Overview of SBSE

CS454, Autumn 2017
Shin Yoo
Search-Based Software Engineering

• Application of all the optimisation techniques we have seen so far, to the various problems in software engineering.

• Not web search engines :( 

• Not code search :( 

Minimise

Cost
Development Time

Maximise

Satisfaction
Fairness

Capture Requirements
Generate Tests
Explore Designs
Maintain/Evolve
Regression Testing
Good Starting Points


Cost Estimation

• Evolve mathematical functions (symbolic regression) that would predict the project development effort based on various input variables.

Project Planning

- Team allocation to project work packages, including the possibility of abandonment (i.e. work no longer needed/practical) and rework (i.e. additional work needed).

Next Release Problem

• Find the ideal set of requirements that balances customer requests, resource constraints, and interdependencies between requirements.


Optimising Source Code

• Random sampling of code transformation to find compiler optimisation


• Automated Parallelisation

Test Data Generation

- Many, many different approaches and ideas; too many to list all:
Regression Testing

• Pareto-efficient Test Suite Minimisation:

• Test Case Prioritisation:

• Multi-objective Prioritisation:

Figure 3: Boxplots of the APFDc metric across all studied subjects. MOEAs and their variants show higher median values and smaller variances.
Maintenance & Reverse Engineering

- Module Clustering: assign modules to clusters based on their relationships


---

**Figure 3. A Module Dependency Graph and its Modularisation using Bunch, taken from [65]**
Deep Parameter Optimisation

• Reveal a property hidden in software as a parameter for tuning.


Monte Carlo Tree Search for Program Synthesis (instead of GP)

Move B

Reward (winning rate) = \frac{2}{3}

Fitness: \frac{2}{3}
Fitness: \frac{1}{3}
Fitness: 0

Reward = \frac{1}{3}
Most of the papers published on SBSE, stored and categorised online:

http://crestweb.cs.ucl.ac.uk/resources/sbse_repository/
Hints for Project Ideas

• Your own experience and/or research

• Reading SBSE papers

• Reading SBSE Challenge Track from SSBSE Conference (see conference proceedings from 2013 and onwards)
Project Pitch (26th October)

• 5 minute sales pitch on what you plan to do.

• **Explicitly** describe the following:

  • Problem

  • How to formulate as search/optimisation

  • How to evaluate, using which data (remember to allow **sufficient** time for data preparation)
Paper Presentations (starting from 31st October)

- Each group will **read two research papers and present one** between 31 October and the end of the semester: 30 minutes for the talk, followed by a complementary short lecture and/or a discussion session.

- There will be a reading list, from which you can choose. Also, after you decide on your project idea, I may be able to help pointing you towards helpful references, which can then be presented.

- At each paper presentation session, two groups participate: one group presents a paper, another group becomes dissenters (i.e. prepares difficult questions). **Dissenters should read the paper too!**
Talk Guideline

• Describe the underlying SE problem clearly

• What is the problem? What are the existing approaches?

• Present the essential SBSE information

• What are the representation and the fitness? What is the choice of algorithms, and why?

• If necessary, read other relevant papers :}
**Dissenters Guideline**

- Don’t ask “I don’t understand this”: finish reading before the class!

- Use “Threats to Validity” section as a starting point, whenever there is one.

- Be suspicious; try to imagine an alternative way.

- Again, if you want to be really successful, you’d read relevant papers - who knows, someone may have already done your job :)
Peer Evaluation

- There will be evaluation forms for everyone in the class during paper presentations - you can note whatever the presenting team has done right/wrong and submit it to me anonymously.