Per-class algorithm selection for black-box optimisation

Koen van der Blom and Carola Doerr

Sorbonne Université

2023-11-28
Summary

• Algorithm selection (AS) is great, e.g., for SAT → Mostly per-instance

• Black-box optimisation → No a priori instance-specific information

• Per-class AS → Use problem properties for AS over sets of instances
Summary

• General data-driven AS framework
  → Outperform hand-made SOTA system by Meta (NGOpt)
• Extensions to grey-box problems possible, to use more information

Per-class algorithm selection for black-box optimisation
This talk

• Why we care about per-class algorithm selection

• How data-driven per-class algorithm selection works

• Comparison to hand-designed system

• Discussion of generality and extensibility
Algorithm selection [Rice, Adv. in Comp. 1976]

- No single algorithm is the best for every problem
Per-instance algorithm selection

Feature Extractor

Training Instances

Run on

Algorithms

Performance data

Selector

New Instance

Input

Input

Instance specific Feature data

Predicted Algorithm

Output

Input
Black-box optimisation

• Optimise a problem with unknown internals

• Use sampling strategy to find the optimum
  • Evolutionary algorithms
  • Bayesian optimisation
  • ...

• No a priori instance specific $\rightarrow$ Costs function evaluations to compute
  • Description
  • Features
Per-class algorithm selection

Problem description

Training Instances

Run on

Algorithms

Performance data

Selector

New Instance

Input

High-level problem properties

Includes

Output

Predicted Algorithm

Includes

Problem description

Includes

High-level problem properties

Includes

Run on
Properties vs. features

• **Features**
  - Describe a problem instance
  - Differ across instances
  - Require function evaluations to determine

• **Properties**
  - Describe the higher level problem class or domain
  - Same across instances (in the same class/domain)
  - Known a priori
Pros and cons to per-class algorithm selection

• No per-instance selection, but for larger groups

• Pro
  • Free – Can make a decision before doing any evaluations
  • General – Don’t have to design features for each problem variant

• Con
  • Being less specific will usually mean lower performance
Data-driven selector – Basic idea

- Training Instances
- Run on
- Algorithms
- Performance data
- Selector
- Includes
- High-level problem properties
- Aggregated ranking
- Defines
- Problem description

Per-class algorithm selection for black-box optimisation
Considered problem properties

• Dimensions – Number of decision variables
• Budget – Number of function evaluations

• Properties we keep fixed
  • Variable type – Continuous
  • No constraints
  • No noise
Comparison

• NGOpt39
  
  • Hand-designed competence map / selection wizard – “what is good where”
    [Meunier et al., TEVC 2022]

• Data-driven selector
  
  • Choose from the algorithms included in NGOpt39 based on performance data

• Same algorithms + implementation, different selection method
Training

• Budgets: [1, ..., 10000]

• Dimensions: [1, ..., 100]

• Algorithms: All those used by NGOpt39 in this domain (34 total)

• Problems: 24 BBOB functions, instance 1, 25 repetitions each

Ranking

• Per budget-dimension pair
  • Give an algorithm 1 point for each run in the top 25 per problem
  • Runs below the top 25 tied with the 25\textsuperscript{th} best run are also awarded a point
  • Algorithm with most points is chosen for the budget-dimension pair
Data-driven vs. NGOpt39
Testing

• Budgets: [200, 300, 500, 1000, ..., 5000, 6000, ..., 10000]

• Dimensions: [2, 3, 5, 10, ..., 50, 60, ..., 100]

• Algorithms:
  • 5 per budget-dimension pair
  • All those chosen by NGOpt39, or in the top 4 performers (excluding NGOpt39)

• Problems: 828 MA-BBOB functions, 1 repetition each
MA-BBOB – Many affine combinations

Fig. from [Vermetten et al., GECCO 2023]

Our selection of combined BBOB functions

Fig. from [Dietrich and Mersmann, PPSN 2022]
Test ranking

• One run per algorithm per problem

• Per budget-dimension pair
  • Give an algorithm 1 point for each problem where it is the best
  • Algorithms tied with the best algorithm on a problem also get a point
  • Algorithm with most points is chosen for the budget-dimension pair
Data-driven vs. NGOpt39 on MA-BBOB
Wins compared to the VBS on MA-BBOB
Loss to the VBS on MA-BBOB (% of problems)

Data-driven

NGOpt39

Per-class algorithm selection for black-box optimisation
Moving forward

• Improve training instance set for generalisability

• Progressively add algorithms

• Evaluate different selection methods, e.g., for:
  • Worst case performance
Generalisation to other problem domains

Variable type?
- Discrete
  - Single-objective
    - Unconstrained
  - Multi-objective
    - Constrained
- Continuous
- Mixed-variable
Extension to grey-box setting

• Include user provided properties in the data-driven analysis

• Property needs to be known for the training data

• E.g., multimodal (F15-F24 in BBOB)
Take home

• Per-class algorithm selection for black-box optimisation works

• Data-driven system improves over hand-designed selectors

• General conceptual framework $\rightarrow$ No features, no problem
  • Problems with constraints, multiple objectives, ... $\rightarrow$ No problem!

• Extendable to grey-box setting
  • Have a multimodal problem?
    $\rightarrow$ Selection based on multimodal problems in the data
References


