Diplomarbeitspräsentation

Algorithm Selection for the Graph Coloring Problem

Martin Schwengerer

Motivation

- Some problems (called NP-hard problems) cannot be solved efficiently
- Focus on heuristic algorithms
- But: None of them is perfect on all problems (known as “No Free Lunch” theorem [1])

Problem: Which heuristic should be used?

One approach: Select always the algorithm from which we expect the best performance.

Example Problem: GCP

- The Graph Coloring Problem (GCP) is a well-known NP-hard problem
- Input: Graph \( G = (V, E) \)
- Objective: assign each node a color such that
  - no adjacent nodes have the same color
  - the total number of colors is minimized.

There exist different heuristic approaches for GCP like tabu search, simulated annealing, genetic algorithms, ant colony optimization, ...

Our Approach:

Use Machine Learning techniques for automated algorithm selection for the GCP!

1. Identify characteristic features of a graph
2. Evaluate the performance of several state-of-the-art solvers for the GCP
3. Train classification algorithms to predict the best algorithm for a new GCP instance.

Step 1: Identify Instance Features

We identified 78 features of a GCP instance that can be calculated in polynomial time based on:

- Graph Size
- Node degree
- Clustering Coefficient
- Clique Size
- Greedy Coloring Algorithms
- Local Search Attributes
- Lower- and upper bounds
- Tree Decomposition

Step 2: Evaluation of Several State-Of-The-Art Heuristics

We tested 7 heuristic algorithms:

- Foo-PartialCol (FPC),
- Hybrid Evolutionary Algorithm (HEA),
- Iterated Local Search (ILS),
- Multi-Agent Fusion Search (MAFS),
- MACOL,
- MMT, and
- TABUCOL (TABU) on 3 public available instance sets

1265 graphs
Total runtime: roughly 90.000 hours

Results:
- no heuristics dominates all others on all graphs
- some heuristics show better performance on graphs with certain features

Step 3: Training Phase

- Trained 6 well-known classification algorithms:
  - k-nearest Neighbor (KNN),
  - C4.5 Decision Trees (C4.5),
  - Bayes Networks (BN),
  - Multi-Layer Perceptrons (MLP),
  - Random Forest (RF), and
  - Support-Vector Machines (SVM),
- Different discretization methods (MDL, KON),
- Analyzed the impact of using a subset of heuristics on the overall quality of the prediction.

Evaluation & Results

We compared our automated algorithm selection solvers with the existing solvers on 152 new generated instances:

Number of best solutions obtained by the different heuristics. The red bar denotes that this algorithm achieved the highest number within the subset of instances.

Results:
- Classification algorithms predicts for up to 70.39% of the graphs the most suited algorithm
- Improvement of \(+33.55\%\) compared with the best solver

References