Search-based Testing using State-based Fitness

Raluca Lefticaru, Florentin Ipate
Department of Computer Science and Mathematics
University of Pitesti, Romania
Email: raluca.lefticaru@gmail.com, florentin.ipate@ifsoft.ro

Abstract

This paper investigates the adequacy of a general form of fitness function, based on a state-based specification, by characterizing the associated search problem and the dynamics of the applied metaheuristic. The measurement approach of the fitness landscape is experimented on various examples and suggests some tuning choices to be made for the metaheuristic considered.

Introduction. A popular objective function for structural search-based software testing, namely the *approach level + normalized branch level* ($al + nbl$) form [2] was tailored to a state machine [1], in order to find input parameters for a given path in a state machine which satisfy some given constraints for each transition. The *approach level* was adapted to correspond to all followed transitions from the machine path (instead of only critical branches like in structural testing [2]); the *normalized branch level* was derived from the predicate of the first unsatisfied transition constraint, using the same transformations as in structural search-based testing [2]. Landscape analysis of fitness functions is an important issue and this paper makes an attempt to characterize the search problems associated with the fitness functions used in search-based testing, employing the following measures: *autocorrelation* and *fitness distance correlation* (FDC) [3].

Empirical Evaluation. Waeselynck et al. [3] investigated a measurement approach on various landscapes (using the diameter, the autocorrelation and the generation rate of better solution) to find adequate setting of simulated annealing parameters, applied to test generation. For our evaluation, we first used the Calendar problem (Cal 1) inspired from [3], for which the authors considered 24 landscapes, corresponding to the combinations of 4 fitness functions and 6 neighborhood operators. However, several fitness functions can be built for Cal1 problem and we considered an artificial state machine and 6 associated general fitness functions, of the form $al + nbl$. We retained for comparison two fitness functions from [3] (an exact distance to the optimum and an approximate distance to the optimum, which are good metrics for the problem) and we compared them to the $al + nbl$ functions, which have a general form and do not incorporate any additional information which could give a better guidance. The special designed fitness functions had better autocorrelation and FDC coefficients, in comparison to the state-based fitness functions. This difference was significant when generating test data with simulated annealing (the special designed fitness functions clearly outperformed the state-based ones in most cases). However, the difference in success rates obtained in generating test data with genetic algorithms and particle swarm optimization was smaller, and some success rates obtained by the $al + nbl$ functions were comparable to the ones obtained by especially designed fitness functions.

Conclusions. The measurements performed and the success rates obtained by the metaheuristics employed showed that the general state-based fitness functions $al + nbl$ may produce results comparable to those produced by fitness functions designed especially for a particular situation. Further experiments performed suggest some tuning choices for genetic algorithms with state-based fitness functions having non-linear constrained transitions for integer parameters. Future work concerns analyzing other variants of fitness function for state-based testing, performing experiments on a larger benchmark of real world objects and extending the approach to the case in which the method parameters have more complex types.

References

[1] R. Lefticaru and F. Ipate. Automatic state-based test generation using genetic algorithms. In *SYNASC 2007*, pages 188–195, 2007.

[2] P. McMinn. Search-based software test data generation: a survey. *Softw. Test., Verif. Reliab.*, 14(2):105–156, 2004.

[3] H. Waeselynck, P. Thévenod-Fosse, and O. Abdellatif-Kaddour. Simulated annealing applied to test generation: landscape characterization and stopping criteria. *Empirical Software Engineering*, 12(1):35–63, 2007.