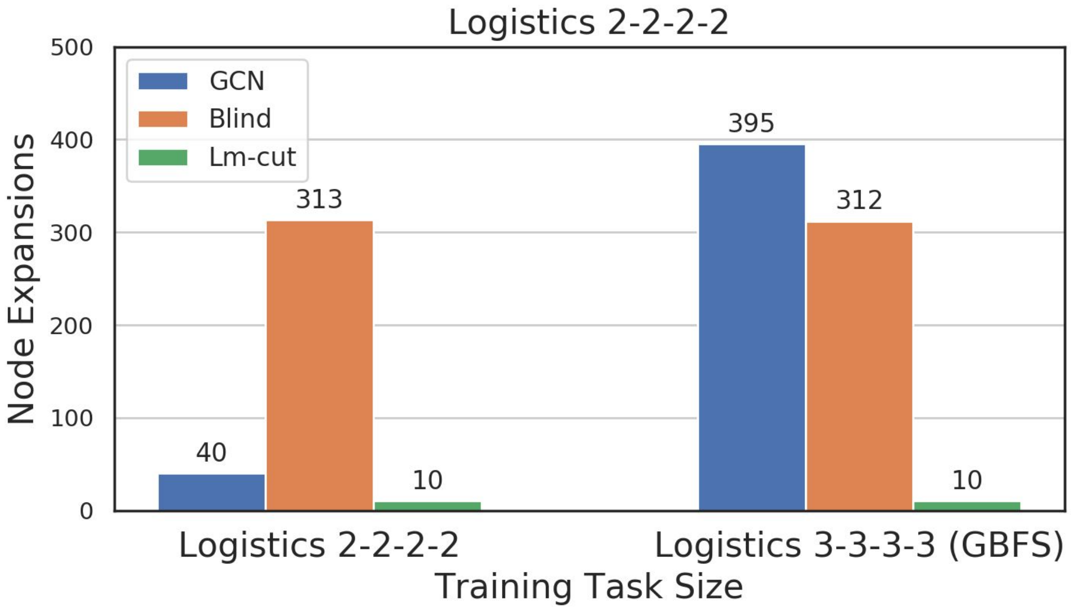
Blocksworld 3
Node Expansions
18
16
14
12
10
8
6
4
2
0
GCN
Blind
Lm-cut
10.13
5.67
5.73
9.97
8.0
5.67
9.4
9.92
5.68
Blocks 3
Blocks 5
Blocks 6 (GBFS)
Training Task Size

Results



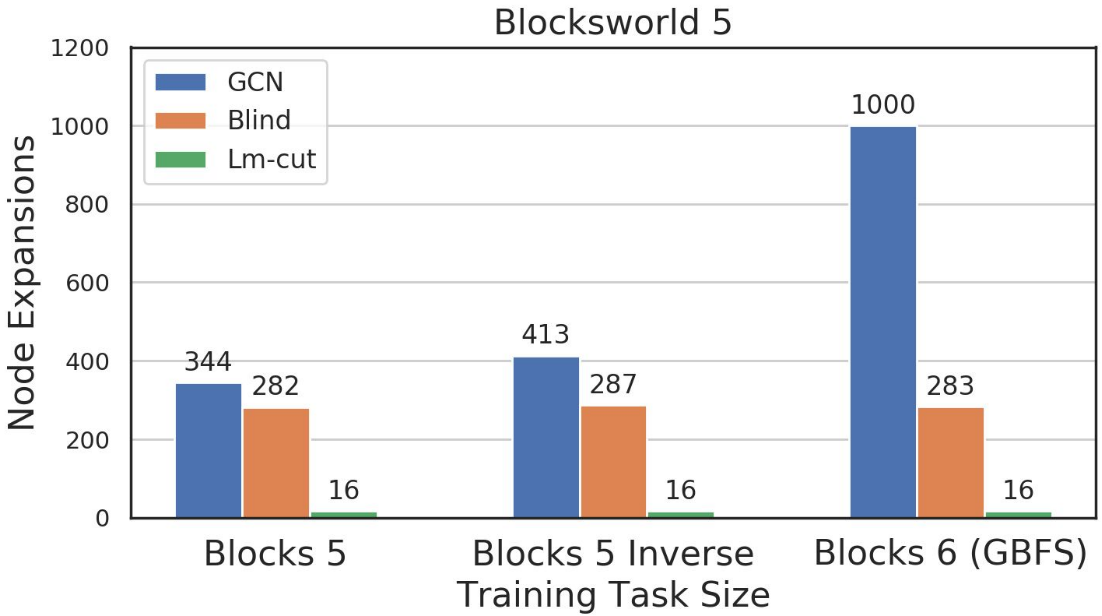
Logistics 2-1-2-1
GCN
Blind
Lm-cut
Node Expansions
0
5
10
15
20
3.87
12.67
3.73
12.95
12.58
3.73
Logistics 2-2-2-2
Logistics 3-3-3-3 (GBFS)
Training Task Size

Results



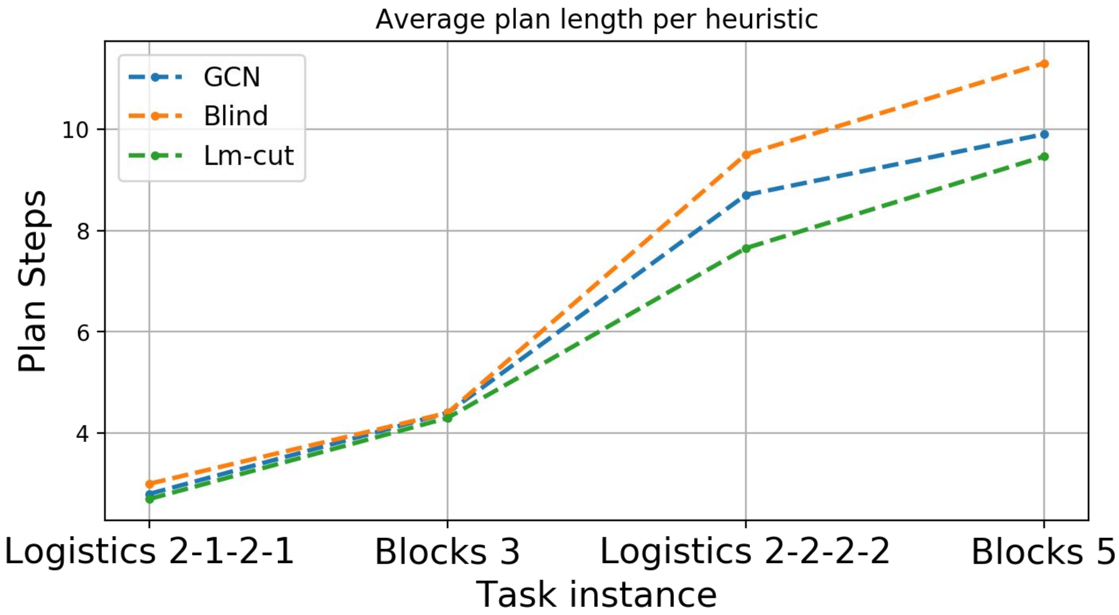
Logistics 2-2-2-2
GCN
Blind
Lm-cut
Node Expansions
0
100
200
300
400
500
40
313
10
395
312
10
Logistics 2-2-2-2
Logistics 3-3-3-3 (GBFS)
Training Task Size

Results

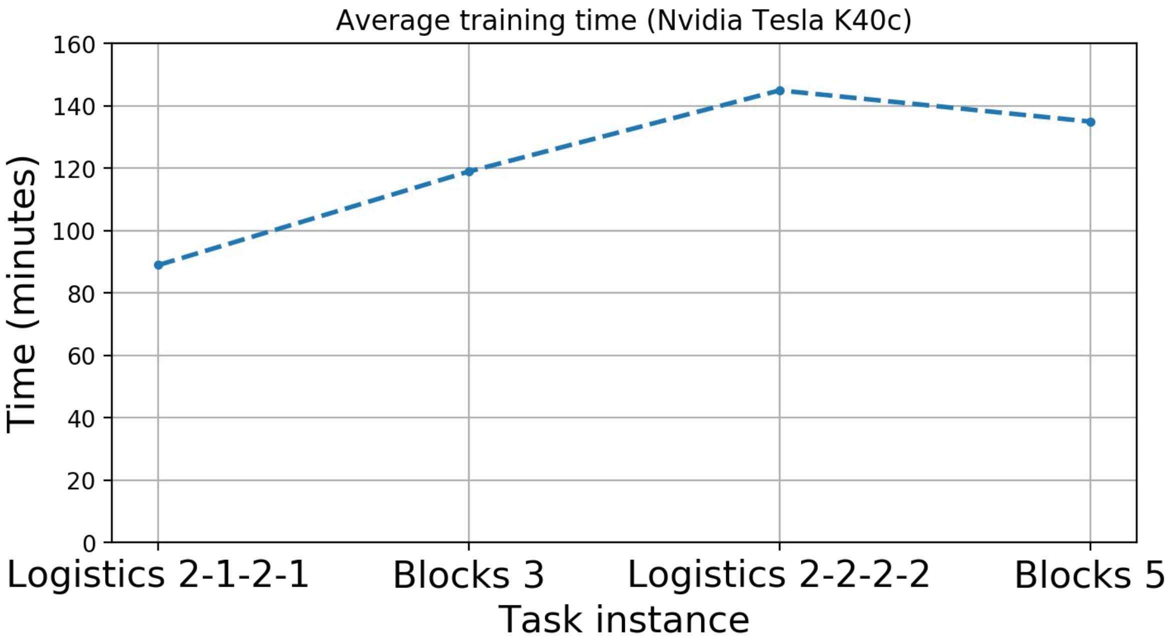


Blocksworld 5
Node Expansions
1200
1000
800
600
400
200
0
GCN
Blind
Lm-cut
344
282
16
413
287
16
1000
283
16
Blocks 5
Blocks 5 Inverse
Blocks 6 (GBFS)
Training Task Size

Results


Average plan length per heuristic
GCN
Blind
Lm-cut
Plan Steps
10
8
6
4
Logistics 2-1-2-1
Blocks 3
Logistics 2-2-2-2
Blocks 5
Task instance

Results



Average training time (Nvidia Tesla K40c)
Time (minutes)
160
140
120
100
80
60
40
20
0
Logistics 2-1-2-1
Blocks 3
Logistics 2-2-2-2
Blocks 5
Task instance

Conclusions

- Our approach is unable to learn semantic information.
- Good results can be achieved through exhaustive training.
- Unfeasible for large domain instances.

Future Work

- Employ different GCN implementations (varying graph sizes).
- Investigate different network architectures.
- Improve node sampling techniques for graph generation.