Generating Easy and Hard Problems using the Proximate Optimality Principle

John A.W. McCall
IDEAS Research Institute
Robert Gordon University
Aberdeen, UK
j.mccall@rgu.ac.uk

Lee A. Christie
IDEAS Research Institute
Robert Gordon University
Aberdeen, UK
l.a.christie4@rgu.ac.uk

Alexander E.I. Brownlee
Division of Computing Science
and Mathematics
University of Stirling, UK
sbr@cs.stir.ac.uk

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search

Keywords

Problem Generation, Proximate Optimality, Estimation of Distribution Algorithms, Landscapes

1. INTRODUCTION

An important central question in metaheuristics is "Given an algorithm, for which problem classes is it well suited?" This is a current topic of interest in the research community (e.g. [3]). In this extended abstract we explore the Proximate Optimality Principle (POP) and use it to define an approach to generation of easy and hard problems for a given algorithm. Section 2 shows how POP leads naturally to a concept of coherence between metrics in representation and objective spaces. Section 3 describes a method for generating coherent and "anti-coherent" problems on bit string representations. Section 4 describes experiments on these problems using a steepest ascent hill climber. We analyse our results, showing the effect of coherence on algorithm performance. Section 5 contains our conclusions and suggestions for further research.

2. PROXIMATE OPTIMALITY PRINCIPLE

The Proximate Optimality Principle (POP) is often cited as a property relating the representation of a problem with the fitness assigned to each solution in the representation. Examples include [1, 4, 5, 7–11]. POP is often stated in terms like "good solutions possess some similar structure" [5]. Metaheuristics make moves within the solution representation during search, seeking to find better solutions by learning which moves will increase fitness. When a problem has POP, it is believed, the relationship between fitness and solution configuration is sufficiently easy to learn. Here we consider a stronger concept of *coherence*, where neighbouring solutions in representation space have similar fitness. Coherence will imply POP, though POP does not necessarily imply coherence. For the rest of this abstract, we will consider a local hillclimber acting on bitstrings, though the approach is applicable to more general algorithms and representations. Similarity of solution representation and similarity of fitness gives rise to two metrics on bitstrings. For bitstrings s and t, we define the *representation metric* $D_r(s,t)$ to be the usual Hamming metric, which defines neighbourhoods for a local hillclimber. Similar solutions to s will exist in a ball in representation space of small radius $B_\delta(s) = \{t : D_r(s,t) < \delta\}$. Solutions with similar fitness to s will belong to the pre-image under f of a ball in objective space of small radius $B_\epsilon(f(s)) = \{r \in R : |f(s) - r| < \epsilon\}$. Given a fitness function, f, on bitstrings, we define the *fitness metric*, $D_f(s,t)$ to be $|f(s) - f(t)|$. It is the metric induced by f on the representation space. Coherence can therefore be characterised by the extent to which the solutions in the pre-image of small real intervals cluster in the representation space inside local neighbourhoods of solutions with fitnesses in those intervals. In other words there is a large intersection between $f^{-1}(B_\epsilon(f(s)))$ and the union of all $B_\delta(t)$ where $f(t) = f(s)$.

3. PROBLEM GENERATION

We now describe a constructive approach for generating coherent bitstring problems for local hillclimbers. We first define fitness for a set S of seed solutions. These are chosen at random and we construct the complete graph G_S on these where the edge between solutions s and t is given weight $D_r(s,t)$. We now construct a minimum spanning tree for T_S for G_S, using Prim's algorithm [6]. This sorts the seed solutions in such a way as to minimise distance, or maximise similarity, summed over the tree. We now choose a root for T_S and assign it fitness $|S|$. Fitness for other solutions in S is assigned by decrementing S by tree depth. Thus for the solutions in S, D_r and D_f are coherent. We extend coherence across the space by interpolating fitness for any solution s not in S as follows. Let p and q be the two solutions in S nearest to s with respect to D_r. We define $f(s)$ to be:

$$f(s) = ((1-t)f(p) + tf(q)) \tag{1}$$
$$\text{where: } t = D_r(s,p)/(D_r(s,p) + D_r(s,q))$$

That is, the fitness of s is weighted to lie between that of p and q depending on the similarity to each. The resultant function is strongly coherent by construction. Moreover,we can amend the procedure to construct "anti-coherent" functions by selecting T_S to be a maximum spanning tree of G_S,

GECCO '15 July 11-15, 2015, Madrid, Spain

ACM ISBN 978-1-4503-3488-4/15/07.
DOI: http://dx.doi.org/10.1145/2739482.2764890

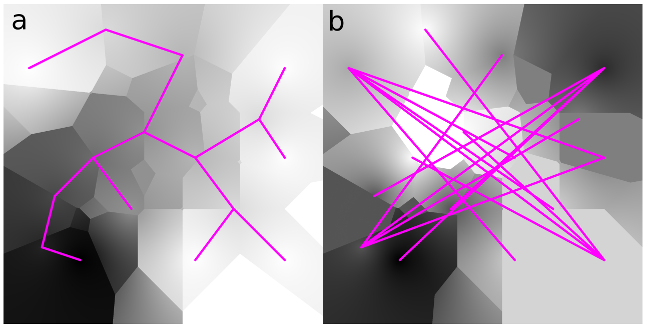

Figure 1: Minimum and maximum spanning trees over fitness levels, rendered in 2D space

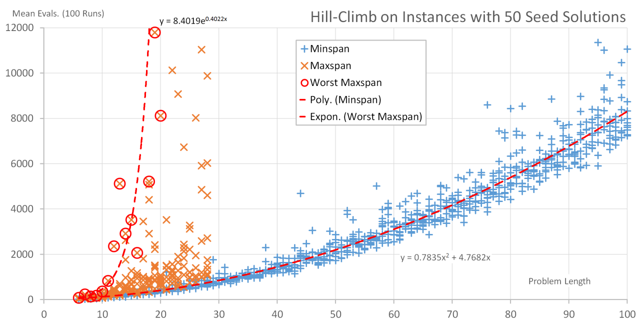

Figure 2: Number of evaluations required by the hillclimber vs bitstring length

again using Prim's algorithm but with negative costs. Here the fitness gradient on T_S runs between solutions that are maximally dissimilar but for other solutions it is the opposite meaning that fitness and representation are highly incoherent. We call functions generated in this way "anti-coherent".

Figures 1 a and b illustrate this in a simple 2 dimensional space. Each figure uses the same seed set S and tray level indicates level of fitness (white is low). In Figure 1a, T_S is a minimum spanning tree and a clear gradient (with a few small plateaux) can be seen running from low to high along the edges of the tree. In Figure 1b, T_S is a maximum spanning tree and, as a result, f is a much more complex function with many local optima.

4. EXPERIMENTAL RESULTS

We hypothesise that, for a hillclimber, coherent functions will be easy and anti-coherent functions will be hard.

We generated 10 coherent functions for each length on bitstrings of length 6-100 and the same number of anti-coherent functions using the same seed sets. Seed sets were generated by uniformly at random sampling 50 distinct points from the search space. For each function we ran a multi-restart steepest ascent hillclimber 100 times and recorded the time taken to solve the problem as a function of problem size.

Figure 2 plots, for each function, the average number of evaluations required by the hillclimber to solve it against bitstring length. Coherent functions are plotted as '+' (blue) and anti-coherent functions as '×' (orange). Curves fitted to the data show that coherent functions are solved in polynomial time whereas anti-coherent functions require exponential time to solve. This confirms our hypothesis.

5. CONCLUSIONS

In this short paper we have defined a concept of coherence based on POP. Coherence relates similarity in representation space with similar fitness in objective space using neighbourhoods defined by metrics. Our experiments show that coherent functions are easy and "anti-coherent" functions are hard for a steepest ascent hillclimber. This is consistent with the widely-held that problems that satisfy POP are amenable to solution. Our approach has focussed solely on a hillclimber and works precisely because the representation space metric coincides with local neighbourhoods on which the hillclimber operates. More complex algorithms will operate on the representation in more complex ways and the concept of coherence will need to be generalised. One current line of active research [1, 2] is to use probabilistic graphical models on representation space to structure fitness in ways coherent with the action of algorithm operators.

6. REFERENCES

[1] A. E. I. Brownlee, J. A. W. McCall, and L. A. Christie. Structural coherence of problem and algorithm: An analysis for EDAs on all 2-bit and 3-bit problems. In *Proc. IEEE CEC*, 2015. To appear.

[2] L. A. Christie, J. A. McCall, and D. P. Lonie. Minimal walsh structure and ordinal linkage of monotonicity-invariant function classes on bit strings. *Proc. GECCO*, pages 333–340, 2014.

[3] J. He, T. Chen, and X. Yao. On the easiest and hardest fitness functions. *Evolutionary Computation, IEEE Transactions on*, 19(2):295–305, April 2015.

[4] H. Li and D. Landa-Silva. An elitist grasp metaheuristic for the multi-objective quadratic assignment problem. *Proc. EMO (LNCS 5467)*, pages 481–494, 2009.

[5] D. Niizuma, K. Yasuda, and A. Ishigame. Multipoint tabu search based on proximate optimality principle – application of parts concept. *IEEJ Transactions on Electr Electr Eng*, 2(6):635–642, 2007.

[6] R. C. Prim. Shortest connection networks and some generalizations. *Bell system technical journal*, 36(6):1389–1401, 1957.

[7] A. Salhi, J. A. V. Rodríguez, and Q. Zhang. An estimation of distribution algorithm with guided mutation for a complex flow shop scheduling problem. In *Proc. GECCO*, pages 570–576. ACM, 2007.

[8] J. Sun, Q. Zhang, and X. Yao. Meta-heuristic combining prior online and offline information for the quadratic assignment problem. *IEEE Transactions on Cybernetics*, 44(3):429–444, Mar 2014.

[9] D. Tuzun and L. I. Burke. A two-phase tabu search approach to the location routing problem. *European Journal of Operational Research*, 116(1):87 – 99, 1999.

[10] K. Yaguchi, K. Tamura, K. Yasuda, and A. Ishigame. Basic study of proximate optimality principle based combinatorial optimization method. *2011 IEEE Int. Conf. on Systems, Man, and Cybernetics*, Oct 2011.

[11] Q. Zhang, J. Sun, and E. Tsang. An evolutionary algorithm with guided mutation for the maximum clique problem. *IEEE T Ev. C*, 9(2):192–200, 2005.