Information Sciences 188 (2012) 17–43

Contents lists available at SciVerse ScienceDirect

Information Sciences

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

Ockham's Razor in memetic computing: Three stage optimal memetic exploration

Giovanni Iacca [a], Ferrante Neri [a,*], Ernesto Mininno [a], Yew-Soon Ong [b,1], Meng-Hiot Lim [c]

[a] Department of Mathematical Information Technology, P.O. Box 35 (Agora), University of Jyväskylä, 40014 Jyväskylä, Finland
[b] School of Computer Engineering, Nanyang Technological University Block N4, 2b-39, Nanyang Avenue, Singapore 639798, Singapore
[c] School of Electrical and Electronic Engineering, Nanyang Technological University Singapore, S1-B1a-05, Nanyang Avenue, Singapore 639798, Singapore

ARTICLE INFO

Article history:
Received 2 April 2011
Received in revised form 31 October 2011
Accepted 11 November 2011
Available online 19 November 2011

Keywords:
Memetic computing
Evolutionary algorithms
Memetic algorithms
Computational intelligence optimization

ABSTRACT

Memetic computing is a subject in computer science which considers complex structures as the combination of simple agents, memes, whose evolutionary interactions lead to intelligent structures capable of problem-solving. This paper focuses on memetic computing optimization algorithms and proposes a counter-tendency approach for algorithmic design. Research in the field tends to go in the direction of improving existing algorithms by combining different methods or through the formulation of more complicated structures. Contrary to this trend, we instead focus on simplicity, proposing a structurally simple algorithm with emphasis on processing only one solution at a time. The proposed algorithm, namely three stage optimal memetic exploration, is composed of three memes; the first stochastic and with a long search radius, the second stochastic and with a moderate search radius and the third deterministic and with a short search radius. The bottom-up combination of the three operators by means of a natural trial and error logic, generates a robust and efficient optimizer, capable of competing with modern complex and computationally expensive algorithms. This is suggestive of the fact that complexity in algorithmic structures can be unnecessary, if not detrimental, and that simple bottom-up approaches are likely to be competitive is here invoked as an extension to memetic computing basing on the philosophical concept of Ockham's Razor. An extensive experimental setup on various test problems and one digital signal processing application is presented. Numerical results show that the proposed approach, despite its simplicity and low computational cost displays a very good performance on several problems, and is competitive with sophisticated algorithms representing the-state-of-the-art in computational intelligence optimization.

1. Introduction

Emerging technologies in computer science and engineering, as well as the demands of the market and the society, often impose the solution, in the every day life, of complex optimization problems. The complexity of today's problems is due to various reasons such as high non-linearities, high multi-modality, large scale, noisy fitness landscape, computationally expensive fitness functions, real-time demands, and limited hardware available (e.g. when the computational device is portable and cheap). In these cases, the use of exact methods is unsuitable because, in general, there is not sufficient prior

* Corresponding author. Tel.: +358 14 260 1211; fax: +358 14 260 1021.
E-mail addresses: giovanni.iacca@jyu.fi (G. Iacca), ferrante.neri@jyu.fi (F. Neri), ernesto.mininno@jyu.fi (E. Mininno), asysong@ntu.edu.sg (Y.-S. Ong), emhlim@ntu.edu.sg (M.-H. Lim).
[1] Tel.: +65 6790 6448; fax: +65 6792 6559.

0020-0255/$ - see front matter
doi:10.1016/j.ins.2011.11.025

knowledge (hypotheses) on the optimization problem; thus, computational intelligence approaches become not only advisable but often the only alternative to face the optimization.

Scientific research in computational intelligence optimization can be classified into two general categories.

- In the first case, by following the No Free Lunch Theorem (NFLT) [71], the application problem becomes the starting point for the algorithmic design, i.e. after an analysis of the problem, an algorithm containing components to address the specific features of the problem is implemented. Amongst domain specific algorithms, in [18] an ad hoc Differential Evolution (DE) is implemented for solving the multisensor fusion problem; in [55] DE based hybrid algorithm is designed to address an aerodynamic design problem; in [8], an optimization approach is given with reference to the study of a material structure; in [3,41] a computational intelligence approach is designed for a control engineering problem while in [44,43] a medical application for Human Immunodeficiency Virus (HIV) is addressed; in [63] a DE based hybrid algorithm is implemented to design a digital filter for paper production industry.
- In the second case, computer scientists attempt to perform the algorithmic development with the aim of designing a robust algorithm, i.e. an algorithm capable to display a respectable performance on a diverse set of test problems. Usually, the newly designed algorithms are tested on a set of test problems, see [58]. Some examples of articles containing this kind of approach are [69,68,33,2,7,54].

Regardless of the aim of the designer, usually the algorithmic design does not result into a fully novel computational paradigm. On the contrary, computer scientists, on the basis of the results previously attained in literature perform an unexplored algorithmic coordination in order to detect the lowest possible value of the objective functions. In our view, the most typical approaches which describe the "mental process" of the computer scientists, when a novel algorithmic design is performed, can be subdivided into the following three categories.

1. Starting from an existing optimization algorithm, its structure is "perturbed" by slightly modifying the structure and adding on extra components. Obviously, this approach attempts to obtain a certain performance improvement in correspondence to the proposed modifications. A successful example of this research approaches is given in [2] where a controlled randomization on DE control parameters appear to offer a promising alternative to the standard DE framework, see also [42]. Other examples are given in [7,26] where the variation operator combining the solutions of a population is modified in the context of DE and Particle Swarm Optimization (PSO), respectively. Other examples of PSO based algorithms obtained by modifying the original paradigm are shown in [72,59].
2. Starting from a set of algorithms, they are combined in a hybrid fashion with the trust that their combination and coordination leads to a flexible structure displaying a better performance than the various algorithms considered separately. Two examples of recently proposed algorithms which are basically the combination, by means of activation probabilities, of various meta-heuristics are given in [66,51]. A very similar concept is contained in the idea of ensemble, see [28,29], where multiple strategy concur by means of a self-adaptive/randomized mechanism to the optimization of the same fitness function. Another similar concept is given in [54] where multiple search strategies, a complex randomized self-adaptation, and a learning mechanism are framed within a DE structure. In [45] a combination of multiple algorithms is performed by assigning a certain success probability to each of them to detect the global optimum. In [1,46] multiple algorithmic components are coordinated by means of the structural mapping of the population. In [35] a coordination scheme which promotes a sequence of local search activations is proposed. In [67] a heuristic technique assists PSO in selecting the desired solutions while solving multi-objective optimization problems. In [56], a Cellular Automata scheme is integrated in the standard PSO velocity update rule in order to modify the particle trajectories and avoid them being trapped in local optima. Another good example of this algorithmic philosophy is the Frankenstein's PSO, see [36], which combines several successful variants of PSO in order to make an ultimate PSO version.
3. Starting from some knowledge of the problem features, the problem-knowledge is integrated within an algorithmic structure. These algorithms usually make use of a theoretical background in order to enhance the performance of a metaheuristic framework. A typical case of this approach is in [12,11] where, on a solid theoretical basis, the search directions (by means of the distribution of solution) progressively adapt to the shape of the landscape. This mechanism allows the algorithm to be rotational invariant and thus keen to handle the non-separability of the functions. By following a similar logic, two rotational invariant versions of DE are introduced in [52]. In the context of PSO, a theoretical approach justifying the employment of inter-particle communication is presented in [10].

According to the modern definition given in [49], these three categories fall within the umbrella name of Memetic Computing (MC). More specifically, MC is defined as "a paradigm that uses the notion of meme(s) as units of information encoded in computational representations for the purpose of problem-solving", where meme is an abstract concept which can be for example a strategy, an operator, or a search algorithm. In other words, a MC is strictly related to the concept of modularity and a MC structure can be seen as a collection of interactive modules whose interaction, in an evolutionary sense, leads to the generation of the solution of the problem. In this sense MC is a much broader concept with respect to a Memetic Algorithm (MA), which according to the definition in [13] is an optimization algorithm composed of an evolutionary framework and a list of local search algorithms activated within the generation cycle, of the external framework (see also [38,37]). In this paper, we will refer to the unifying concept of MC and will consider each algorithm, in the light of the definition in [49], as a