A Java Framework for (Online) Algorithm Selection

With Use Case on the Generalized Assignment Problem

Hans Degroote[1], José Luis González Velarde[2] and Patrick De Causmaecker[1]
[1]KU Leuven Department of Computer Science - CODeS & imec-ITEC
[2]Tecnologico de Monterrey, Mexico

Overview

- **Object oriented system for algorithm selection**
  - ➢ Algorithms automatically selected and executed
- All core components of algorithm selection are explicitly present
  - ➢ User chooses concrete implementation for specific problem scenario
- **Minimal user-input required**
  - ➢ Executables + scripts for extracting performance and feature values
- Applied to the **Generalized Assignment Problem**

Elements of Algorithm Selection

Core elements

-Algorithm space **A**

-Instance space **I**

a distribution over I: ***D***

-Performance mapping **p: I x A -> R**

-Selection mapping **λ: I -> A**

Helper elements

-Feature space **F**

=> selection mapping: **λ: (I ->) F -> A**

-Training data **H:** set of tuples **(i,φ_i, a, p(a,i))**

-Selection mapping init strategy: **β: H -> Λ**

=> e.g. linear regression, decision tree, K-NN

Note: all this is for **deterministic performance**

What can the system be used for?

- **Executing algorithms** and keeping track of results
  - ➢ Executables are automatically called
  - ➢ Performance is extracted and added to database
- Standard one-shot **offline algorithm selection**
  - ➢ Uses WEKA or user-defined ML methods
- **Online algorithm selection**
  - ➢ Process new data to improve selection mapping
- Starting from zero (no training data)
  - ➢ Active learning?

Idea for future: human in the loop

- Identify instance regions with poor performance
- Identify the cause:
  - ➢ No algorithm performs well (=> develop new)
  - ➢ Features cannot distinguish (=> develop new)
  - ➢ Init strategy is not good enough (=> find better)

Using the System: Workflow

For each algorithm:

Obtain executables

Write script for performance extraction

For each feature:

Write script for feature value extraction

The only user effort

Choose a strategy β for creating selection mappings

=>An interface to WEKA is implemented

Add training data (optional)

Start performing online algorithm selection

```
Algorithm 1 Processing online instances
for Online instance $i$ do
    $\lambda = \beta(H)$ (get selection mapping)
    $f_i$ = Extract required feature values
    Add feature values to database
    $a = \lambda(f_i)$ (select algorithm)
    Run algorithm
    Extract performance from result file
    Add performance to database
```

Use Case: Generalized Assignment Problem

Problem: assign each job to exactly 1 agent, with job-specific resource usage and a maximum resource capacity for each agent

Goal: minimise assignment costs

Applications: scheduling, routing, production planning…

$$\begin{aligned} \text{minimize} \quad & \text{cost}(\sigma) = \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \\ \text{subject to} \quad & \sum_{j \in J} a_{ij} x_{ij} \leqslant b_i \quad \forall i \in I, \\ & \sum_{i \in I} x_{ij} = 1 \quad \forall j \in J, \\ & x_{ij} \in \{0,1\} \quad \forall i \in I \text{ and } \forall j \in J. \end{aligned}$$

J: *jobs*
I: *agents*
c_{ij}: *assignment-cost matrix*
a_{ij}: *resource-usage matrix*
b_i: *capacity constraints*
x_{ij}: *assignment matrix*
σ: *assignment*

Taken from [1]

Why GAP as use case?

=> It often must be solved repeatedly in limited time

- ➢ Models problems at the Operational level
- ➢ Occurs as a subroutine when solving bigger problems

Challenges

- Obtaining executables (and getting them to work)
- Identifying good features
- Deciding how to measure performance
  - ➢ Specific application must be taken into account

Contact
Hans Degroote
KU Leuven Department of CS
Email: Hans.Degroote@kuleuven.be

References

[1] *Yagiura, Mutsunori, Toshihide Ibaraki, and Fred Glover. "A path relinking approach with ejection chains for the generalized assignment problem." European journal of operational research 169.2 (2006): 548-569.*

[2] *Laguna, M., Kelly, J. P., González-Velarde, J., & Glover, F. (1995). Tabu search for the multilevel generalized assignment problem. European journal of operational research, 82(1), 176-189.*

[3] *Souza, Danilo S., Haroldo G. Santos, and Igor M. Coelho. "A Hybrid Heuristic in GPU-CPU Based on Scatter Search for the Generalized Assignment Problem." Procedia Computer Science 108 (2017): 1404-1413.*

[4] *Alfandari, Laurent, Agnes Plateau, and Pierre Tolla. "A two-phase path relinking algorithm for the generalized assignment problem." Proceedings of the Fourth Metaheuristics International Conference. 2001.*

comex
combinatorial optimization:
metaheuristics & exact methods