# SEP592
# AI Based Software Engineering, Summer 2020

## Individual Term Project: Software Engineering Applications of Metaheuristic Optimisation

### 1 Aim

To improve *some* aspects of software systems and/or their development life cycle using metaheuristic optimisation techniques that we have looked at during the course. The submission dedaline is **23:59, 21 August 2020**.

### 2 Project Ideas

There are a few ways to find ideas to work on:

- **Consider your own experience as a developer:** if there is some aspect of software development that you think you can improve by automatically searching for a solution, you can deploy algorithms that we have learnt.
- **Replicating and/or Improving Literature:** there are very good surveys [4, 2, 5, 3] and position papers [1] on SBSE, as well as individual research work. You can find a research topic/idea that looks attractive to you, and replicate the work in a small scale. Of course, if you can find a way to improve techniques in the literature, it is even more desirable! There are a few possible places to look:
  - International Symposium on Search Based Software Engineering (SSBSE): `http://dblp.uni-trier.de/db/conf/ssbse/index.html`
  - SBSE Track at ACM Genetic Algorithm and Evolutionary Computation Conference (GECCO): `https://dblp.uni-trier.de/db/conf/gecco/` (Look for Search-based Software Engineering section under each yer's proceedings)
  - International Workshop on Search Based Software Testing (SBST): `https://dblp.org/db/conf/icse/`. URL is for all workshops collocated with International Conference on Software Engineering - look for SBST, which has been running since 2014.

  Many major software engineering conferences have accepted many SBSE papers over the years, so you are also encouraged to read proceedings of International Conference on Software Engieering (ICSE), Foundations of Software Engineering (FSE or ESEC/FSE), Automated Software Engineering (ASE), International Conference on Software Testing, Verification, and Validation (ICST), and International Symposium on Software Testing and Analysis (ISSTA). If you are unsure about the scope of a paper, please consult me.
- **Practical Challenges:** The SSBSE conference, which I mentioned above, has also been running a Challenge Track since 2013. Each year, the organisers choose a few software systems in different languages, and participants are encouraged to do *something* related to those systems using SBSE. Applications that have been submitted so far include: fault prediction, energy consumption improvement, searching for GUI crash events, test case prioritisation, test data generation, deep parameter optimisation, etc.

I understand that the scope is very wide and initially choosing a subject can be a bit confusing. You are welcome to consult any of the T/As and/or me, either by email or face to face. If you want to see me, it is best to make an explicit appointment by email(`shin.yoo@kaist.ac.kr`).

## 3 Presentations

You are required to make two presentations: one at the beginning of the project duration, introducing the group's idea (10 minutes each, 29th July 2020), and another at the end of the project duration, presenting the outcome of the group efforts (exact time to be annouced).

## 4 Points to Cover

Your report should cover the following points:

- Definition of the problem: clearly state what needs to be optimised, and why.
- Methodology: which algorithm/approach you are using to optimise, and why.
- Evaluation: show how much improvement your technique achieves quantitatively.

## 5 Deliverables

Everyone should submit the following deliverables by the submission deadline (**23:59, 21 August 2020**):

- **Implementation**: set up a public GitHup repository for your team, and collaborate using this. The repository should contain sufficient information so that we can execute your project.
- **Report**: include a report that contains the above points. There is no page limit. Include the GitHub repository link.

Submit the report in PDF.

## 6 Guidelines

- Be serious of plagiarism: do not lift either code or text from outside - do your own work.
- Group projects always have the aspect of social dynamics, and this is part of software engineering experience. Actively participate in the project. Be objective, polite, and reasonable about peer assessment.
- You are free to choose any programming language.
- Make sure your submission is self-contained. If it has any external dependency, either include it in the repository or provide a detailed instruction on how to install them. We expect your project to work out of the box with reasonable ease.

## References


[1] Mark Harman. The current state and future of search based software engineering. In *FOSE '07: 2007 Future of Software Engineering*, pages 342–357, Washington, DC, USA, 2007. IEEE Computer Society.

[2] Mark Harman, S. Afshin Mansouri, and Yuanyuan Zhang. Search based software engineering: A comprehensive analysis and review of trends techniques and applications. Technical Report TR-09-03, Department of Computer Science, King's College London, April 2009.

[3] Mark Harman, S. Afshin Mansouri, and Yuanyuan Zhang. Search-based software engineering: Trends, techniques and applications. *ACM Computing Surveys*, 45(1):11:1–11:61, December 2012.

[4] Philip McMinn. Search-based software test data generation: A survey. *Software Testing, Verification and Reliability*, 14(2):105–156, June 2004.

[5] Outi Räihä. A survey on search-based software design. Technical Report D-2009-1, Department of Computer Science, University of Tampere, 2009.