

Search-Based Software Evolution

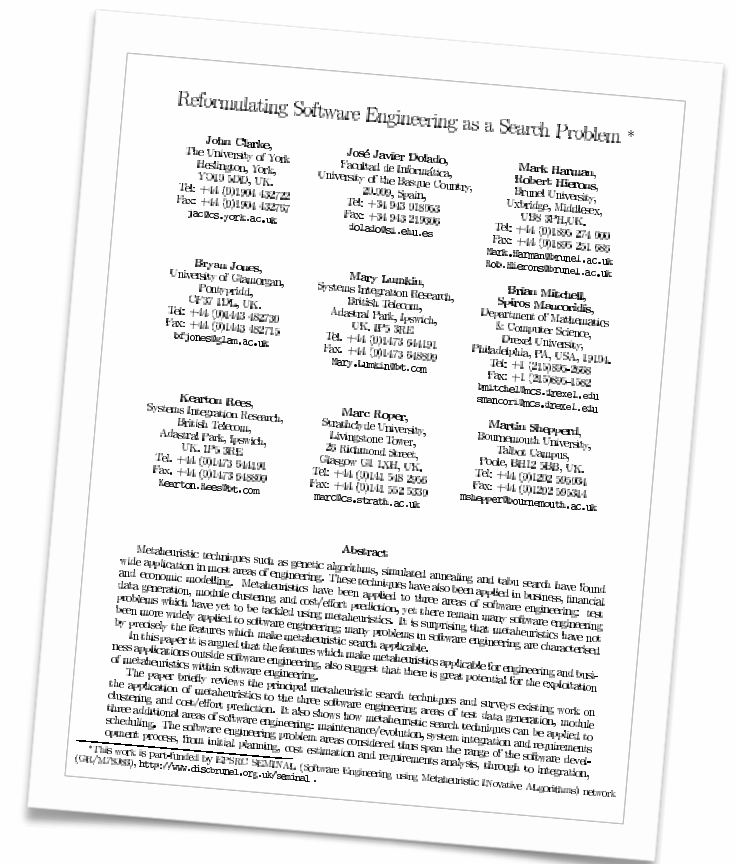
Software Evolution – L7T1

Dr. Vadim Zaytsev aka @grammarware, March 2021



SE is a Search Problem

- Software engineering
 - many constraints
 - interrelated
 - clear $\leftarrow ? \rightarrow$ vague
- Changing **one** aspect has large impact
- How about not seeking *the* solution?
 - seek an *acceptable* solution



Search-Based Software Engineering

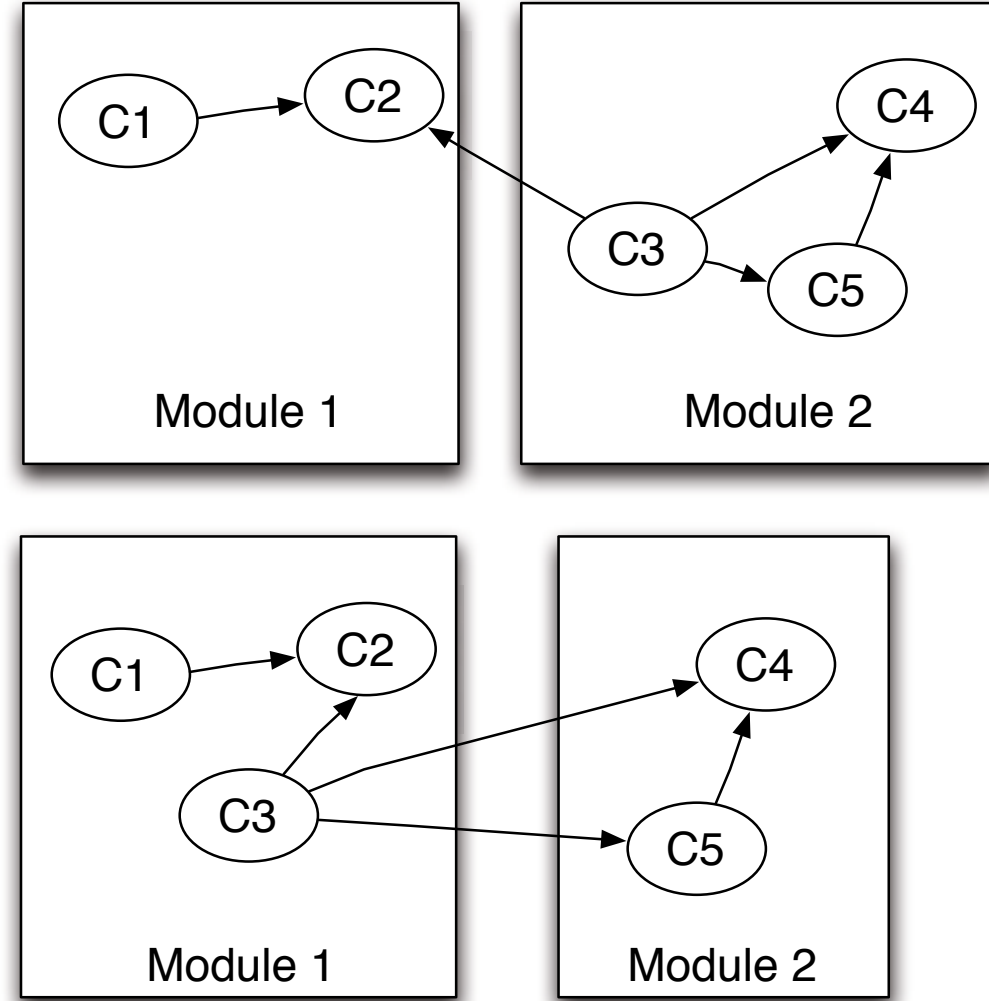
- Represent the problem
 - find an encoding
- Fitness function
 - quantitative evaluation of quality
- Operators
 - to produce new solutions

Genetic Terminology

- Chromosome
 - problem representation
- Genotype
 - decision
- Phenotype
 - objective

Example: Modularisation

- Objective
 - high cohesion / low coupling
- Encoding
 - class diagram? SDG?
- Fitness
 - metrics
- Operators
 - refactorings



Search-Based Algorithms

- Hill Climbing
 - start at a random point, consider neighbours
- Simulated Annealing
 - explores less fit options, cools down with time
- Particle Swarm Optimisation
 - shared-memory solution population
- Genetic Algorithms
 - survival of the fittest

Other SBSE-Solvable Problems

- Complexity reduction
 - graphs + metrics + refactoring
- Automatic program repair
 - patch + test suite + code mutations
- Program optimisation
 - runtime
 - performance/testability/security/...
 - program transformations

Conclusion

- Software evolution can be seen as a search problem
- What's a **chromosome**? **genotype**? **phenotype**?
- Can automate routine/numerous maintenance tasks
 - tackles the quality challenge
- Q&A Sessions @ Canvas
 - ⇒ v.zaytsev@utwente.nl
 - ⇒ <https://discord.gg/n7VQAPNBPD>