REGAL: Transfer Learning For Fast Optimization of Computation Graphs

Paliwal et al.

Review by Ross Tooley
Contents

1. Motivation
2. Methods
   1. Genetic Algorithms
   2. Graph Neural Networks
3. Results
4. Discussion
Motivation
Scheduling data-flow graphs
Aim

• Minimise Peak Memory

Also consider

• Scheduler running time

Model Simplifications

• Discrete, equal time steps
Method
Scheduling pipeline

Pre-trained Graph Neural Network

Data-flow graph

Genetic Algorithm (BRKGA)

Data-flow graph

GA Parameters

Schedule
Scheduling pipeline

Data-flow graph

Pre-trained Graph Neural Network

This is Representation Learning

Data-flow graph

Genetic Algorithm (BRKGA)

This is the Optimizer

Schedule
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

- Initialisation
- Measure Fitness
- Sort
- Cross-over
- Mutate
- Re-initialise
Biased Random Key (BRKGA)

Initialise from D

Measure Fitness unspecified

Sort \( \pi_e : \text{elites} \)

Copy elites

Re-initialise from D

Cross-over elites and non-elites using \( \rho \)

\( \pi_e \)

\( \pi_c \)

\( \pi - \pi_e - \pi_c \)
BRKGA has two ‘per-node’ parameters:

D: per-node beta-distribution

\[ \rho: \text{per-feature probability} \]

\[ \rho_1 \quad \rho_2 \quad \rho_3 \quad \ldots \]

Elite Parent  Child  Non-Elite Parent

BRKGA has two ‘per-node’ parameters:

D: per-node beta-distribution

\[ \rho: \text{per-feature probability} \]

We learn them from the GNN.
Graph Neural Networks

REGAL

• Accumulates an action vector $y$ at each node
• Action vectors map to $D$ and $\rho$
• REINFORCE-based learning
• Using Peak Memory as reward function
Summary
Scheduling Pipeline

Results
The dataset?

TensorFlow dataset
- Mine 372 medium-size graphs from shared cluster
- Split into \{test, validate, train\}
- Multiply set by 100 by applying noise to input tensors

XLA dataset
- Get 32 large-size graphs from existing benchmarks

The cluster?
Peak Memory Results

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>TensorFlow dataset (test)</th>
<th>XLA dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>% Improv.</td>
<td>% Gap</td>
</tr>
<tr>
<td></td>
<td>over BRKGA5K</td>
<td>from best</td>
</tr>
<tr>
<td>CP SAT</td>
<td>-1.77%</td>
<td>13.89%</td>
</tr>
<tr>
<td>GP + DFS</td>
<td>-6.51%</td>
<td>16.63%</td>
</tr>
<tr>
<td>Local Search</td>
<td>0.63%</td>
<td>8.65%</td>
</tr>
<tr>
<td>BRKGA 5K</td>
<td>0%</td>
<td>9.65%</td>
</tr>
<tr>
<td>Tuned BRKGA</td>
<td>0.8%</td>
<td>8.54%</td>
</tr>
<tr>
<td>GAS</td>
<td>0.16%</td>
<td>9.33%</td>
</tr>
<tr>
<td>REGAL</td>
<td><strong>3.56%</strong></td>
<td><strong>4.44%</strong></td>
</tr>
</tbody>
</table>

# Scheduler Running Time Results

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>TensorFlow dataset (test)</th>
<th>XLA dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>CP SAT</td>
<td>~2 hours</td>
<td>12+ hours</td>
</tr>
<tr>
<td>GP + DFS</td>
<td>144 sec</td>
<td>500 sec</td>
</tr>
<tr>
<td>Local Search</td>
<td>122 sec</td>
<td>1343 sec</td>
</tr>
<tr>
<td>BRKGA 5K</td>
<td>0.89 sec</td>
<td>8.82 sec</td>
</tr>
<tr>
<td>Tuned BRKGA</td>
<td>1.04 sec</td>
<td>10.0 sec</td>
</tr>
<tr>
<td>GAS</td>
<td>1.04 sec</td>
<td>10.1 sec</td>
</tr>
<tr>
<td>REGAL</td>
<td>1.04 sec</td>
<td>10.1 sec</td>
</tr>
</tbody>
</table>

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?