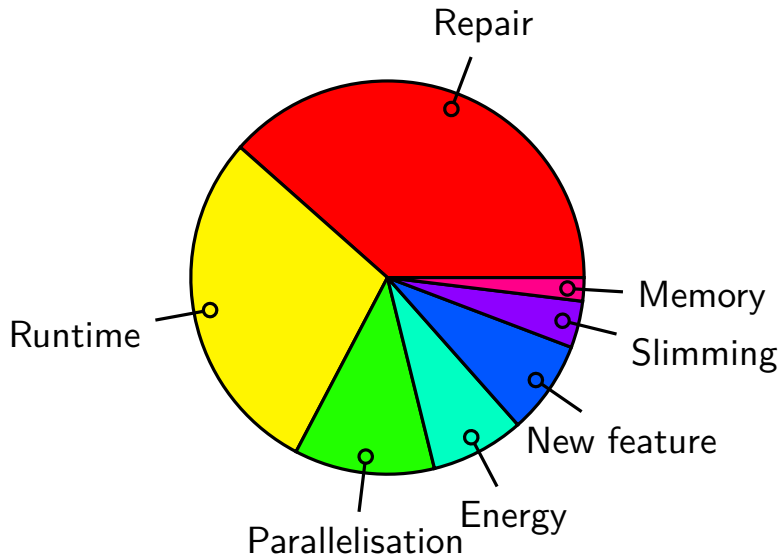# On Adaptive Specialisation in Genetic Improvement

Aymeric Blot   Justyna Petke

University College London
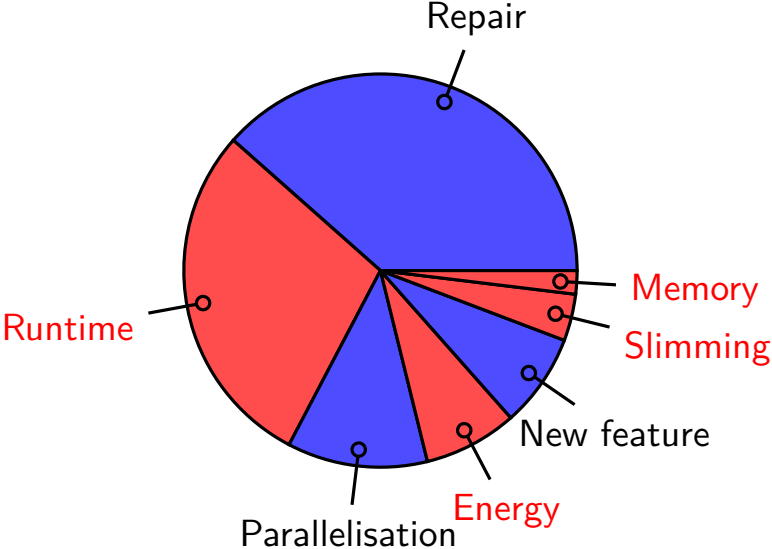
GI@GECCO'19 — 13 July 2019
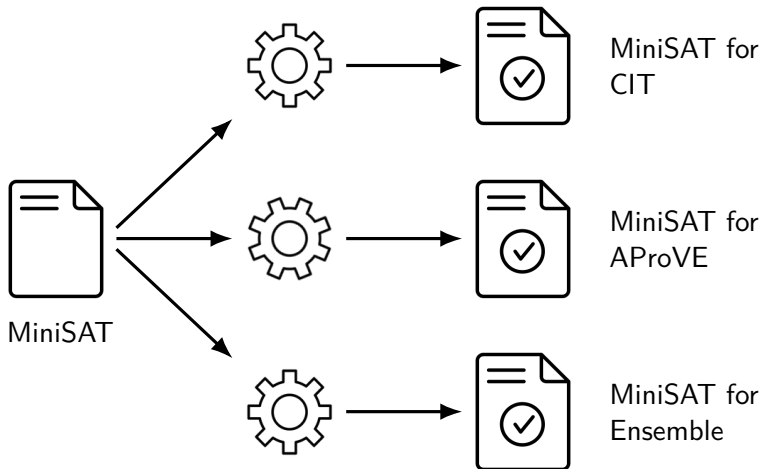
Genetic Improvement

Genetic Improvement

Manual **Software Specialisation**

Petke et al., IEEE Transactions on Software Engineering, 2018

# Adaptive Software Specialisation
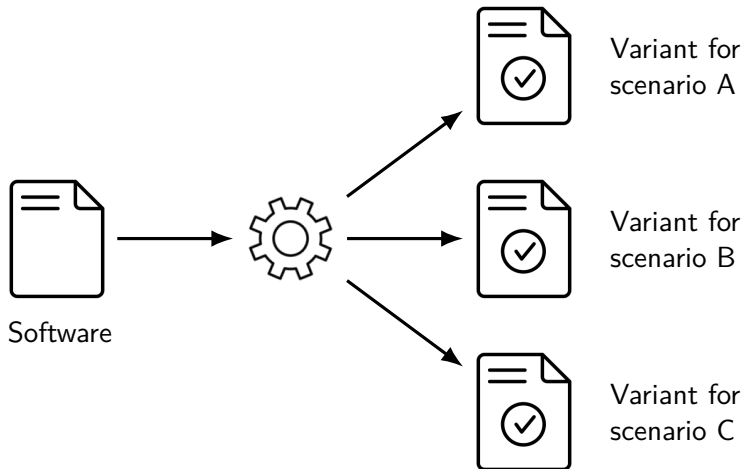
In a Nutshell

Assumptions

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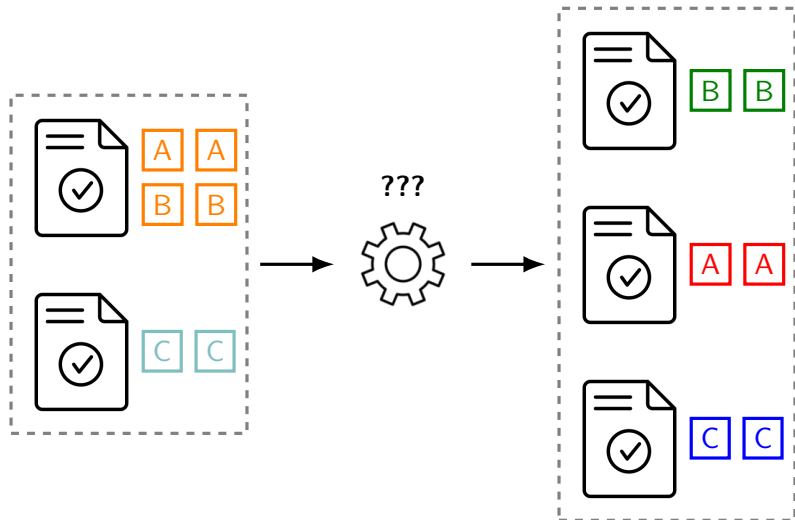- ▶ Input boundaries can be surprising

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
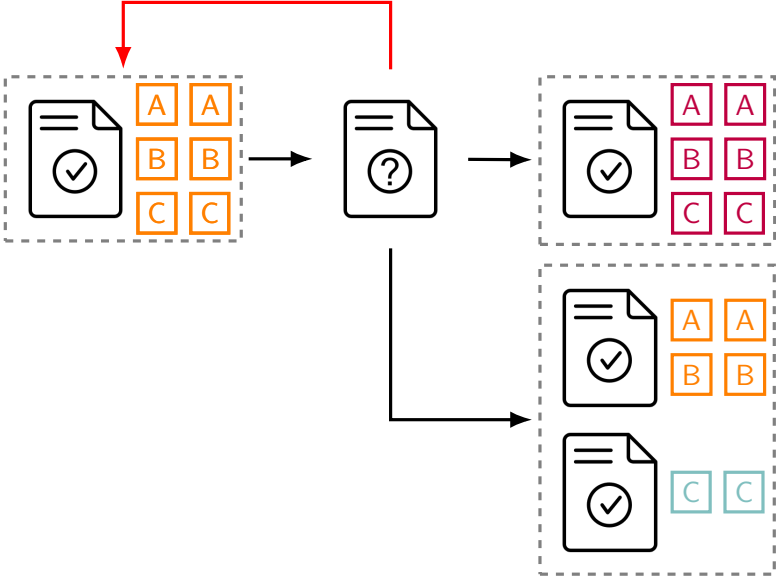
Evolving a Mapping of Mutated Software

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

Justyna Petke, Saemundur O. Haraldsson, Mark Harman, William B. Langdon, David Robert White, and John R. Woodward.
Genetic improvement of software: A comprehensive survey.
IEEE Transactions on Evolutionary Computation, 22(3):415–432, 2018.

Justyna Petke, Mark Harman, William B. Langdon, and Westley Weimer.
Specialising software for different downstream applications using genetic improvement and code transplantation.
IEEE Transactions on Software Engineering, 44(6):574–594, 2018.

Lin Xu, Holger H. Hoos, and Kevin Leyton-Brown.
Hydra: Automatically configuring algorithms for portfolio-based selection.
In *AAAI Conference on Artificial Intelligence (AAAI 2010)*, pages 210–216. AAAI Press, 2010.