IPC: A Benchmark Data Set for Learning with Graph-Structured Data

https://github.com/IBM/IPC-graph-data

Patrick Ferber [1,2], Tengfei Ma [3], Siyu Huo [3], Jie Chen [3,4], Michael Katz [3]. [1] University of Basel, [2] Saarland University, [3] IBM Research, [4] MIT-IBM Watson AI Lab

Purpose

- Benchmarking graph methods (e.g., graph kernels, GNNs)
- Offering a large-scale data set complementary to current benchmarks
- Leveling up computational challenges for graph methods

Data Set Summary

- IPC has two self-contained versions---grounded and lifted---from the same set of planning problems
- Both versions are directed graphs. The lifted version is acyclic
- Each version contains 2,439 graphs (train/val/test = 2,008/286/145)
- Each graph has 17 numeric target values
- Graph nodes are equipped with one-hot features
- Graph generator and labeling (target values) are computer programmed; hence, the data set can be extended if needed

Characteristics and Implications

- The graphs are significantly larger than other benchmarks (39% in IPC-grounded and 63% in IPC-lifted have > 1k nodes)
  - The largest graph is the memory bottleneck in GNN training.
- Graph sizes are highly skewed
  - What does it mean by saying "a graph with 10 nodes is similar to another one with 100k nodes?"
- Similar to other data sets, the IPC graphs are not necessarily connected.
  - However, the main connected component generally dominates.
- The graphs have a moderate diameter
  - Does diameter affects the number of GNN layers?
- The lifted graphs are the most sparse.

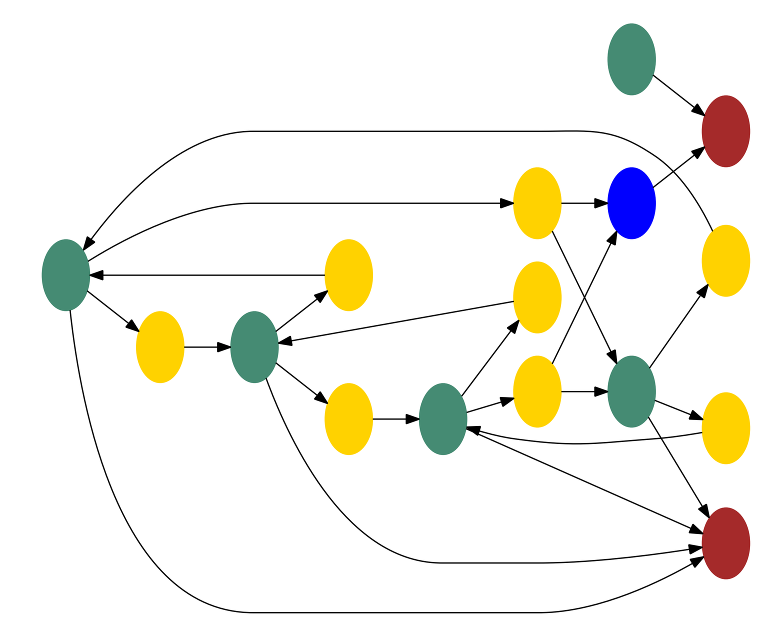
Graph Construction

- For *Problem Description Graph*, see Sievers, S., Katz, M., Sohrabi, S., Samulowitz, H., and Ferber, P. Deep learning for cost-optimal planning: Task-dependent planner selection. In Proc. AAAI 2019.
- For *Abstract Structure Graph*, see Sievers, S., Röger, G., Wehrle, M., and Katz, M. Theoretical foundations for structural symmetries of lifted PDDL tasks. In Proc. ICAPS 2019.

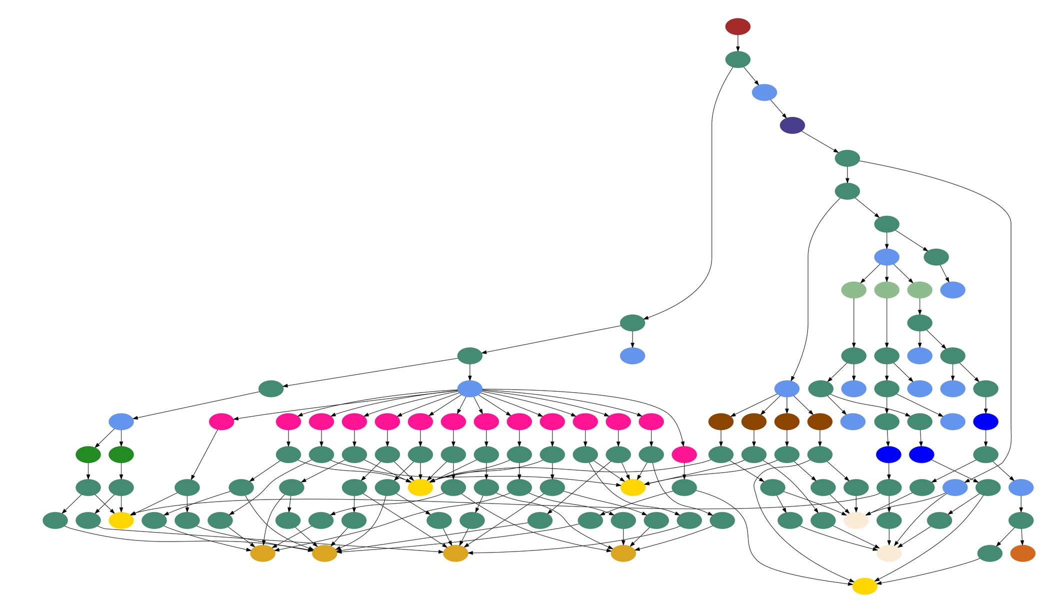
```
(define (domain grid-visit-all)
(:requirements :typing)
(:types        place - object)
(:predicates
      (connected ?x ?y - place)
      (at-robot ?x - place)
      (visited ?x - place))

(:action move
:parameters (?curpos
             ?nextpos - place)
:precondition (and
             (at-robot ?curpos)
             (connected ?curpos ?nextpos))
:effect (and
             (at-robot ?nextpos)
             (not (at-robot ?curpos))
             (visited ?nextpos)))
)
```

```
(define (problem grid-2)
(:domain grid-visit-all)
(:objects
    loc-x0-y0
    loc-x0-y1
    loc-x1-y0
    loc-x1-y1 - place)
(:init
    (at-robot loc-x1-y1)
    (visited loc-x1-y1)
    (connected loc-x0-y0 loc-x1-y0)
    (connected loc-x0-y0 loc-x0-y1)
    (connected loc-x0-y1 loc-x1-y1)
    (connected loc-x0-y1 loc-x0-y0)
    (connected loc-x1-y0 loc-x0-y0)
    (connected loc-x1-y0 loc-x1-y1)
    (connected loc-x1-y1 loc-x0-y1)
    (connected loc-x1-y1 loc-x1-y0))
(:goal
    (and
        (visited loc-x1-y0)
        (visited loc-x1-y1)))
)
```

(a) Problem (in PDDL)



(b) Grounded graph



(c) Lifted graph

Background on Planning

- The data set is based on tasks in International Planning Competitions.
- The goal of cost-optimal planning is to produce provably cost-optimal solutions within a time limit.
- There exists many cost-optimal planners. Which one to use?
- To answer the question, we set a portfolio of 17 planners, construct a graph for each planning problem, and apply graph classification.
- We construct two graph versions:
  - Problem Description Graph (called grounded representation)
  - Abstract Structure Graph (called lifted representation)
- We use the problems in IPC 2018 as the test set and those in prior-year competitions as the training/validation set.

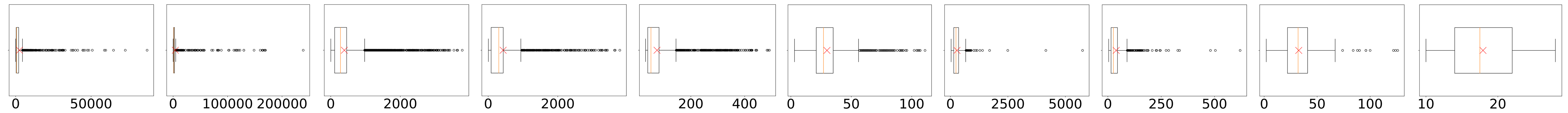
Example Use

- Multilabel classification: For each planning task (graph), predict which planners (targets) solve the task within time limit.
- Refined problem: Because only one planner is needed, it suffices to choose the one with the lowest probability of timeout.
- Accuracy results. Left: using the split provided by the data set. Right: using random train/val re-split on IPC-lifted.

Method	Grounded	Lifted
CNN	73.1%	86.9%
GCN	80.7%	87.6%
GG-NN	77.9%	81.4%

Method	Domain Splits	Random Splits
CNN	82.1% (6.6%)	86.1% (5.5%)
GCN	85.6% (5.5%)	87.2% (3.5%)
GG-NN	76.6% (5.8%)	74.4% (2.7%)

Graph Size Distribution



(a) IPC-grounded (b) IPC-lifted (c) REDDIT-MULTI-12k (d) REDDIT-BINARY (e) COLLAB (f) NCI1 (g) DD (h) PROTEINS (i) ENZYMES (j) MUTAG

	IPC-grounded	IPC-lifted	REDDIT-MULTI-12k	REDDIT-BINARY	COLLAB	NCI1	DD	PROTEINS	ENZYMES	MUTAG
Type	directed	DAG	undirected	undirected	undirected	undirected	undirected	undirected	undirected	undirected
#Graphs	2,439	2,439	11,929	2,000	5,000	4,110	1,178	1,113	600	188
Total #Nodes	6,233,856	9,816,948	4,669,116	859,254	372,474	122,747	334,925	43,471	19,580	3,371
Max #Nodes	87,140	238,909	3,782	3,782	492	111	5,748	620	126	28
Mean (Std) #Nodes	2555.9 (6099.0)	4025.0 (14507.6)	391.4 (428.7)	429.6 (554.1)	74.5 (62.3)	29.9 (13.6)	106.5 (284.3)	39.1 (45.8)	32.6 (15.3)	18.0 (4.6)
Mean (Std) Ave Degree	12.3 (131.0)	2.9 (35.1)	4.7 (27.6)	4.6 (41.3)	132.0 (158.5)	4.3 (1.6)	10.1 (3.4)	7.5 (2.3)	7.6 (2.3)	4.4 (1.5)
Mean (Std) #CC	1.09 (0.61)	1.14 (0.49)	2.81 (2.65)	2.48 (2.47)	1 (0)	1.19 (0.57)	1.02 (0.18)	1.08 (0.52)	1.24 (3.61)	1 (0)
Mean (Std) Diameter	8.2 (2.3)	17.1 (1.5)	10.9 (3.1)	9.7 (3.1)	1.9 (0.3)	13.3 (5.1)	19.9 (7.7)	11.6 (7.9)	10.9 (4.8)	8.2 (1.8)