

Some Recent Work on Multi-objective Approaches to Search-Based Software Engineering

Xin Yao

<http://www.cs.bham.ac.uk/~xin>

**CERCIA, School of Computer Science
The University of Birmingham
UK**

**Inspired by David's call for more
online optimisation ...**

... online evolution of my talk ...

The Latest Work on Multi-objective Approaches to Search-Based Software Engineering

Out of 14 Full Papers at SSBSE'13:

Wesley Klewerton Guez Assuncao, Thelma Elita Colanzi, Silvia Regina Vergilio, and Aurora Pozo, “On the Application of the Multi-Evolutionary and Coupling-Based Approach with Different Aspect-Class Integration Testing Strategies”

Lionel Briand, Yvan Labiche, and Kathy Chen, “A Multi-objective Genetic Algorithm to Rank State-Based Test Cases”

Zheng Li, Yi Bian, Ruilian Zhao, and Jun Cheng, “A Fine-Grained Parallel Multi-objective Test Case Prioritization on GPU”

Jeffery Shelburg, Marouane Kessentini, and Daniel R. Tauritz, “Regression Testing for Model Transformations: A Multi-objective Approach”

28.6% of the `SBSE community`. More in the proceedings.

100% of best papers at SSBSE'13!

... continue evolving ...

***Not So Recent* Work on Multi-objective
Approaches to Search-Based
Software Engineering**

Overview

1. Multi-objective **optimisation**

- Why bother, except for another publication?

2. Multi-objective **learning**

- Software Effort Estimation (SEE)

3. Concluding remarks

**Inspired by Barros and Farzat's
challenge talk yesterday (another
award-winning work) ...**

Software Module Clustering: Background

Problem?

Software module clustering is the problem of **automatically organizing software units into modules to improve the program structure.**

Why?

A well-modularized software system is easier and cheaper to develop and maintain.

What is `good`?

A good module structure is regarded as one that has a **high** degree of ***cohesion*** and a **low** degree of ***coupling***.

Software Module Clustering: Previous Solutions

- The module clustering problem is essentially a graph partitioning problem, which is known to be NP-hard. Heuristic algorithms are usually used.
- Previous work on the module clustering problem used **a single-objective formulation** of the problem, combining cohesion and coupling into a single objective, called modularisation quality (MQ). However, **it is not so easy to decide the relative weights that should be applied to each.**

Software Module Clustering As A Multi-objective Problem

Why don't we treat cohesion and coupling as two separate objectives?

If we are to use a multi-objective approach, why limit ourselves to two objectives?

Our Solution

The objectives of **the equal-size cluster (ECA) approach** are defined as follows:

1. the sum of intra-edges of all clusters (maximizing),
2. the sum of inter-edges of all clusters (minimizing),
3. the number of clusters (maximizing),
4. MQ (maximizing),
5. the difference between the maximum and minimum number of modules in a cluster (minimizing).

K. Praditwong, M. Harman and X. Yao, ``**Software Module Clustering as a Multi-Objective Search Problem**,“ *IEEE Transactions on Software Engineering*, 37(2):264-282, March/April 2011.

Which MOEA?

... continue evolving ...

Ancient Work on Multi-objective Approaches to Search-Based Software Engineering

Which Algorithm to Use?

- One of the first algorithms most people would think of is **NSGA-II**.
- However, it **performed poorly** for many objective optimisation problems.
- V. Khare, X. Yao and K. Deb, "**Performance Scaling of Multi-objective Evolutionary Algorithms**," In *Proc. of the 2nd International Conference on Evolutionary Multi-Criterion Optimization (EMO'03)*, Lecture Notes in Computer Science, Vol. 2632, Carlos M. Fonseca, Peter J. Fleming, Eckart Zitzler, Kalyanmoy Deb and Lothar Thiele (Eds.), Springer-Verlag, April 2003, pp.376-390.

Our Solution: Two-Archive Algorithm

The algorithm separates non-dominated solutions into two archives, and is thus called the Two-Archive algorithm.

The two archives focused on convergence and diversity, respectively.

Our experimental results have shown that the Two-Archive algorithm outperforms other MOEAs on **problems with a large number of objectives, i.e., many objective optimisation.**

K. Praditwong and X. Yao, “**A new multi-objective evolutionary optimisation algorithm: the two-archive algorithm,**” *Proc. of the 2006 International Conference on Computational Intelligence and Security (CIS'2006)*, 3-6/11/2006, Guangzhou, China. IEEE Press, Volume 1, pp.286-291.

Multi-objective Approach to Software Module Clustering: Test Problems

The experimental studies the application of the algorithms to 17 different MDGs.

The numbers of modules vary from 20 to over 100.

Future work: to evaluate the algorithm on Apache ant as described by Barros and Farzat yesterday.

What are the research questions we are trying to answer here?

- K. Praditwong, M. Harman and X. Yao, ``**Software Module Clustering as a Multi-Objective Search Problem**,'' *IEEE Transactions on Software Engineering*, 37(2):264-282, March/April 2011.

MQ Value as Assessment Criterion

- *How well does the two-archive multi-objective algorithm perform when compared against the Bunch approach using the MQ value as the assessment criterion?*
- This question compares the two-archive approach to Bunch, using Bunch's own fitness value (MQ).
- It would be expected that Bunch, optimizing for the single objective of MQ, should be able to outperform the two-archive approach, which is optimizing for a balance between this and several other objectives.

Experimental Result (I)

Somewhat surprisingly, **ECA was never outperformed by the hill-climber**, which was the state-of-the-art that specifically designed to optimise MQ.

In particular, **ECA outperformed the hill-climber for all weighted MDGs significantly.**

Why Bother (I)

- **Introducing additional objectives does not necessary mean that the quality on the original objective will get worse.**
- **In fact, it is well known in the evolutionary computation community, both theoretically and experimentally, that one could solve single objective problems by adding helper objectives.**
- **Even at SSBSE'13, we had excellent presentation:**
 - ***Arina Buzdalova, Maxim Buzdalov, and Vladimir Parfenov, “Generation of Tests for Programming Challenge Tasks Using Helper-Objectives”***

Cohesion and Coupling

How well do the two archive algorithm and the Bunch perform at optimizing cohesion and coupling separately?

MQ was defined in order to optimise cohesion and coupling. Why not look at cohesion and coupling directly?

Experimental Result (II)

For both cohesion and coupling, **in all but one of the problems studied, the ECA approach outperforms the hill-climbing approach with statistical significance.** The remaining case has no statistically significant difference between ECA and Bunch.

Why Bother (II)

- **It does what it is supposed to do.**
- **MO approaches address better the challenge mentioned by several speakers here that an optimisation algorithm cares only about the fitness function, which might have unintended consequences.**
- **MO approaches help to make “syntactical” optimisation more “meaningful” by including more “truthful” objectives.**

Inspired Again by David's Tutorial ...

where he pointed out an SBSE challenge of protocol tuning

Protocol Tuning

Motivation

Protocol tuning can yield significant gains in energy efficiency and resource requirements, **i.e., non-functional properties**, which is of particular importance for sensornet systems in which resource availability is severely restricted.

Problem formulation

Five objectives, covering performance (2), reliability (1), and efficiency (2)

Experimental Study

First, we applied factorial design and statistical model fitting methods to reject insignificant factors and locate regions of the problem space containing near-optimal solutions by principled search.

Second, we applied the **Strength Pareto Evolutionary Algorithm 2** and **Two-Archive Evolutionary Algorithm** to explore the problem space.

J. Tate, B. Woolford-Lim, I. Bate and X. Yao, ``**Evolutionary and Principled Search Strategies For Sensornet Protocol Optimisation,**'' *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 42(1):163-180, February 2012.

Obtained Results

Multi-objective evolutionary algorithms (MOEAs) can significantly outperform a simple factorial design experimental approach when tuning sensornet protocols against multiple objectives, producing higher quality solutions with lower experimental overhead.

The two-archive algorithm outperformed the SPEA2 algorithm, at each generation and in the final evolved solution, for each protocol considered in this paper.

Why Bother (III)

- It is **not always true** that evolutionary approaches are slower than other competing approaches and multi-objective approaches are slower than single objective approaches although this is not usually the reason for using multi-objective approaches.

Reliability is important not just for protocol design, it is important for all software systems.

Optimal Testing Resource Allocation in Modular Software Systems

A software system is typically comprised of a number of modules. Each module needs to be assigned appropriate testing resources before the testing phase. Hence, a natural question is **how to allocate the testing resources to the modules so that the reliability of the entire software system is maximized.**

Z. Wang, K. Tang and X. Yao, ``**Multi-objective Approaches to Optimal Testing Resource Allocation in Modular Software Systems,**'' *IEEE Transactions on Reliability*, 59(3):563-575, September 2010.

Experimental Study

We formulated OTRAPs as **two types of multi-objective problems**.

1. Consider the reliability of the system and the testing cost as two objectives.
2. The total testing resource consumed was also taken into account as the third objective.

NSGA-II was used to solve the **two-objective** problem.

A Harmonic Distance Based Multi-Objective Evolutionary Algorithm (HaD-MOEA) was proposed and applied to the **three-objective** problem.

Obtained Results

MOEAs not only managed to achieve *almost the same solution* as that which can be attained by single-objective approaches, but also found *simultaneously a set of alternative solutions*.

These solutions showed different trade-offs between the reliability of a system and the testing cost, and hence can facilitate informed planning of a testing phase.

Z. Wang, K. Tang and X. Yao, ``Multi-objective Approaches to Optimal Testing Resource Allocation in Modular Software Systems," *IEEE Transactions on Reliability*, 59(3):563-575, September 2010.

Beyond Optimisation

- **SBSE has exclusively focused on optimisation, and rightly so.**
- **However, many SE optimisation problems are inherently linked to learning and modelling, especially predictive modelling.**
- **Example?**
 - **Keeping my tendency of looking backwards**

Software Effort Estimation (SEE)

Problem:

- **Estimation of the effort required to develop a software project (e.g., in person-hours).**
- **Based on features such as functional size (numerical), required reliability (ordinal), programming language (categorical), development type (categorical), team expertise (ordinal), etc.**

Importance:

- **Main factor influencing project cost.**
- **Overestimation vs. underestimation.**

Machine Learning in SEE

- **Uses completed projects as training examples for creating SEE models.**
- **Can be used as decision support tools.**
- **Examples:**
 - **Multilayer Perceptrons (MLPs).**
 - **Radial Basis Function networks (RBFs).**
 - **Regression Trees (RTs).**

Different Performance Measures in SEE

Several different performance measures are used in SEE.

Mean Magnitude of the Relative Error:

$$MMRE = \frac{1}{T} \sum_{i=1}^T MRE_i,$$

where $MRE_i = |\hat{y}_i - y_i|/y_i$; \hat{y}_i is the predicted effort; and y_i is the actual effort.

Percentage of estimations within 25% of the actual values:

$$PRED(25) = \frac{1}{T} \sum_{i=1}^T \begin{cases} 1, & \text{if } MRE_i \leq \frac{25}{100} \\ 0, & \text{otherwise} \end{cases}.$$

Logarithmic Standard Deviation:

$$LSD = \sqrt{\frac{\sum_{i=1}^T \left(e_i + \frac{s^2}{2} \right)^2}{T - 1}},$$

where s^2 is an estimator of the variance of the residual e_i and $e_i = \ln y_i - \ln \hat{y}_i$.

Different Performance Measures in SEE (State-of-the-art)

- **There is no agreed single universal performance measure.**
- **The relationship among different measures in SEE is not well understood.**
- **Existing SEE approaches use at most one measure during the learning procedure. It is unclear whether a model/learner trained using one measure would still perform well under a different measure.**
- **Many paper did not even report the measure they used in training!**

SEE by Multi-objective Learning

We propose to view SEE as a multi-objective learning problem:

- Each performance measure is considered explicitly as an objective to be optimised.
- A multi-objective algorithm can be used to create SEE models that are generally good in terms of all objective measures (e.g., non-dominated models), but that present different trade-offs among these measures.
- These different trade-offs can help us to understand to what extent different measures behave differently and what the relationship among these measures is.
- They help to enhance the robustness of the models.

Multi-objective Ensemble Learning

- An **ensemble** consists of a group of learning machines, i.e., base (individual) learners.
- **Multi-objective learning is ideal for ensembles.**
 - H. Chen and X. Yao, ``**Multiobjective Neural Network Ensembles based on Regularized Negative Correlation Learning**,'' *IEEE Transactions on Knowledge and Data Engineering*, 22(12):1738-1751, December 2010.
 - A Chandra and X. Yao, ``**Ensemble learning using multi-objective evolutionary algorithms**,'' *Journal of Mathematical Modelling and Algorithms*, 5(4):417-445, December 2006.
- Measures that behave very differently from each other can be used as a source of diversity to create SEE ensembles (***Pareto ensemble***).

Yet More Multi-objective Optimisation?

- **Not really. Learning is different from optimisation because the goal of learning is to improve generalisation, e.g., to maximize the predictive ability of learned SEE models.**
- **Yet, generalisation cannot be quantified accurately and precisely in practice. It can only be estimated.**
- **Minimising an error function is not exactly the same as maximising the generalisation.**

Some Research Questions

1. What is the relationship among different performance measures for SEE?
2. Can we use different performance measures as a source of diversity to create SEE ensembles? In particular, can that improve on the performance measures used as objectives with respect to a standard learning algorithm for the same type of base model?
3. Is it possible to outperform the state-of-the-art?

L. Minku and X. Yao. **Software Effort Estimation as a Multi-objective Learning Problem.** ACM Transactions on Software Engineering, 2012 (accepted).
<http://www.cs.bham.ac.uk/~minkull/publications/MinkuYaoTOSEM12.pdf>

Relationship among Different Performance Measures (RQ1)

Interesting insight into the learning problem:

- **MMRE, PRED(25) and LSD behave even more differently than one might have initially thought.**
- **MMRE and LSD can present even opposite behaviour.**

Choosing a Non-dominated Model

- A software manager could pick a desired trade-off from the non-dominated front.
- Picking a trade-off may still not be easy, as previously mentioned. However, it is possible to use the Pareto ensemble to provide a good trade-off automatically.
- This is rather different from optimisation where a solution has to be selected from the non-dominated solutions.
- **MO learning is more natural than MO optimisation.**

Ensemble vs. Individual Learners (RQ2)

- **Better/similar performance for all data sets.**
- **MO learning does not necessarily compromise individual performance.**

Results for large (> 60) data sets:

Data Set	Pareto Ensemble		
	LSD	MMRE	PRED(25)
Wins	6/8	5/8	7/8
P-value	0.0000	0.0012	0.0003

***P-value of Wilcoxon test over all runs and all large data sets.**

Results for small (< 35) data sets:

Data Set	Pareto Ensemble		
	LSD	MMRE	PRED(25)
Wins	3/5	2/5	3/5
P-value	0.1170	0.7166	0.0004

***P-value of Wilcoxon test over all runs and all small data sets.**

Robustness

The ensembles did well on those performance measures that were **not** used in multi-objective learning, which provides an additional evidence towards our robustness claim.

Comparative Study

Nine approaches considered:

- MLP, RBF, RT, log+EBA, Bagging+MLP, Bagging+RT, Bagging+RBF, Rand+MLP, NCL+MLP.

Additional performance measures for evaluation:

- MdMRE, MAE, MdAE.

- L. Minku and X. Yao. **Software Effort Estimation as a Multi-objective Learning Problem**. ACM Transactions on Software Engineering, 2012 (accepted).
<http://www.cs.bham.ac.uk/~minkull/publications/MinkuYaoTOSEM12.pdf>

Pareto Ensemble vs. Other Approaches

Best ranked approach for each data set:

Approach	LSD	MMRE	PRED(25)	MdMRE	MAE	MdAE
Cocomo81	RT	Bag+MLP	Bag+MLP	Bag+MLP	Bag+MLP	Bag+MLP
Sdr	RT	RT	Bag+RT	RT	RT	RBF
Nasa	Bag+RT	RT	Bag+MLP	Bag+MLP	Bag+RT	Bag+RT
Desharnais	Bag+RT	Bag+MLP	Pareto Ens	Pareto Ens	Pareto Ens	Pareto Ens
Nasa93	RT	RT	RT	RT	RT	RT
Org1	Bag+RBF	Pareto Ens				
Org2	Bag+RT	Pareto Ens				
Org3	Pareto Ens	Pareto Ens	Log + EBA	Log + EBA	Log + EBA	Log + EBA
Org4	Bag+RBF	Pareto Ens	RT	RT	Pareto Ens	Pareto Ens
Org5	Bag+RT	Log + EBA	Bag+RBF	Rand + MLP	Bag + RT	RT
Org6	Bag+RBF	Pareto Ens	Pareto Ens	Pareto Ens	Bag + RBF	Pareto Ens
Org7	Bag+RT	Log + EBA	Log + EBA	Log + EBA	Bag + RBF	Pareto Ens
OrgAll	RT	Pareto Ens				

- Pareto ensemble was **ranked first most often** for the ISBSG data sets (more heterogeneous). Friedman tests were used.
- They are **more robust**, even according to measures that were not used in multi-objective learning.

Pareto Ensemble vs. Other Approaches

Number of times ranked best:

Approach	LSD	MMRE	PRED(25)	MdMRE	MAE	MdAE
Pareto Ens	1	6	5	5	5	7
RT	4	3	2	3	2	2
Bag+RT	5	0	1	0	2	1
Bag+MLP	0	2	2	2	1	1
Log + EBA	0	2	2	2	1	1
Bag+RBF	3	0	1	0	2	0
Rand+MLP	0	0	0	1	0	0
RBF	0	0	0	0	0	1
Total	13	13	13	13	13	13

- Pareto ensemble was **most often ranked first**, except for LSD.

But How Much Better?

Statistically significant difference might not be a large-enough difference to be of interest in practice.

Effect size has often been used to indicate the extent of the difference.

Choosing between Pareto ensemble and other approaches results in many **medium or large effect sizes, representing a considerable practical impact.**

Recap

RQ1: What is the relationship among different performance measures for SEE?

- **Different measures behave very differently from each other, sometimes presenting even opposite behaviours.**

RQ2: Can we use different performance measures as a source of diversity to create SEE ensembles? In particular, can that improve on the performance measures used as objectives with respect to a standard learning algorithm for the same type of base model?

- **Yes, MMRE, PRED(25) and LSD behave very diversily, being appropriate for creating ensembles. The Pareto ensemble of MLPs obtained statistically better/similar results to single MLPs.**

RQ3: Is it possible to outperform the state-of-the-art?

- **Yes.**

Concluding Remarks (I)

- **Multi-objective optimisation can be regarded as an approach to add more “meanings” to a syntactical optimisation process.**
- **We need optimisers that understand what we really want, not just interpret our numerical fitness functions literally.**
- **Finding a better optimiser that gives you a smaller or larger near optima may not be an interesting future work by itself. Solving the original software engineering problem is.**

Concluding Remarks (II)

Learning is at least as important as optimisation for SBSE.

- **Some software engineering problems are learning problems, e.g., SEE and software defect prediction.**
- **Expensive fitness evaluation is a real challenge for some (many?) SBSE problems. Why not using surrogate models? Learning is essential in building surrogate models.**
- **For optimisation (or online *) in a dynamic environment, predictive modelling is also essential.**

Ensemble learning can be made easier through multi-objective algorithms.

Questions?