Landscape State Machines: Tools for Evolutionary Algorithm Performance Analyses and Landscape/Algorithm Mapping

David Corne[1], Martin Oates[1,2], and Douglas Kell[3]

[1] School of Systems Engineering, University of Reading,
Whiteknights, Reading RG6 6AY, United Kingdom
d.w.corne@reading.ac.uk

[2] Evosolve Ltd, Stowmarket, Suffolk, United Kingdom,
moates@btinternet.com

[3] Department of Chemistry, UMIST, Manchester, United Kingdom
dbk@umist.ac.uk

Abstract. Many evolutionary algorithm applications involve either fitness functions with high time complexity or large dimensionality (hence very many fitness evaluations will typically be needed) or both. In such circumstances, there is a dire need to tune various features of the algorithm well so that performance and time savings are optimized. However, these are precisely the circumstances in which prior tuning is very costly in time and resources. There is hence a need for methods which enable fast prior tuning in such cases. We describe a candidate technique for this purpose, in which we model a landscape as a finite state machine, inferred from preliminary sampling runs. In prior algorithm-tuning trials, we can replace the 'real' landscape with the model, enabling extremely fast tuning, saving far more time than was required to infer the model. Preliminary results indicate much promise, though much work needs to be done to establish various aspects of the conditions under which it can be most beneficially used. A main limitation of the method as described here is a restriction to mutation-only algorithms, but there are various ways to address this and other limitations.

1 Introduction

The study of fitness landscapes [14] in the context of evolutionary search strives to understand what properties of fitness landscapes seem correlated with the success of specific evolutionary algorithms (EAs). Much progress has been made, with a number of landscape metrics under investigation as well as sophisticated statistical techniques to estimate landscape properties [1,5,7,9,11,13]. Meanwhile, much effort has also gone into constructing landscapes with well understood properties in attempt to yield hypotheses and guidelines which may apply to 'real' landscapes [4,6,8,11,12]. For the most part, a striking aspect of such investigations has been the ability of EAs consistently to undermine predictions [1,6,10]. Correlation between proposed metrics or landscape features and evolutionary algorithm difficulty tends to be weak, or bothered by the presence of convincing counterexamples. Here we present an alternative ap-

S. Cagnoni et al. (Eds.): EvoWorkshops 2003, LNCS 2611, pp. 187–198, 2003.

proach to exploring landscapes and algorithm performance on them. We model a landscape as a finite state machine, whose states represent fitness levels and state transition probabilities characterize mutation. Such a model can be approximately inferred from sampling during an EA (or other algorithm) run, and then used for test-driving a range of algorithms under consideration to apply to the 'real' landscape.

This promises to be valuable in cases where good results are at a premium, but fitness evaluations on the 'real' landscape are prohibitively expensive, precluding intensive *a priori* algorithm comparison and/or parameter tuning; in contrast, a fitness evaluation on a landscape state machine (LSM) is computationally trivial. The success of this technique for algorithm comparison depends on the degree to which the LSM approximation captures those features of the real landscape which are salient in terms of algorithm comparison. Viability also depends on whether sufficiently useful LSMs can be inferred without incurring undue cost in prior search of the real landscape. We start to investigate these questions, and conclude that the technique is promising.

2 Landscape State Machines

What we call a landscape state machine (LSM) is simply a finite state machine (FSM) which models certain aspects of a search landscape. To be specific, given a search space E, an operator M (which takes a point $s \in E$, and returns another point from E) and an associated transition matrix T (such that t_{ij} gives the probability of yielding point $j \in E$ after applying M to $i \in E$). An LSM model of this landscape is a set of states and arcs (S, A), such that S corresponds to a partition of the set E, and A corresponds to an abstraction of T. In the extreme, the LSM can model a landscape precisely, with precisely one state for each point in E, and A corresponding precisely to T.

In the general case, one state in the LSM will map onto many points in E. We will normally expect the mapping between S and E to be such that each state s corresponds to a set of points in E with equivalent fitness. More generally, we may define an equivalence relation R, which partitions E into c of equivalence classes $E_1,...,E_c$. The states in the LSM can then correspond precisely to the equivalence classes. In the case that the equivalence relation R forces equality in both fitness and genotype, the LSM model becomes the exact model described above. More generally, and as we later do in section 4, we might define R in such a way that states may be associated with a partition of the fitnesses into bands (e.g. all points with fitness between 0.2 and 0.3 might define an equivalence class).

2.1 Examples and Motivation

Before we discuss the uses of this simple LSM concept, an example will serve to clarify and motivate the issues involved. Consider the simple MAX-ONES problem, in which a candidate solution is a binary string of length L, and the fitness of a candidate (which is to be maximised) is the number of 1s it contains. Further, imagine we are interested in addressing this problem with an EA, and will employ the single-gene bit-