Information Sciences 298 (2015) 468–490

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Structural bias in population-based algorithms

Anna V. Kononova [a,*], David W. Corne [a], Philippe De Wilde [b], Vsevolod Shneer [c], Fabio Caraffini [d,e]

[a] Department of Computer Science, Heriot-Watt University, Edinburgh EH14 4AS, UK
[b] School of Computing, University of Kent, Canterbury CT2 7NZ, UK
[c] School of Mathematical and Computer Sciences, Heriot-Watt University, Edinburgh EH14 4AS, UK
[d] Centre for Computational Intelligence, School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, UK
[e] University of Jyväskylä, Department of Mathematical Information Technology, P.O. Box 35 (Agora), 40014 Jyväskylä, Finland

ARTICLE INFO

Article history:
Received 8 August 2014
Received in revised form 5 November 2014
Accepted 22 November 2014
Available online 4 December 2014

Keywords:
Structural bias
Algorithmic design
Evolutionary computation
Population-based algorithm
Optimisation

ABSTRACT

Challenging optimisation problems are abundant in all areas of science and industry. Since the 1950s, scientists have responded to this by developing ever-diversifying families of ‘black box’ optimisation algorithms. The latter are designed to be able to address any optimisation problem, requiring only that the quality of any candidate solution can be calculated via a ‘fitness function’ specific to the problem. For such algorithms to be successful, at least three properties are required: (i) an effective informed sampling strategy, that guides the generation of new candidates on the basis of the fitnesses and locations of previously visited candidates; (ii) mechanisms to ensure efficiency, so that (for example) the same candidates are not repeatedly visited; and (iii) the absence of structural bias, which, if present, would predispose the algorithm towards limiting its search to specific regions of the solution space. The first two of these properties have been extensively investigated, however the third is little understood and rarely explored. In this article we provide theoretical and empirical analyses that contribute to the understanding of structural bias. In particular, we state and prove a theorem concerning the dynamics of population variance in the case of real-valued search spaces and a ‘flat’ fitness landscape. This reveals how structural bias can arise and manifest as non-uniform clustering of the population over time. Critically, theory predicts that structural bias is exacerbated with (independently) increasing population size, and increasing problem difficulty. These predictions, supported by our empirical analyses, reveal two previously unrecognised aspects of structural bias that would seem vital for algorithm designers and practitioners. Respectively, (i) increasing the population size, though ostensibly promoting diversity, will magnify any inherent structural bias, and (ii) the effects of structural bias are more apparent when faced with (many classes of) ‘difficult’ problems. Our theoretical result also contributes to the ‘exploitation/exploration’ conundrum in optimisation algorithm design, by suggesting that two commonly used approaches to enhancing exploration – increasing the population size, and increasing the disruptiveness of search operators – have quite distinct implications in terms of structural bias.

* Corresponding author.
E-mail addresses: anna.kononova@gmail.com (A.V. Kononova), d.w.corne@hw.ac.uk (D.W. Corne), p.dewilde@kent.ac.uk (P. De Wilde), v.shneer@hw.ac.uk (V. Shneer), fabio.caraffini@gmail.com (F. Caraffini).

http://dx.doi.org/10.1016/j.ins.2014.11.035
0020-0255/

1. Introduction

Successful implementation of any randomised population-based optimisation algorithm depends on the efficiency of both its sampling component and exploitation of previously sampled information. Among other fields, Evolutionary Computation (EC) [13] provides various examples of randomised population-based search strategies. Greatly simplified, any evolutionary computation algorithm is a guided re-sampling strategy where movement of points is directed by its operators assisted by selection criteria based on currently attained values of the objective function. A vast body of research in the field of Evolutionary Computation deals with efficient exploitation of information already contained within the population [49] while little attention has been paid to investigation of whether or not a specific combination of algorithmic operators is actually capable of reaching all parts of the search space efficiently. This paper attempts to draw attention to this issue and starts to investigate the latter question.

Inspection of recent literature [40,9,38,55] confirms the presence of a tendency to (over-) complicate both the design of individual algorithmic operators and the logic of their assembly, counter to the rationale of the well-known Occam's razor,[1] sometimes to such a degree that the end result turns out to be intractable. Researchers seem regularly to be swayed by an attraction towards 'multiplying entities beyond necessity'. We suggest that a materially greater contribution to the understanding of population-based algorithms and their design can be obtained via 'going back to basics'. More specifically, the great majority of optimisation algorithms fall within a class of generate-and-test methods, iteratively alternating between these two components until a termination criterion is met. Ideally, the generating/sampling component of such methods should have the following characteristics [51]:

1. future samples should be biassed by information obtained from previously visited points, i.e., the algorithm should be *informed*,
2. future samples should be previously unvisited samples i.e., the algorithm should be *non-redundant*,
3. every solution in the search space should be equally accessible i.e., the algorithm should be *complete*.

It is worth noting our use of the phrase *equally accessible* within the 'completeness' characteristic. Often, for example, algorithm designers may be subconsciously swayed by the fact that the randomness of the initialisation process means that every part of the search space is *reachable*, and hence feel no further need to consider this characteristic. Reachability and completeness (the way we define it here) are however very different. For example, if a stochastic hill-climbing algorithm includes a uniform random initialisation in $\mathbb{R}^n$, then all points are reachable, however if the perturbation operator is designed to add only integer-valued vectors, then there are extreme variations in the accessibilities of different points in the space.

Clearly, evolutionary computation methods build richly upon their 'generate-and-test' backbone architecture. However the above guidelines remain valid, and, in practice, they translate well into rules for algorithm design. The first two properties – informedness and non-redundancy in the sampling process – have been extensively researched, each from a variety of viewpoints. To some extent, however, contributions related to these two properties have appeared in diverse and unconnected literature, using varying terminology, and there remains a need to creatively assimilate their findings.

For example, with sufficient imagination one can see that the *informedness* property is closely linked to the concepts of exploration, exploitation, and their balance, which is considered to be primary in the behaviour of evolutionary algorithms (EAs), as examples of stochastic "generate-and-test" methods [11]. Exploration and exploitation are fundamental for evolutionary optimisation [13] but surprisingly, several decades after the first examples of EAs have been proposed, they still lacked even a proper definition. Over the following years, a lot of research has been carried out in this direction - the latest survey of results can be found in [49]. The current consensus definitions consider *exploration* as the process of visiting entirely new regions of a search space whilst *exploitation* as the process of visiting those regions of a search space within the neighbourhood of previously visited points [49].

The second characteristic, *non-redundancy*, has been investigated under the guise of 'non-revisiting' algorithms. Inspired by ideas from Tabu search [15,16], basic evolutionary algorithms have been extended to ensure the non-revisiting property [52–54]. Another direction of research into the non-redundancy property is the study of diversity management in evolving populations. Diversity in populations can refer to differences in solutions in either the values of coordinates ('genotypic' diversity) or the objective function ('phenotypic' diversity). To date, no single measure exists which can suitably characterise diversity in the face of all kinds of problems and search logics [37]. The situation is further complicated by the fact that a diverse population offers benefits at some stages of evolutionary process (helps avoid premature convergence to local optima) and creates obstacles in others (impedes exploitation) [7]. The most popular diversity-preserving mechanisms include [18] niching, crowding, restricted mating, sharing, multiploidy, elitism, injection, alternative replacement strategies [32] and fitness uniform selection [20].

Much promising research is also carried out that tries to explore the connections between *informedness* and *non-redundancy*, stemming from the fact that exploration of the search space is only possible if populations are diverse enough [49]. However, different amounts of exploration and exploitation are needed for different optimisation problems. Currently there

[1] Originally attributed to William of Occam, reformulated by Betrand Russell as "Whenever possible, substitute constructions out of known entities for inferences to unknown entities". [43]