

How Multi-Objective Genetic Programming Is Effective for Software Development Effort Estimation?

Filomena Ferrucci, Carmine Gravino, and Federica Sarro

Via Ponte Don Melillo, 84084 Fisciano (SA), Italy
{fferrucci, gravino, fsarro}@unisa.it

The idea of exploiting search-based methods to estimate development effort is based on the observation that the effort estimation problem can be formulated as an optimization problem. As a matter of fact, among possible estimation models, we have to identify the best one, i.e., the one providing the most accurate estimates. Nevertheless, in the context of effort estimation there does not exist a unique measure that allows us to compare different models and consistently derives the best one [1]. Rather, several evaluation criteria (e.g., MMRE and Pred(25)) covering different aspects of model performances (e.g., underestimating or overestimating) are used to assess and compare estimation models [1]. Thus, considering the effort estimation problem as an optimization problem we should search for the model that optimizes several measures. From this point of view, the effort estimation problem is inherently multi-objective. Nevertheless, all the studies that have been carried out so far on the use of search-based techniques for effort estimation exploited single objectives (e.g., [2][3]). Those studies have also revealed that the use of some measures as fitness function (e.g., MMRE) decreased a lot the accuracy in terms of other summary measures [2]. A first attempt to take into account different evaluation criteria has been reported by Ferrucci *et al.* [3], where Genetic Programming (GP) exploited a combination of two measures (e.g., Pred(25) and MMRE) in a single fitness function, providing encouraging results. Obviously, an approach based on combination of measures is the simplest way to deal with the multi-objective problem but this can determine biases since there is no defined way of aggregating different objectives. Thus, we investigated the use of a Multi-Objective Genetic Programming (MOGP) approach with the aim to verify its effectiveness. To the best of our knowledge this is the first attempt to apply a MOGP approach in this field. In particular, we designed and experimented an adaptation to GP of the Non dominated Sort Genetic Algorithm-II (NSGA-II) exploiting multi-objective functions based on a different number and type of evaluation criteria. Moreover, we compared the performance of MOGP with GP using different fitness functions. The preliminary empirical analysis, carried out with some publicly available datasets included in the PROMISE repository [4], revealed that the choice of the evaluation criteria employed in the definition of the fitness function affects the overall accuracy of both MOPG and GP. Nevertheless, no significant statistical difference has been found between the best results achieved with MOGP and GP. Thus, the use of a more sophisticated technique, such as MOGP, seems to be not cost/effective in this context. However, this is a preliminary analysis

that needs to be deepened also using more data. Moreover, the use of other multi-objective optimization approaches could be exploited to investigate whether there are improvements in the accuracy of the obtained estimation models.

References

- [1] Kitchenham, B., Pickard, L., MacDonell, S., Shepperd, M.: What accuracy statistics really measure. *IEEE Procs. Software* 148(3), 81–85 (2001)
- [2] Burgess, C., Lefley, M.: Can Genetic Programming Improve Software Effort Estimation: A Comparative Evaluation. *Inform. and Softw. Technology* 43(14), 863–873 (2001)
- [3] Ferrucci, F., Gravino, C., Oliveto, R., Sarro, F.: Genetic Programming for Effort Estimation: An Analysis of the Impact of Different Fitness Functions. In: *Procs. of SSBSE 2010*, pp. 89–98 (2010)
- [4] PROMISE Repository of Emp. Softw. Engine. data, <http://promisedata.org/repository>