REGAL: Transfer Learning For Fast Optimization of Computation Graphs

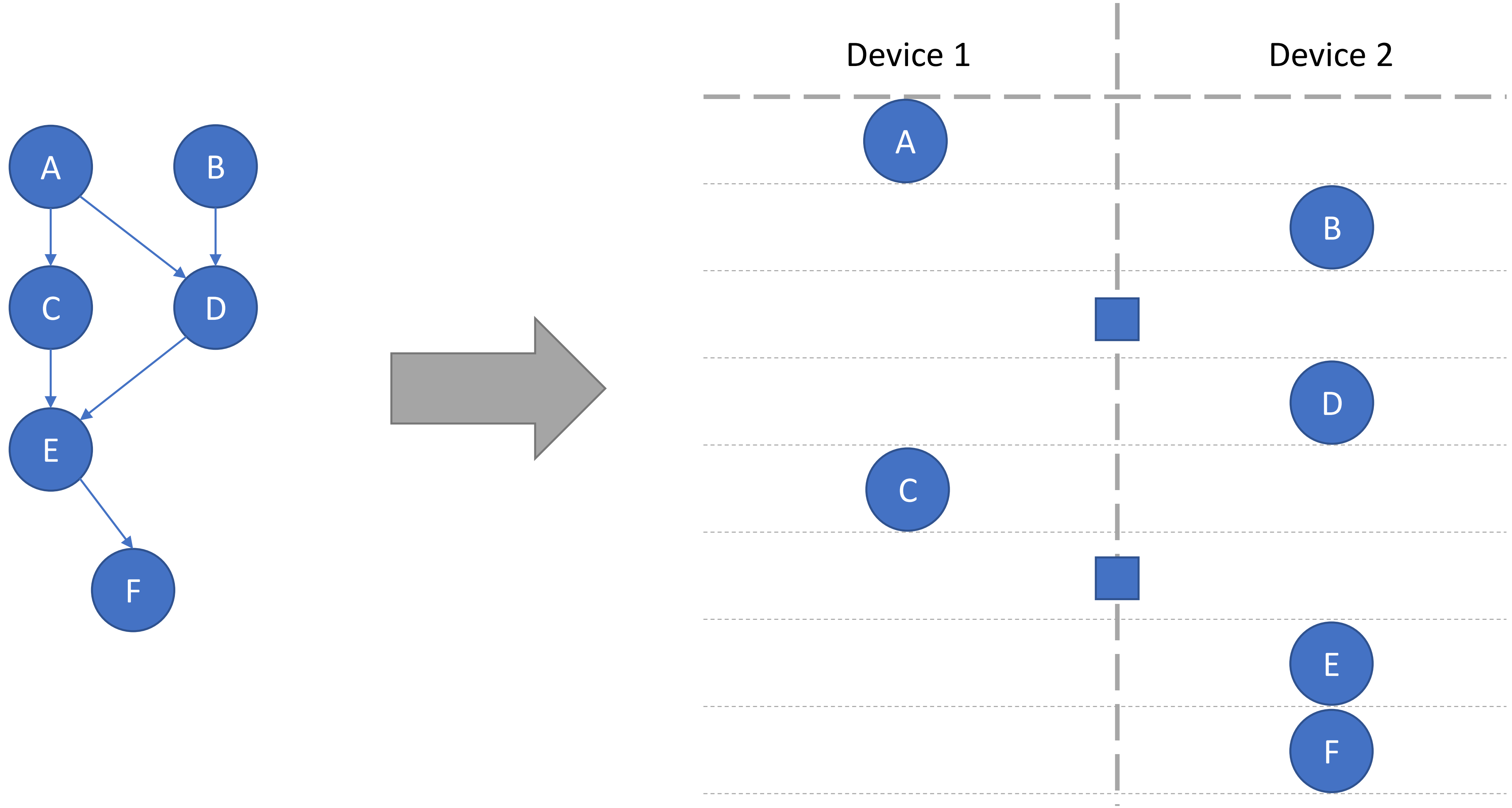
Paliwal et al.

Review by Ross Tooley

Contents

Motivation

Scheduling data-flow graphs



Aim

- Minimise **Peak Memory**

Also consider

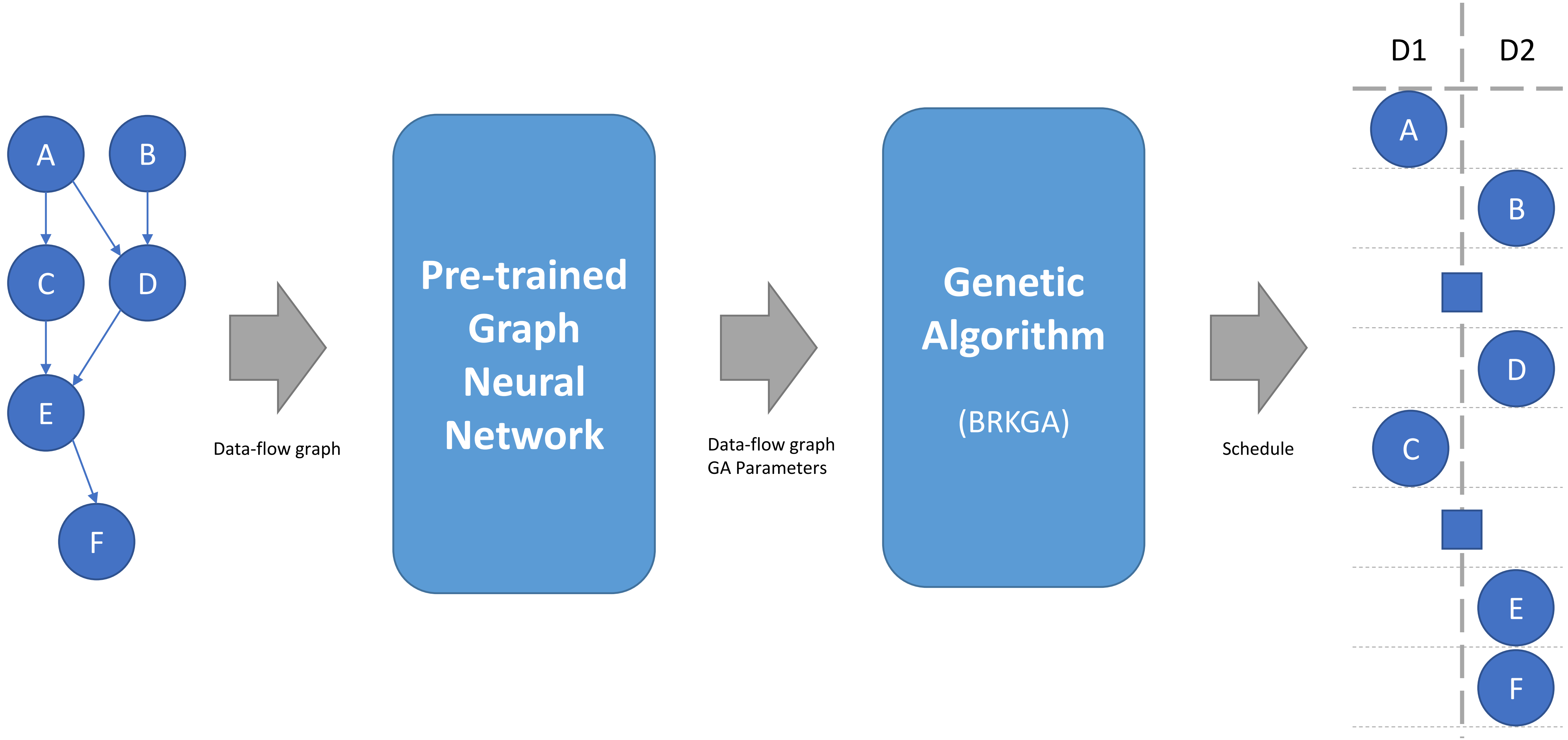
- Scheduler running time

Model Simplifications

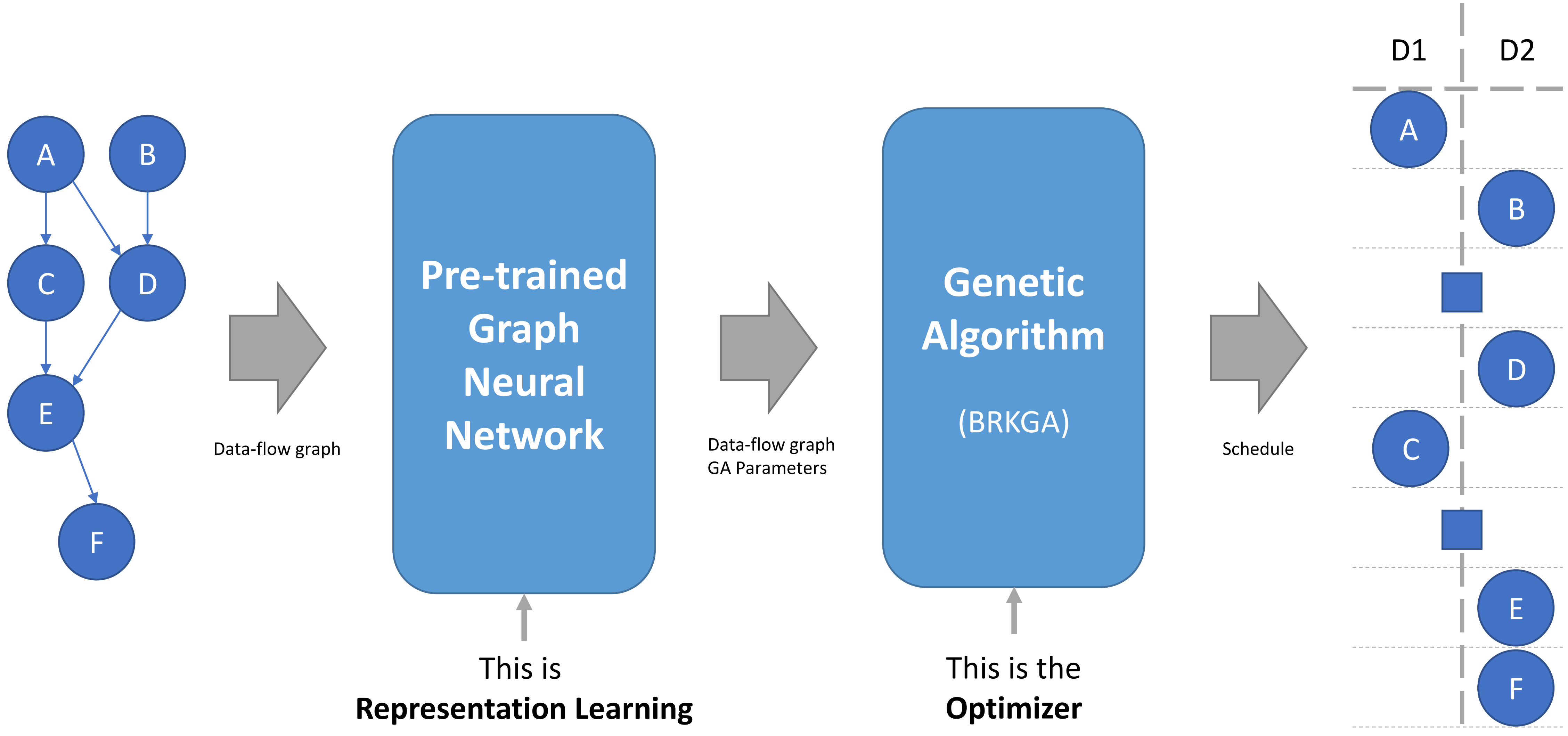
- Discrete, equal time steps

Method

Scheduling pipeline

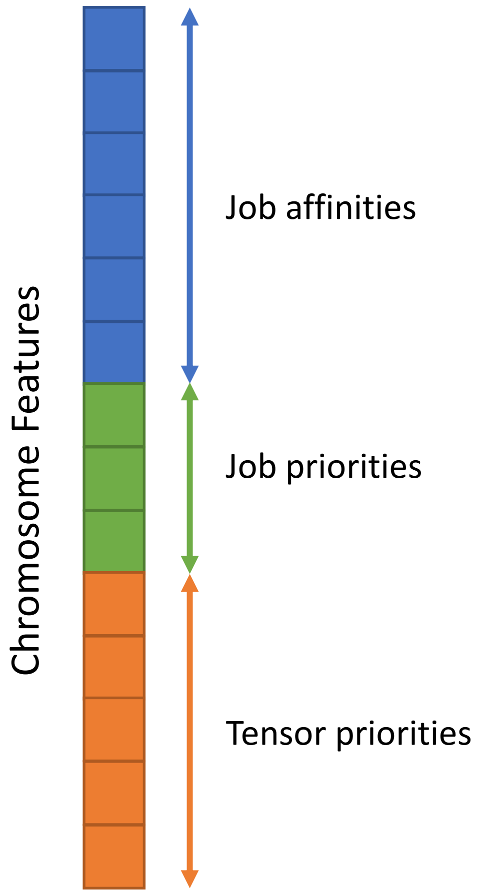


Scheduling pipeline



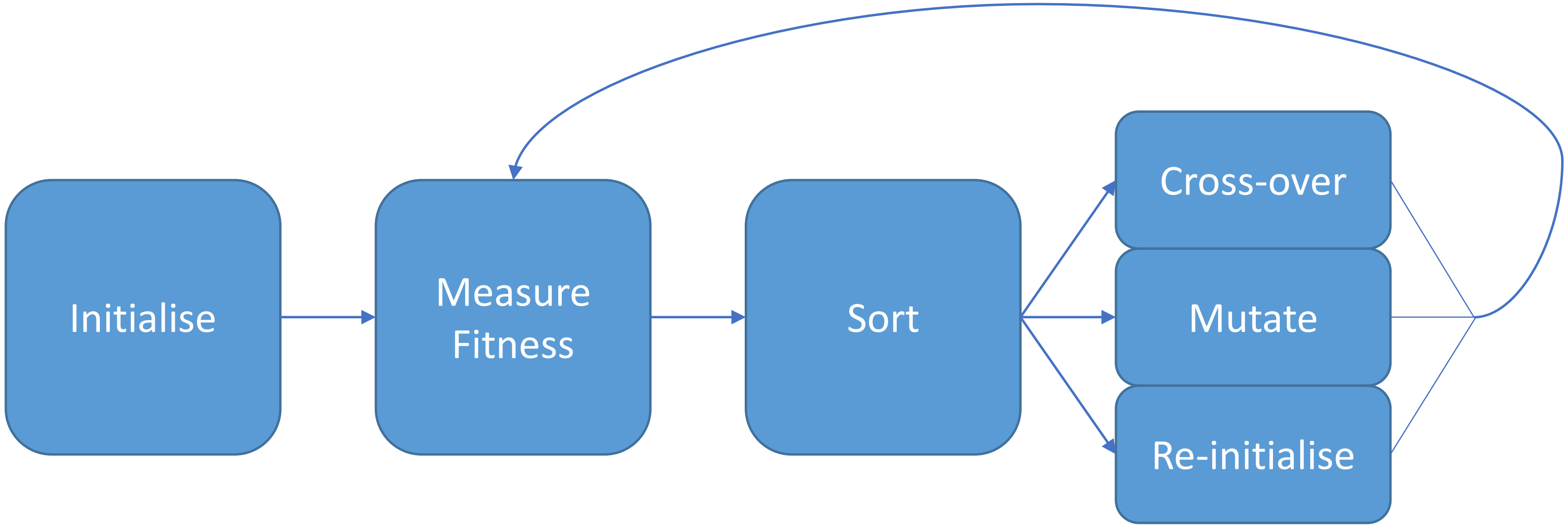
Genetic Algorithms

Chromosomes

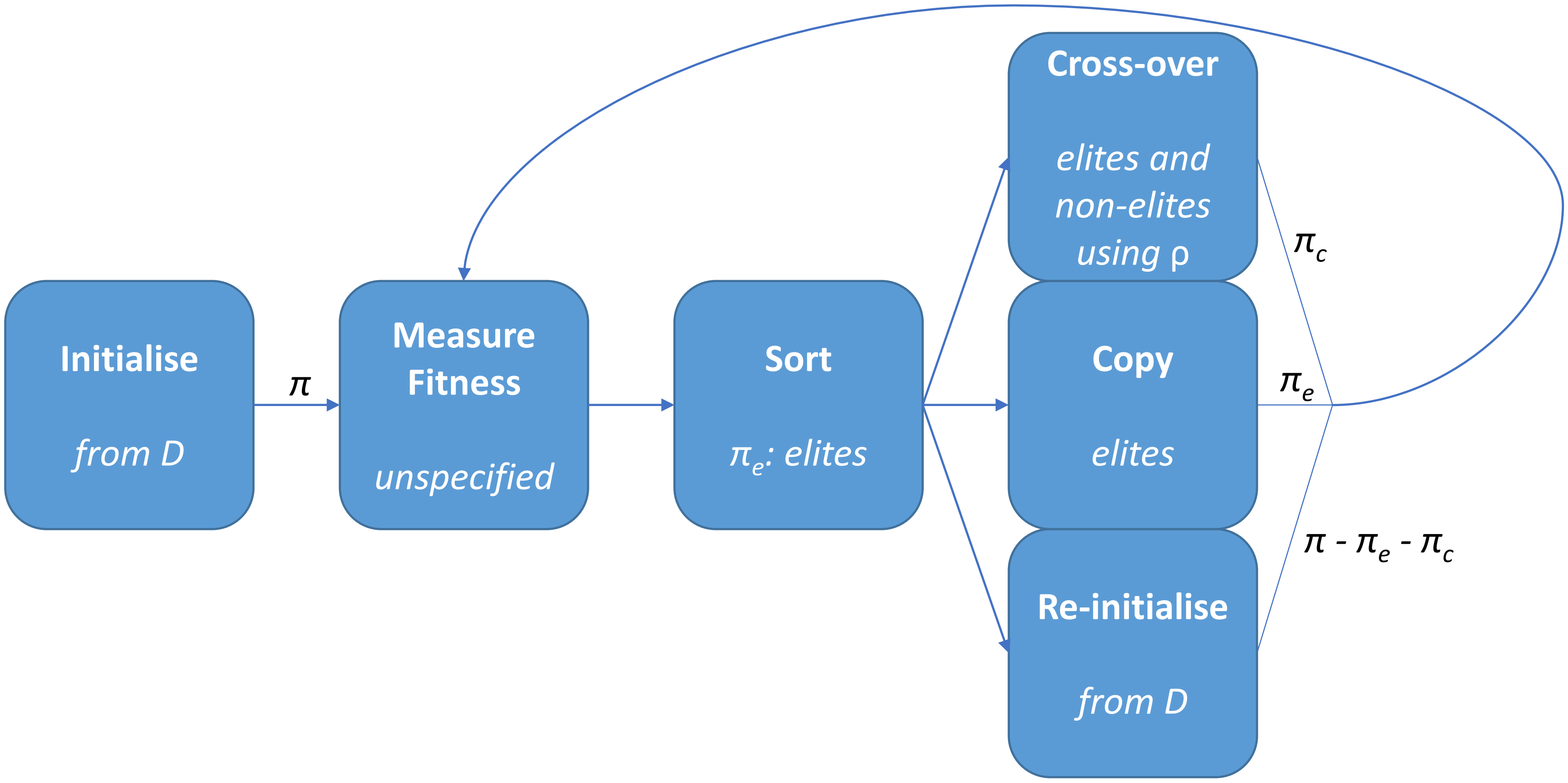


1. Topological sort on data-flow graph
2. Use affinities to assign machines
3. Use priorities to schedule on the machine

Genetic Algorithms

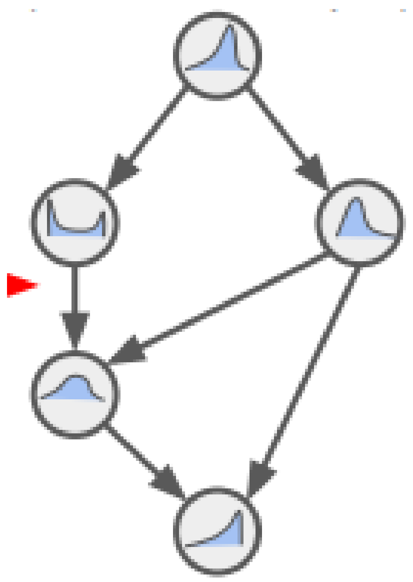


Biased Random Key (BRKGA)



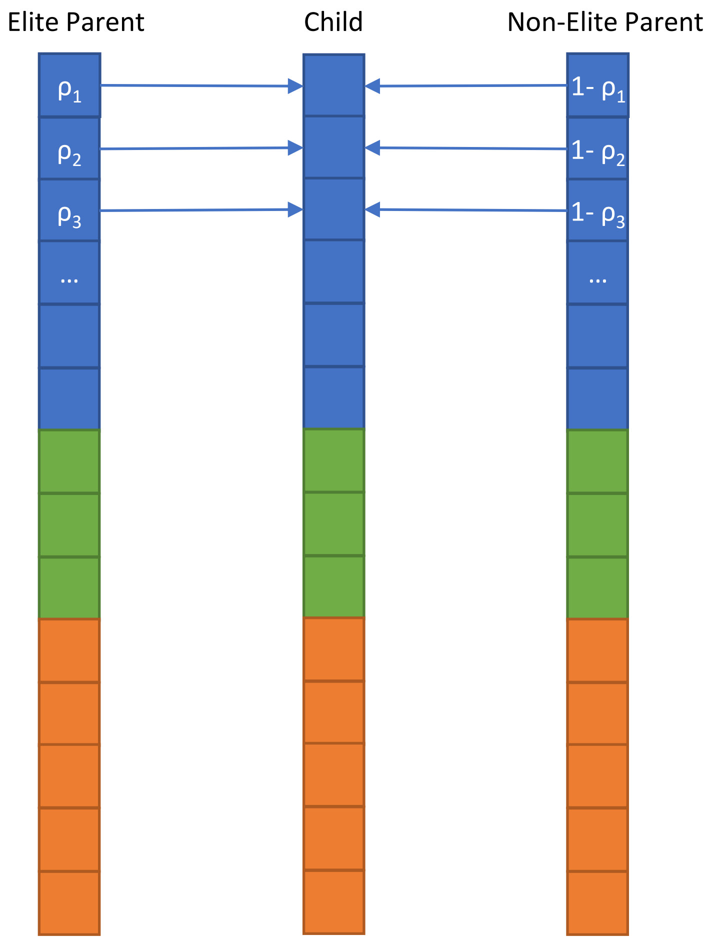
BRKGA has two 'per-node' parameters:

D: per-node beta-distribution



Paliwal, A., Gimeno, F., Nair, V., Li, Y., Lubin, M., Kohli, P., & Vinyals, O. (2019). Regal: Transfer learning for fast optimization of computation graphs. arXiv preprint arXiv:1905.02494.

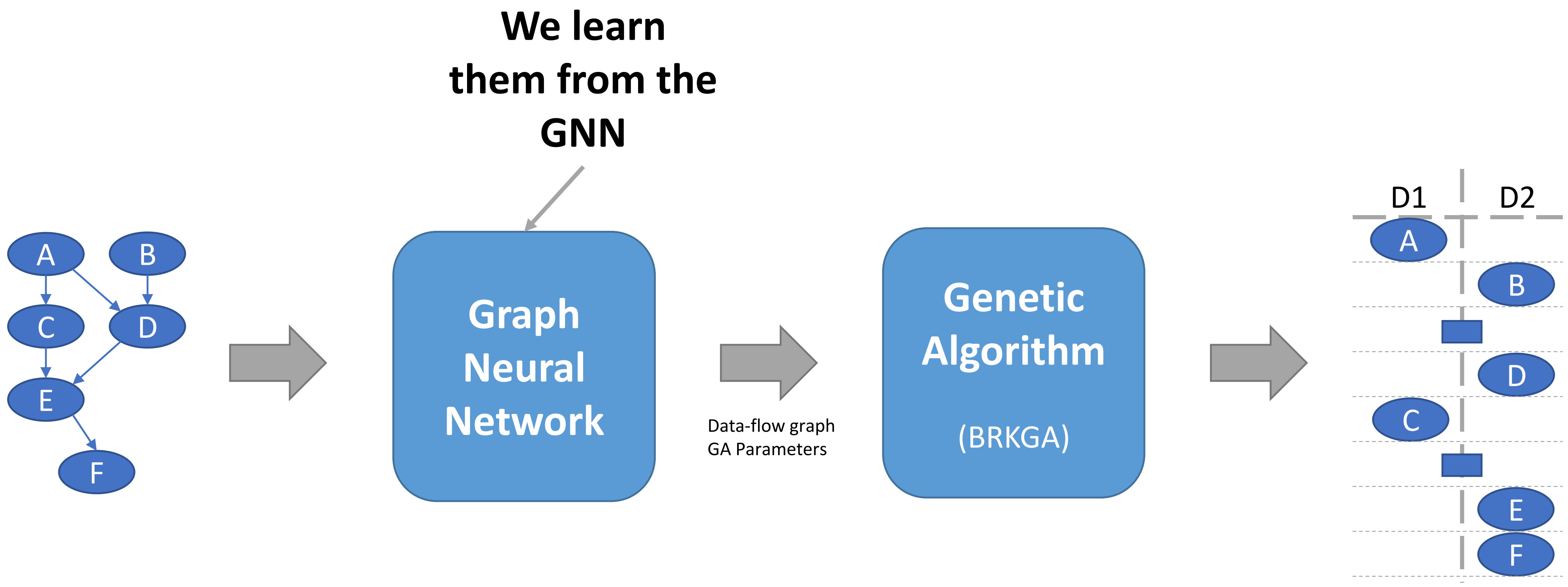
ρ: per-feature probability



BRKGA has two ‘per-node’ parameters:

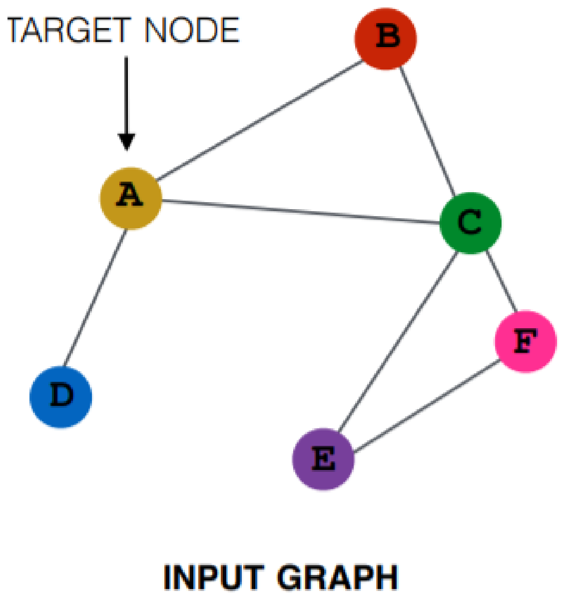
D: per-node beta-distribution

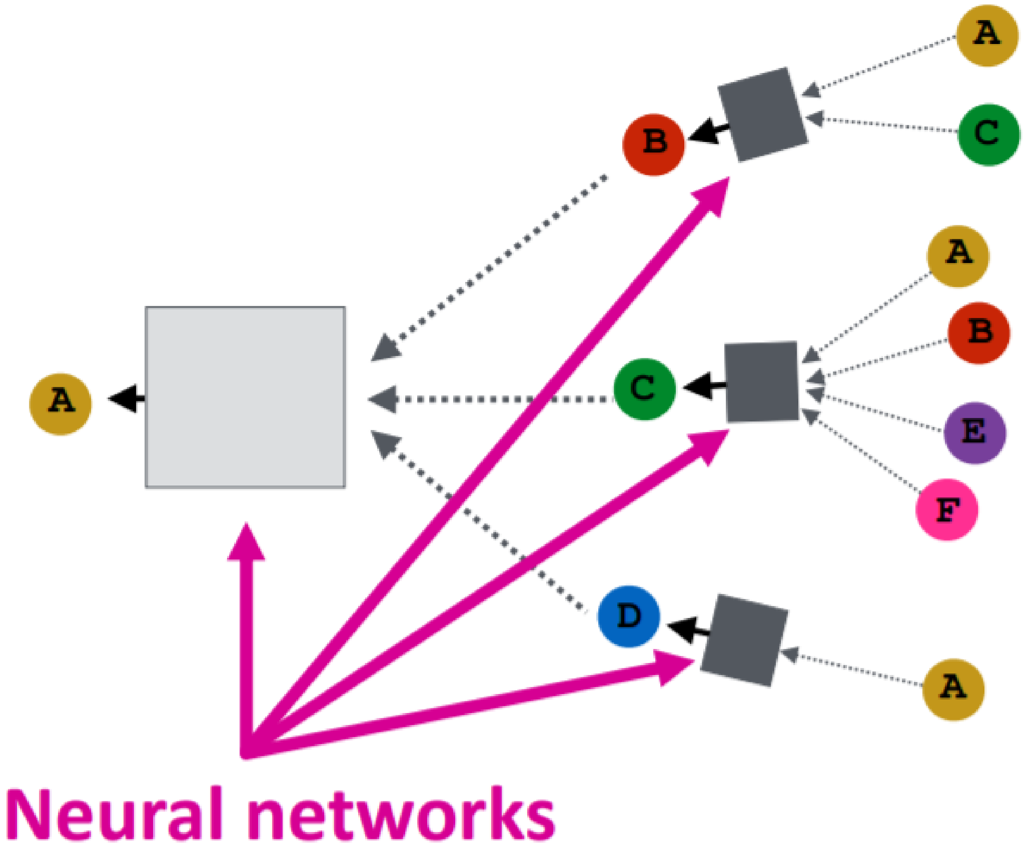
ρ: per-feature probability



Graph Neural Networks

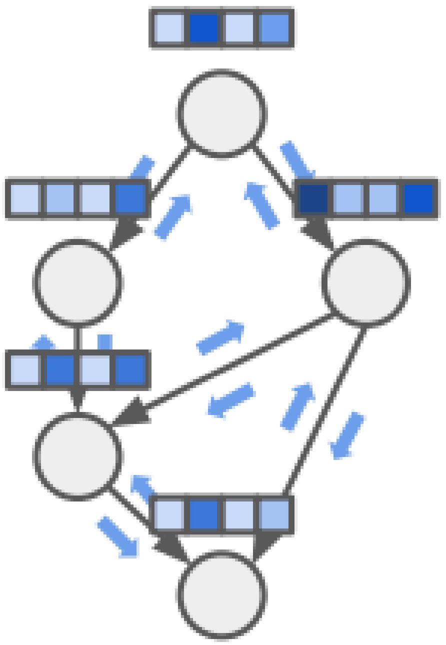
Graph Neural Networks





Medium: https://medium.com/neuralspace/graphs-neural-networks-in-nlp-dc475eb089de

REGAL

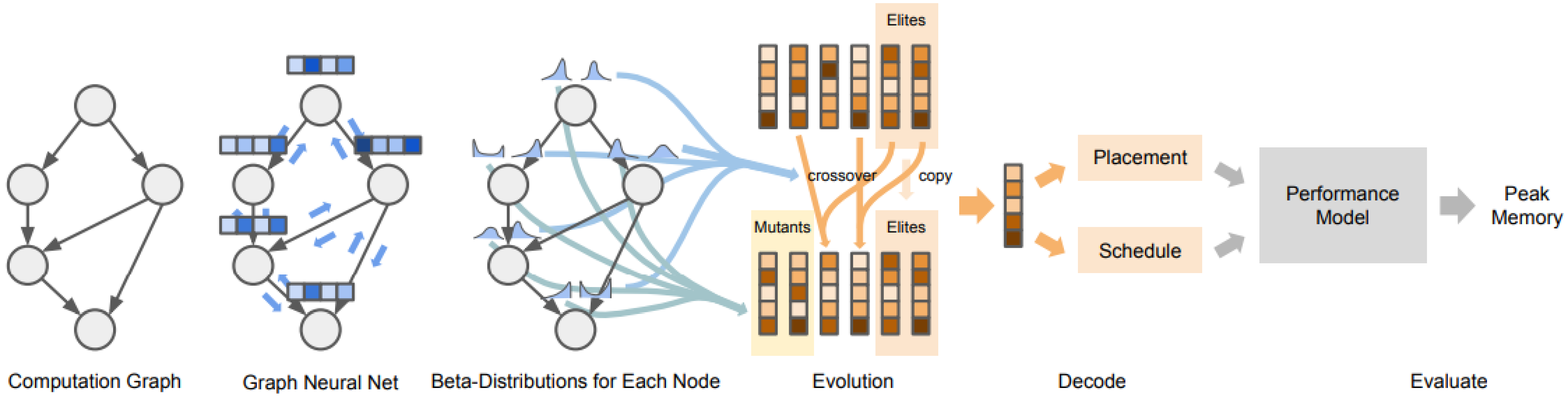


- Accumulates an action vector **y** at each node
- Action vectors map to D and ρ

- REINFORCE-based learning
- Using Peak Memory as reward function

Summary

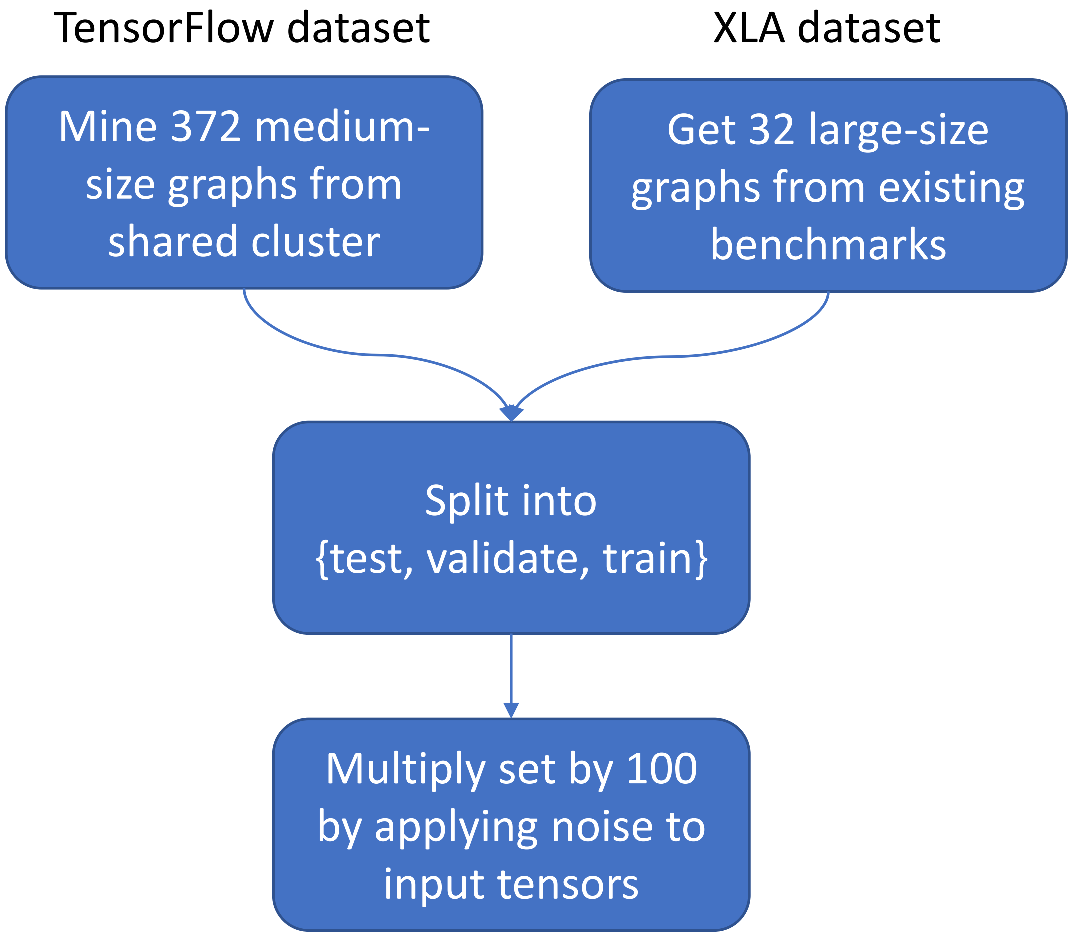
Scheduling Pipeline



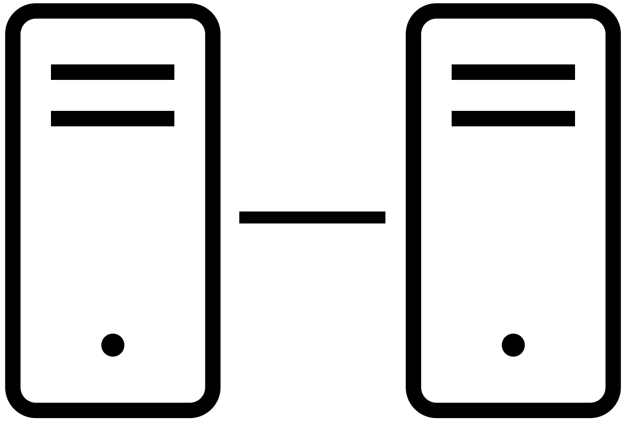
Paliwal, A., Gimeno, F., Nair, V., Li, Y., Lubin, M., Kohli, P., & Vinyals, O. (2019). Regal: Transfer learning for fast optimization of computation graphs. arXiv preprint arXiv:1905.02494.

Results

The dataset?



The cluster?



Peak Memory Results

Algorithm	TensorFlow dataset (test)		XLA dataset	
	% Improv. over BRKGA5K	% Gap from best	% Improv. over BRKGA5K	% Gap from best
CP SAT	-1.77%	13.89%	-47.14%	71.35%
GP + DFS	-6.51%	16.63%	-21.43%	39.86%
Local Search	0.63%	8.65%	-6.69%	21.98%
BRKGA 5K	0%	9.65%	0%	14.04%
Tuned BRKGA	0.8%	8.54%	0.452%	13.52%
GAS	0.16%	9.33%	-1.1%	15.36%
REGAL	**3.56%**	**4.44%**	**3.74%**	**9.40%**

Paliwal, A., Gimeno, F., Nair, V., Li, Y., Lubin, M., Kohli, P., & Vinyals, O. (2019). Regal: Transfer learning for fast optimization of computation graphs. arXiv preprint arXiv:1905.02494.

Scheduler Running Time Results

Table 2: Average running times for all methods.

Algorithm	TensorFlow dataset (test)	XLA dataset
CP SAT	~2 hours	12+ hours
GP + DFS	144 sec	500 sec
Local Search	122 sec	1343 sec
BRKGA 5K	0.89 sec	8.82 sec
Tuned BRKGA	1.04 sec	10.0 sec
GAS	1.04 sec	10.1 sec
REGAL	1.04 sec	10.1 sec

Paliwal, A., Gimeno, F., Nair, V., Li, Y., Lubin, M., Kohli, P., & Vinyals, O. (2019). Regal: Transfer learning for fast optimization of computation graphs. arXiv preprint arXiv:1905.02494.

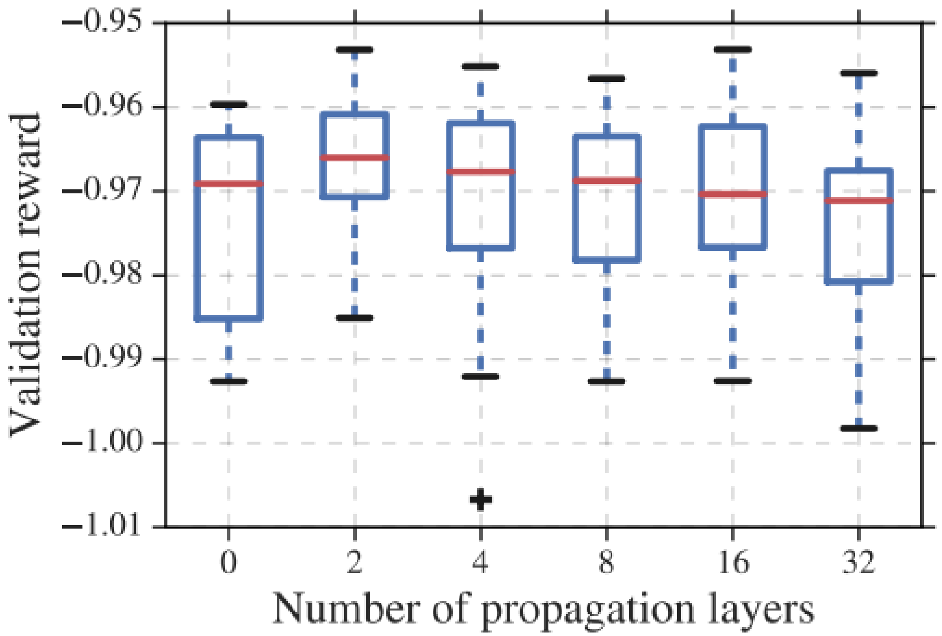
Discussion

Comparison to previous papers’ schedulers

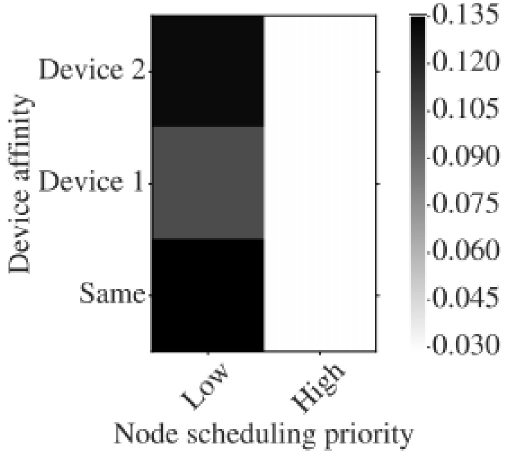
- Uses static scheduling, does not affect data-flow graph
- Optimises Peak Memory rather than Computation Time
- Not tailored towards machine type
- Only evaluated over 2 machines

Representation Learning?

Did REGAL utilise graph structure?



Did REGAL learn a representation of the graph?



Avg Job Memory per Action Bias

Can REGAL be generalised to other metrics?

✓ GNN action vectors and BRGKA chromosomes are metric-independent

✗ The scheduling model depends on discrete, equal time steps

? The learned representations would change!

Closing remarks

- Use of GNN significantly improves BRGKA
- With low overhead

- Learning representations is useful for explanations

- Evaluation only considers 2 machines
- REGAL is complicated!

Thank you for listening

Q&A?