Expert Systems with Applications 41 (2014) 877–885

Contents lists available at ScienceDirect

Expert Systems with Applications

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

An adaptive single-point algorithm for global numerical optimization

Francisco Viveros-Jiménez [a,*], José A. León-Borges [b], Nareli Cruz-Cortés [a]

[a] Centro de Investigación en Computación, Instituto Politécnico Nacional, Av. Juan de Dios Bátiz s/n, Col. Nueva Industrial Vallejo, C.P 07738 México D.F., Mexico
[b] Universidad Politécnica de Quintana Roo, Av. Tulum, Manzana 1, Lote 40, Planta Alta, SM 2, 77500 Cancún, Quintana Roo, Mexico

ARTICLE INFO

Keywords:
Unconstrained problems
Numerical optimization
Hill-climbing
Adaptive behavior

ABSTRACT

This paper describes a novel algorithm for numerical optimization, called Simple Adaptive Climbing (**SAC**). SAC is a simple efficient single-point approach that does not require a careful fine-tunning of its two parameters. SAC algorithm shares many similarities with local optimization heuristics, such as random walk, gradient descent, and hill-climbing. SAC has a restarting mechanism, and a powerful adaptive mutation process that resembles the one used in Differential Evolution. The algorithms SAC is capable of performing global unconstrained optimization efficiently in high dimensional test functions. This paper shows results on 15 well-known unconstrained problems. Test results confirm that SAC is competitive against state-of-the-art approaches such as micro-Particle Swarm Optimization, CMA-ES or Simple Adaptive Differential Evolution.

1. Introduction

In the most general case, global numerical optimization is the task of finding the point (x^*) with the smallest (minimization case) or bigger (maximization case) function value ($f(x^*)$). There exists special cases where the search space is highly constrained, it means that the solution found by the algorithm must be optimal, and it must satisfy the defined problem's constrains too (Cruz-Cortés, Trejo-Pérez, & Coello Coello, 2005; Coello Coello & Cruz-Cortés, 2002, 2004). Other kind of numerical optimization problems are the called *Multi-Objective Optimization* with more than one objective functions to be optimized at the same time (Coello Coello & Cruz-Cortés, 2005). In this work we are interested on designing an algorithm for non-linear unconstrained single-objective optimization problems.

The Evolutionary Algorithms have been widely utilized to find optimal solutions to non-linear optimization problems. The Evolutionary Algorithms are population-based, e.g. they handle a set of possible solutions, further they are probabilistic methods. On the other hand, single-point optimizers such as hill-climbing, and random walks handle one point at a time. Random as well as deterministic versions of the hill-climbing algorithms can be found in the specialized literature.

Most of the recent efficient optimizers for solving unconstrained optimization, such as restart Covariance Matrix Adaptation Evolution Strategy (**CMA-ES**) (Auger & Hansen, 2005), can be considered complex approaches because they use Hessian and covariance matrix. Those approaches are very effective and greatly overcome more of the simple heuristic approaches, as shown in events such as Congress on Evolutionary Computation (Hansen, 2006) and the Black-Box Optimization Benchmarking workshop (Hansen, Auger, Finck, & Ros, 2010). However, we consider that a simpler approach capable of giving quality results is sufficient for most of the times. This paper presents a novel approach based on the idea of having one single point but with a powerful and adaptive mutation strategy: Simple Adaptive Climbing (**SAC**). It is a single-point approach similar to hill-climbing or random walk algorithms. Hill-climbing like algorithms are known for having difficulties solving global optimization problems with multiple local optimum values (as explained further in Section 2). Our algorithm can solve these problems due to the following features:

- It can move to all the possible search directions.
- Its search radius is adaptively adjusted, i.e. it is adjusted depending on the current situation.
- It has a restarting mechanism that allows the algorithm to resume its process when getting stuck in a solution.

We tested our approach on 15 well-known unconstrained problems. Test results show that the algorithm works as well as more complex state-of-the-art approaches, such as micro-Particle Swarm Optimization, CMA-ES, or Simple Adaptive Differential Evolution.

The contents of this paper are organized as follows: First, we give a brief introduction on hill-climbing algorithm in Section 2. Section 3 contains the SAC description. Section 4 contains the experimental design. Section 5 presents a comparison of SAC against four state-of-the-art approaches: Elitist Evolution, micro-Particle Swarm

* Corresponding author.
E-mail addresses: pacovj@hotmail.com (F. Viveros-Jiménez), jleon@upqroo.edu.mx (J.A. León-Borges), nareli@cic.ipn.mx (N. Cruz-Cortés).

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.08.018

Optimization, Simple Adaptive Differential Evolution, and Restart CMA-ES. Finally, Section 6 concludes this paper.

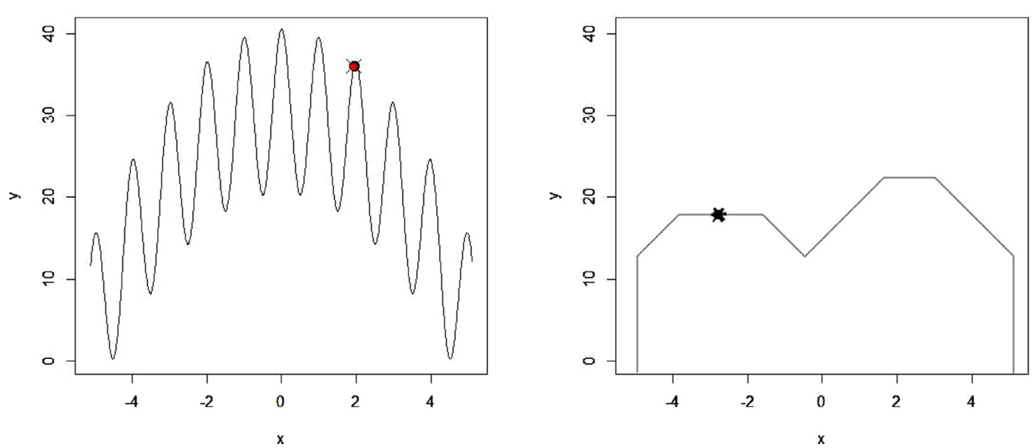
2. Brief review of hill-climbing like algorithms

Let us imagine that you are a mountain climber trying to reach the peak having a real thick fog. Imagine that you have forgotten some important stuff like a compass and the map, but at least you have a lot of food, the perfect climbing suit and equipment, and a machine that tells your current altitude. How will you find the mountain peak? Hill-climbing like algorithms are heuristics for getting you to the mountain peak, that is, keep going to the highest point surrounding you.

Hill-climbing algorithms are single point optimizers with adjustable search radius. Fig. 1 depicts a generalization of hill-climbing algorithms. As we can observe in Fig. 1, these algorithms

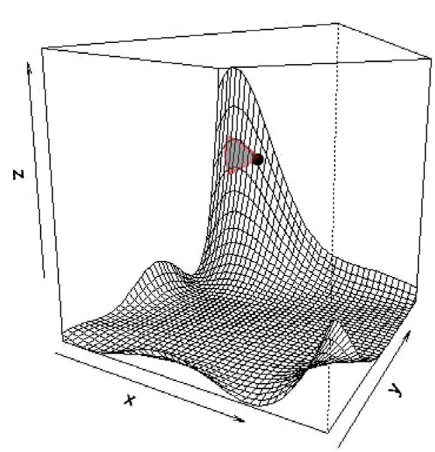
```
Data: parameters for configuring the mutation operator behavior
Result: Xbest (best solution found)
Set X as a random initial solution;
Initialize the search radius vector (steps) using parameters;
while terminationCriterion() do
    Set Xnew = mutate(Xbest, steps, parameters);
    if f(Xnew) < f(Xbest) then
        Set steps = adjustStepswhenSuccess(steps, parameters);
        Set Xbest = Xnew;
    else
        Set steps = adjustStepswhenFailure(steps, parameters);
```

Fig. 1. Generic algorithm for hill climbers (minimization case).



(a) Foothill problem: the searching process is stuck in a local optimum

(b) Plateau problem: the searching process is stuck in a flat surface



(c) Ridge problem: the searching area (dark gray) does not allow improving.

Fig. 2. Most common problems found in hill-climbing algorithms (maximization case).