On Adaptive Specialisation in Genetic Improvement

Aymeric Blot  Justyna Petke

University College London

GI@GECCO’19 — 13 July 2019
Genetic Improvement

- Repair
- Runtime
- Memory
- Slimming
- New feature
- Parallelisation
- Energy

Petke et al., IEEE Transactions on Evolutionary Computation, 2018
Genetic Improvement

- Repair
- Memory
- Slimming
- New feature
- Energy
- Parallelisation
- Runtime

Petke et al., IEEE Transactions on Evolutionary Computation, 2018
Manual Software Specialisation

MiniSAT

- MiniSAT for CIT
- MiniSAT for AProVE
- MiniSAT for Ensemble
Adaptive Software Specialisation

Software

Variant for scenario A

Variant for scenario B

Variant for scenario C
In a Nutshell

Assumptions

▶ Specialisation (GI) is very time consuming
▶ Edits can be shared between applications
▶ Edits can be specific to single applications
▶ Input boundaries can be surprising

Proposition

▶ Merge all training inputs
▶ Start with a single variant
▶ Branch on statistical difference in performance
▶ Grow a mapping of software variants
Partitioning Training Data
Evolving a Mapping of Mutated Software
Challenges

Partitioning
- How to detect heterogeneity?
- With which constraints?

Algorithm
- Which search process is the most relevant?
- Will it be competitive?

Training Data
- Will it scale?
- At what trade off?
Final Words

Adaptive Software Specialisation

- No expert knowledge
- No feature identification/computation/selection
- Automated black box

Why?

- Improve time consumption
- Improve final software variants
- Discover new parameters and compilation switches
Selected References

