Computers & Operations Research 73 (2016) 119–131

Contents lists available at ScienceDirect

Computers & Operations Research

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

Splitting for optimization

Qibin Duan *, Dirk P. Kroese

School of Mathematics and Physics, The University of Queensland, Brisbane 4072, Australia

ARTICLE INFO

Article history:
Received 30 August 2015
Received in revised form
4 April 2016
Accepted 11 April 2016
Available online 13 April 2016

Keywords:
Evolutionary computation
Splitting method
Continuous optimization
Artificial bee colony
Differential evolution

ABSTRACT

The splitting method is a well-known method for rare-event simulation, where sample paths of a Markov process are split into multiple copies during the simulation, so as to make the occurrence of a rare event more frequent. Motivated by the splitting algorithm we introduce a novel global optimization method for continuous optimization that is both very fast and accurate. Numerical experiments demonstrate that the new splitting-based method outperforms known methods such as the differential evolution and artificial bee colony algorithms for many bench mark cases.

1. Introduction

Randomized algorithms have shown to be of significant benefit for solving complicated optimization problems. In particular, such methods are of great use in finding (near) optimal solutions to highly multi-modal functions, "black-box" problems where gradients are difficult to obtain, and problems with complicated constraints. Since 1960s many well-known random algorithms for optimization have been proposed. Many of these algorithms can be viewed as population Monte Carlo algorithms, where a sample (population) of individuals is modified randomly over time in order to produce a high-performing sample according to some chosen objective. Often such algorithms are nature-inspired. Examples include evolution strategy (ES) [1], evolutionary programming (EP) [2], genetic algorithms (GA)) [3] and, more recently, the cross-entropy (CE) method [4], differential evolution (DE) [5], particle swarm optimization (PSO) [6], ant colony optimization(ACO) [7], fast EP (FEP)[8], artificial bee colony (ABC) [9] and many other inventive methods based on the principle of exploration and exploitation.

The splitting method is a well-known method for rare-event simulation, where sample paths of a Markov process are split into multiple copies during the simulation, so as to make the occurrence of a rare event more frequent. The purpose of this paper is to introduce the "splitting" idea to the optimization toolbox for continuous optimization, and to show that the approach, when reduced to its core elements, can outperform other well-known methods in terms of accuracy and speed.

To motivate the splitting technique, we draw on various ideas from rare-event simulation. It has been realized for some time that the problem of minimizing a complicated continuous or discrete function $S(\mathbf{x})$, $\mathbf{x} \in \mathcal{X}$ is closely related to the efficient estimation of rare-event probabilities of the form $\mathbb{P}(S(\mathbf{X}) \le \gamma)$, where $\mathbf{X}$ is a random element of $\mathcal{X}$, distributed according to a given probability density function (pdf), e.g., the uniform pdf on $\mathcal{X}$. The latter requires efficient sampling from the level set $\{\mathbf{x} \in \mathcal{X}: S(\mathbf{x}) \le \gamma\}$. By gradually decreasing γ the level set becomes smaller and smaller until it only contains elements that lie close to the minimizer of S. For γ close to the minimum, the event $\{S(\mathbf{X}) \le \gamma\}$ will be very rare. Hence, it is useful to apply rare event simulation techniques to optimization problems. This is, for example, the premise of the cross-entropy (CE) method, which aims to find a sequence of pdfs $f_1, f_2, f_3, \ldots$ that converges to the pdf that concentrates all its mass in the set of points where S is minimal. In the CE method the densities $f_1, f_2, \ldots$ are parameterized by a fixed-dimensional parameter vector, which is updated at each iteration using the cross-entropy (or Kullback–Leibler) distance. If instead a non-parametric approach is taken, the densities can be represented by a collection of particles, whose distribution is updated at each iteration. This is where the splitting method enters the scene.

The splitting method was first put forward by [10] for time-dependent Markovian models and later generalized in [11] to both static (that is, not involving time) and non-Markovian models. The latter modification is called *Generalized Splitting* (GS), which will be the focus of our discussion below.

The purpose of GS method is to estimate the rare-event probability $\mathbb{P}(S(\mathbf{X}) \le \gamma)$ for some (small) γ, where $\mathbf{X}$ has a specified

* Corresponding author.
E-mail addresses: q.duan@uq.edu.au (Q. Duan), kroese@maths.uq.edu.au (D.P. Kroese).

http://dx.doi.org/10.1016/j.cor.2016.04.015
0305-0548/

nominal distribution. This is done by first defining a sequence of levels $\{\gamma_t\}$ decreasing to γ and then constructing a sequential sampling scheme that samples from the conditional distribution of $\mathbf{X}$ given $\{S(\mathbf{X}) \le \gamma_t\}$. Note that if γ is equal to the minimum of S, then sampling $\mathbf{X}$ conditional on $\{S(\mathbf{X}) \le \gamma\}$ is equivalent to sampling from the minimizer of S. However, the problem is that in general the minimum value is not known, and hence the intermediate values $\{\gamma_t\}$ have to be determined adaptively. This is the motivation for the ADAptive Multilevel splitting algorithm (ADAM) in [12,13]. The ADAM algorithm has be applied to mostly combinatorial optimization problems. For continuous optimization, where the nominal distribution is taken to be uniform, the ADAM algorithm is generally more difficult to apply, as sampling $\mathbf{X}$ conditional on $\{S(\mathbf{X}) \le \gamma_t\}$ may be too time-consuming or complicated.

In this paper we propose to replace the complicated sampling step in the ADAM algorithm with a simpler one, while retaining the other features. Instead of sampling (at stage t) from the uniform distribution on the difficult "level set" $\{\mathbf{x}: S(\mathbf{x}) \le \gamma_t\}$, our sampling scheme involves sampling from a collection of multivariate normal distributions, using a Gibbs sampler. The mean vector and covariance matrix of the normal distributions are determined by the current population of individuals. This simplification greatly increases the applicability of the ADAM method, making it competitive for continuous optimization. We compare the method with the best performing algorithms in this area and demonstrate that it can outperform them for a suite of established test functions.

The rest of the paper is organized as follows. In Section 2, we review the mathematical framework of the GS and ADAM algorithms, and put forward the new splitting idea for continuous optimization. For easy comparison we summarize two well-performing algorithms, DE and ABC, in Section 3. In Section 4, we employ a popular suite of test functions to evaluate the performance of the proposed optimization technique. We describe the precise settings of the numerical experiments and show the comparison between DE, ABC, and the new splitting algorithm for continuous optimization (SCO). Finally, in Section 5, we further discuss the results of the numerical experiments, and compare the proposed algorithm with other algorithms via existing comparative studies.

2. Mathematical framework and algorithms

2.1. Mathematical framework

Let $S(\mathbf{x})$ be a continuous function on $\mathbb{R}^n$. We wish to find the minimum $\gamma^* = \min_{\mathbf{x}} S(\mathbf{x})$ and the global minimizer $\mathbf{x}^* = \operatorname{argmin}_{\mathbf{x}} S(\mathbf{x})$, assuming for simplicity that there is only one minimizer. Let f be some "nominal" pdf f, e.g., the uniform pdf on some bounded subset of $\mathbb{R}^n$. Suppose first that γ^* is known. To find the corresponding $\mathbf{x}^*$ we could sample a random vector $\mathbf{X}$ conditional on the rare event $\{S(\mathbf{X}) \le \gamma^*\}$, which basically means sampling from the argmin set $\{\mathbf{x}^*\}$. This can be done using the GS method by sampling iteratively from intermediate (increasingly rare) events $\{S(\mathbf{X}) \le \gamma_t\}$, for levels $\infty = \gamma_0 \ge \gamma_1 \ge \cdots \ge \gamma_{T-1} \ge \gamma_T = \gamma^*$. Define the *level set* of S corresponding to level γ_t to be the set $\{\mathbf{x}: S(\mathbf{x}) \le \gamma_t\}$. We call it the γ_t-*level set* for short. Let f_t be the conditional pdf of $\mathbf{X} \sim f$ given $\{S(\mathbf{X}) \le \gamma_t\}$; that is, $f_t(\mathbf{x})$ is proportional to $f(\mathbf{x}) I_{\{S(\mathbf{x}) \le \gamma_t\}}$. In particular, we are interested in sampling from $f^*(\mathbf{x}) = f_T(\mathbf{x}) \propto f(\mathbf{x}) I_{\{S(\mathbf{x}) \le \gamma^*\}}$. The GS method works as follows.

Given the sequence of intermediate levels γ_t, $t = 0, \ldots, T$, and an initial sample (population) $\mathcal{X}_0$ from $f_0 = f$, execute the following two phases at each iteration t, from $t=0$ to $t = T - 1$:

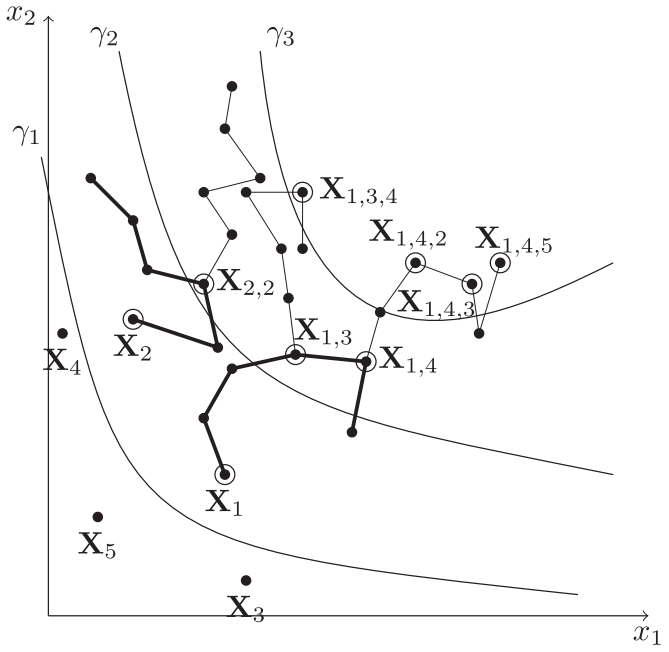

Fig. 1. Illustration of the GS algorithm in 2-D space.

(a) Let $\mathcal{E}_{t+1} = \{\mathbf{x} \in \mathcal{X}_t: S(\mathbf{x}) \le \gamma_{t+1}\}$, which is referred as the *elite set* of $\mathcal{X}_t$. Its size is denoted by N_{t+1}. Note that the elite elements are distributed according f_{t+1}.

(b) Split the elite population in $\mathcal{E}_{t+1}$ to create the next population $\mathcal{X}_{t+1}$, distributed according to f_{t+1}. Increase t by one and go to Step (a).

The splitting step (b) can be implemented in many different ways; for example, in [12] it is done by running a Markov chain on the γ_{t+1}-level set starting from each point in the elite set $\mathcal{E}_{t+1}$ and storing each state in $\mathcal{X}_{t+1}$. The only requirement is that the Markov chain has stationary pdf f_{t+1}.

Fig. 1 illustrates how the splitting is performed on a typical problem in 2-D space. Here, there are three levels, γ_t, $t = 1, 2, 3$, and the initial sample set is $\mathcal{X}_0 = \{\mathbf{X}_1, \ldots, \mathbf{X}_5\}$, where $\mathbf{X}_1, \ldots, \mathbf{X}_5 \overset{\text{iid}}{\sim} f_0$. Since two of the five initial points, namely $\mathbf{X}_1$ and $\mathbf{X}_2$, are such that $S(\mathbf{X}_1)$ and $S(\mathbf{X}_2)$ are below the γ_1 threshold, we have that $N_1 = 2$. The elite points $\mathbf{X}_1$ and $\mathbf{X}_2$ are the starting points of two Markov chains, whose stationary pdf is f_1. The length of each Markov chain is called the *splitting factor*. In this case, the GS algorithm uses the *same* splitting factor, 5, for each chain. This is called GS with *Fixed Splitting*.

Thus, we have two Markov chains on the γ_1-level set that start from $\mathbf{X}_1$ and $\mathbf{X}_2$ respectively and run for 5 steps, which are plotted in thicker lines. For the Markov chain starting from the point $\mathbf{X}_1$, two of five points have entered the γ_2-level set, say $\mathbf{X}_{1,3}$, $\mathbf{X}_{1,4}$, while only one point of the Markov chain starting at $\mathbf{X}_2$ has reached the next level, namely, $\mathbf{X}_{2,2}$. So, $N_2 = 3$. In the final stage, we start three independent Markov chains (of length 5) on the γ_2-level set from points $\mathbf{X}_{1,3}$, $\mathbf{X}_{1,4}$ and $\mathbf{X}_{2,2}$ with the stationary pdf f_2. Of all the points generated in the last stage, four have reached the final level set, so $N_3 = 4$.

In practice γ^* is not known and therefore one cannot determine the intermediate levels beforehand. Instead, one can determine them adaptively via the ADAM algorithm. This involves a *rarity parameter* ϱ. Having again an initial sample set $\mathcal{X}_0$ from f_0, the ADAM algorithm modifies the two steps of the GS algorithm as follows:

(a) Calculate the function value $S(\mathbf{x})$ for each $\mathbf{x} \in \mathcal{X}_t$ and sort these