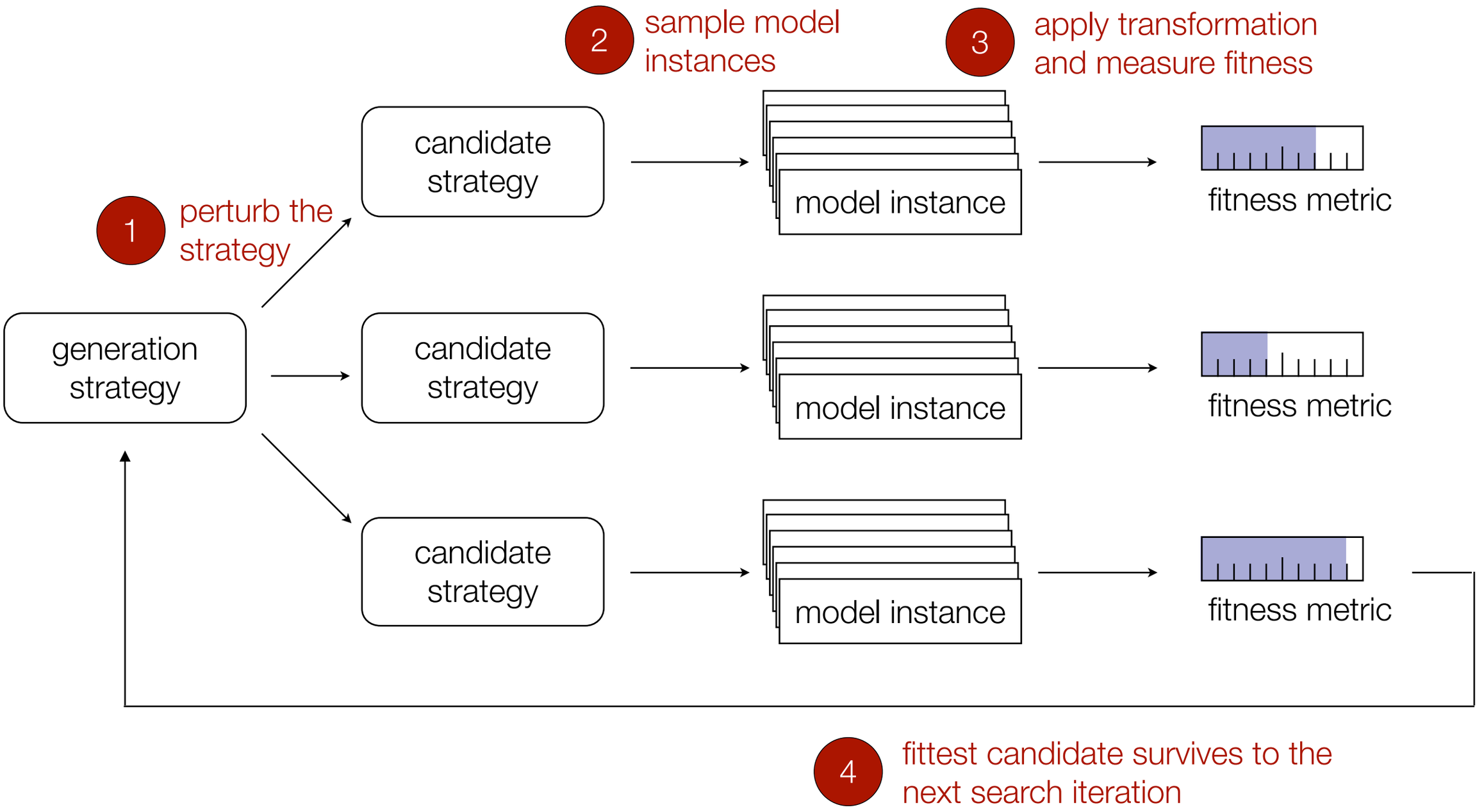
Using Mutation to Guide the Optimisation of Model Transformation Testing Strategies

Simon Poulding
Postdoctoral Researcher funded by the TOCSYC Project
Software Engineering Research Laboratory
Blekinge Institute of Technology, Sweden

Current Work - General Approach

Metaheuristic search is used to optimise a strategy for generating model instances that test a model transformation. A fitness metric guides the search.



Current Work - Implementation

In our current implementation the objective is a strategy that produces small test sets with high fault-detecting ability. The coverage of the code is used as a fitness metric for this objective.

Strategy Representation

stochastic grammar that emits HUTN

$$
\begin{aligned}
\text{S} &\rightarrow \text{A} \\
\text{A} &\rightarrow \text{`A'}\ \text{`\{'}\ \text{`id:'}\ \text{String}\ \text{`b:'}\ \text{B1..*}\ \text{`c:'}\ \text{C0..*}\ \text{`\}'} \\
\text{B1..*} &\rightarrow \text{B} \mid \text{B}\ \text{`,'}\ \text{B1..*} \\
\text{B} &\rightarrow \text{`\{'}\ \text{`cost:'}\ \text{Cost}\ \text{`\}'} \\
\text{Cost} &\rightarrow \text{`[0,100]'} \\
\cdots &\rightarrow \cdots
\end{aligned}
$$

Strategy Perturbations

changes to the production rule weights

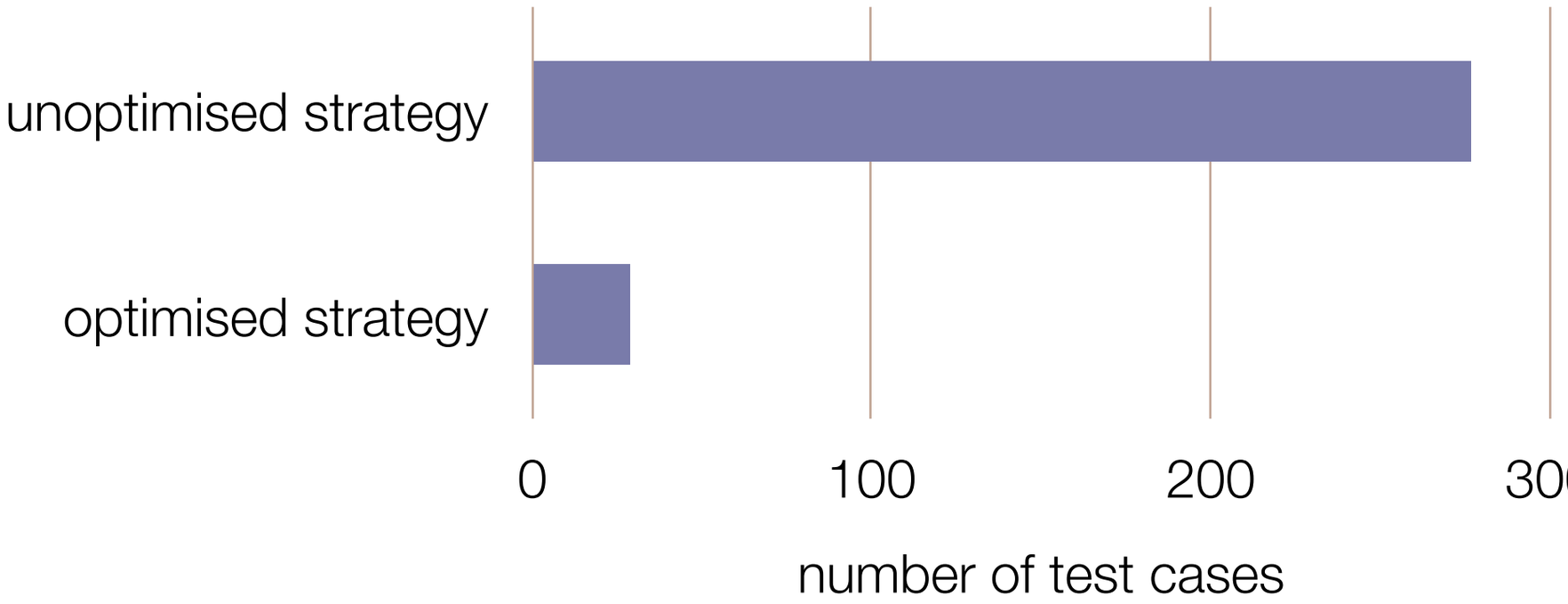
Fitness Metric

based on code coverage

Louis M. Rose and Simon Poulding,
Efficient Probabilistic Testing of Model Transformations using Search,
Proc. 1st International Workshop on Combining Modelling and Search-Based Software Engineering (CMSBSE), 2013

Current Work - Indicative Results

The case study is a transformation used in the process of embodying behaviour in a robot. The fault-detecting ability is compared in terms of the test size required (with high probability) to achieve full coverage.



Louis M. Rose and Simon Poulding,
Efficient Probabilistic Testing of Model Transformations using Search,
Proc. 1st International Workshop on Combining Modelling and Search-Based Software Engineering (CMSBSE), 2013

Workshop Interest

- In our current implementation, code coverage is used to guide the search (and to evaluate the technique).

- Code coverage is obviously not an accurate proxy for the true fault-detecting ability, but can be acceptable metric in some contexts. It's unclear whether this is true for model transformations.

- Would the mutation score be a better fitness metric, in the sense of efficiently guiding the search to testing strategies which detect more faults?

- How should mutants of a transformation be generated for this purpose? What mutation operators are most effective? Are first-order mutants sufficient?

- In general, to what extent does research on mutation testing applied in other domains translate to model transformations?