European Journal of Operational Research 247 (2015) 711–720

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Continuous Optimization

A multi-layer line search method to improve the initialization of optimization algorithms

Benjamin Ivorra[a,*], Bijan Mohammadi[b], Angel Manuel Ramos[a]

[a] Departamento de Matemática Aplicada & Instituto de Matemática Interdisciplinar, Universidad Complutense de Madrid, Plaza de Ciencias, 3, 28040 Madrid, Spain

[b] Institut de Mathématiques et de Modélisation de Montpellier, Université de Montpellier II, Place Eugène Bataillon, 34095 Montpellier, France

ARTICLE INFO

Article history:
Received 20 March 2014
Accepted 17 June 2015
Available online 23 June 2015

Keywords:
Metaheuristics
Global optimization
Multi-layer line search algorithms
Evolutionary algorithms
Gradient methods

ABSTRACT

We introduce a novel metaheuristic methodology to improve the initialization of a given deterministic or stochastic optimization algorithm. Our objective is to improve the performance of the considered algorithm, called core optimization algorithm, by reducing its number of cost function evaluations, by increasing its success rate and by boosting the precision of its results. In our approach, the core optimization is considered as a sub-optimization problem for a multi-layer line search method. The approach is presented and implemented for various particular core optimization algorithms: Steepest Descent, Heavy-Ball, Genetic Algorithm, Differential Evolution and Controlled Random Search. We validate our methodology by considering a set of low and high dimensional benchmark problems (i.e., problems of dimension between 2 and 1000). The results are compared to those obtained with the core optimization algorithms alone and with two additional global optimization methods (Direct Tabu Search and Continuous Greedy Randomized Adaptive Search). These latter also aim at improving the initial condition for the core algorithms. The numerical results seem to indicate that our approach improves the performances of the core optimization algorithms and allows to generate algorithms more efficient than the other optimization methods studied here. A Matlab optimization package called "Global Optimization Platform" (GOP), implementing the algorithms presented here, has been developed and can be downloaded at: http://www.mat.ucm.es/momat/software.htm

1. Introduction

We consider a general optimization problem of the form:

$$\min_{x\in\Omega} h_0(x) \tag{1}$$

where $h_0 : \Omega \to \mathbb{R}$ is the cost function, x is the optimization parameter and $\Omega \subset \mathbb{R}^N$, with $N \in \mathbb{N}$, is the admissible space.

When solving (1) by an iterative procedure the choice of the initial condition is essential. For instance, this is the case with the gradient methods such as the Steepest Descent algorithm (**SD**) (Luenberger & Ye, 2008), the Newton algorithm (Polyak, 2007) or with the Heavy-Ball algorithm (**HB**) (Attouch, Goudou, & Redont, 2000). When h_0 has several local minima these algorithms converge to one of those depending on their initialization. However, these algorithms can still find the global optimum if the initial condition belongs to the attraction basin of the infimum. Another example where the initialization is of prime importance is with Genetics Algorithms (**GA**) (Goldberg, 1989; Gonalves, de Magalhes Mendes, & Resende, 2005) where a lack of diversity in the individuals of the initial population can result to a premature convergence to a local minimum of h_0 (Rocha, Neves, & Ali, 1999).

Thus, developing methods that intend to generate suitable initial conditions is interesting in order to improve the efficiency of existing optimization methods. For a given convergence accuracy, a better initialization may lead to a reduction in the number of functional evaluations, which is particularly important when working with expensive functional evaluations as in industrial design problems (Carrasco, Ivorra, & Ramos, 2012, 2015; Gomez, Ivorra, & Ramos, 2011; Ivorra, Hertzog, Mohammadi, & Santiago, 2006; Ivorra, Mohammadi, & Ramos, 2009, 2014; Ivorra, Redondo, Santiago, Ortigosa, & Ramos, 2013; Muyl, Dumas, & Herbert, 2004).

The idea of improving optimization algorithms by choosing a suitable initialization is widely present in the literature. For instance, the Direct Tabu Search algorithm (**DTS**) (Hedar & Fukushima, 2006; Lamghari & Dimitrakopoulos, 2012) and the Tunneling algorithm (Gomez & Levy, 1982; Levy & Gomez., 1985), are based on a

* Corresponding author. Tel.: +34 91 394 4415.
E-mail address: ivorra@mat.ucm.es (B. Ivorra).

http://dx.doi.org/10.1016/j.ejor.2015.06.044
0377-2217/

modification of the functional by the addition of penalty terms to avoid the algorithm to revisit previously explored regions. Other techniques, like the Greedy Randomized Adaptive Search Procedure (**GRASP**) (Hirsch, Pardalos, & Resende, 2010; Mart, Campos, Resende, & Duarte, 2015) or the Universal Evolutionary Global Optimizer (Redondo, Fernández, García, & Ortigosa, 2009) are based on the construction of a greedy solution combined with a local search step.

Another technique consists in coupling line search methods (Luenberger & Ye, 2008; Vieira & Lisboa, 2014) with another optimization algorithm. For instance, in Gardeux, Chelouah, Siarry, and Glover (2011) the authors propose an optimization method, called **EM323**, well suited for the solution of high-dimensional continuous non-linear optimization problems. The algorithm **EM323** consists in combining the Enhanced Unidirectional Search method (Gardeux, Chelouah, Siarry, & Glover, 2009) with the 3-2-3 line search procedure (Glover, 2010). Another example can be found in Grosan, Abraham, de Mello, and Yang (2007), in the context of Multi-Objective optimization problems. The authors develop a method combining several line search algorithms: one for determining a first point in the Pareto front and another one for exploring the front.

In this work, we propose a novel metaheuristic technique also based on line search methods to dynamically improve the initialization of a given optimization method. The paper is organized as follows. In Section 2 we reformulate problem (1) as a sub-optimization problem where the initial condition of the considered optimization algorithm is the optimization parameter. This new problem is solved by considering an original multi-layer semi-deterministic line search algorithm. In Section 3, we focus on the implementation of our approach by considering two families of optimization algorithms: descent methods (in particular, **SD** and **HB**) and Evolutionary Algorithms (in particular, **GAs**, Controlled Random Search algorithms (**CRS**) (Price, 1983) and Differential Evolution algorithms (**DE**) (Price, Storn, & Lampinen, 2005)). In Section 4, we validate our approach by considering various test cases in both low (Floudas & Pardalos, 1999) and high (Li, Engelbrecht, & Epitropakis, 2013) dimensions. The results are then compared with those given by the following optimization algorithms: **SD**, **HB**, **DTS**, Continuous **GRASP** (**CGR**), **CRS**, **DE** and **GA**.

2. General optimization method

We consider an optimization algorithm $A_0 : V \to \Omega$, called core optimization algorithm (**COA**), to solve problem (1). Here, V is the space where we can choose the initial condition for A_0 (various examples are given in Section 3, for simplicity we can consider $V = \Omega$). The other optimization parameters of A_0 (such as the stopping criterion, the number of iterations, etc.) are fixed by the user. We omit them in the presentation in order to simplify the notations.

We assume the existence of $v \in V$ such that, for a given precision $\epsilon \geq 0$, $h_0(A_0(v)) - \min_{x\in\Omega} h_0(x) < \epsilon$. Thus, solving problem (1) with algorithm **COA** means:

$$\text{Find } v \in V \text{ such that } A_0(v) \in \text{argmin}_{x\in\Omega} h_0(x). \quad (2)$$

In order to solve problem (2), we propose to use a multi-layer semi-deterministic algorithm (called in the sequel the Multi-Layer Algorithm and denoted by **MLA**) based on line search methods (see, for instance, Luenberger & Ye, 2008; Mohammadi & Saïac, 2003; Vieira & Lisboa, 2014).

More precisely, we introduce $h_1 : V \to \mathbb{R}$ as:

$$h_1(v) = h_0(A_0(v)). \quad (3)$$

Thus, problem (2) can be rewritten as

$$\text{Find } v \in V \text{ such that } v \in \text{argmin}_{w\in V} h_1(w). \quad (4)$$

A geometrical representation of $h_1(.)$ in one dimension is shown in Fig. 1 for a situation where the **COA** is the **SD** applied with 10,000 iterations, $\Omega = V = [-10, 6]$ and $h_0(x) = \frac{1}{2}\cos(2x) + \sin(\frac{1}{3}x) + 1.57$. We see that $h_1(.)$ is discontinuous with plateaus. Indeed, the same solution is reached by the algorithm starting from any of the points of the same attraction basin. Furthermore, $h_1(.)$ is discontinuous where the functional reaches a local maximum. One way to minimize such kind of functionals in the one dimensional case is to consider line search optimization methods (such as the secant or the dichotomy methods, see Mohammadi & Saïac, 2003).

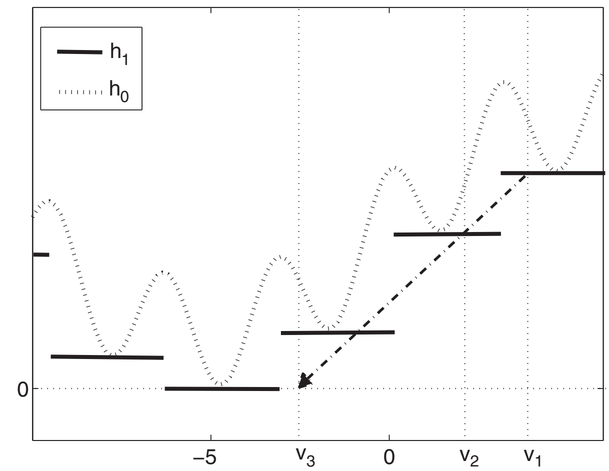

Fig. 1. **(Dotted line)** Graphical representation of $h_0(x) = \frac{1}{2}\cos(2x) + \sin(\frac{1}{3}x) + 1.57$, for $x \in \Omega = V = [-10, 6]$. **(Continuous line)** Graphical representation of $h_1(.)$ when the **SD** is used as **COA** with 10,000 iterations. **(Slash-dotted line)** Graphical representation of one execution of the algorithm $A_1(v_1)$, described in Section 3.1.1, when v_1 is given and $t_1 = 1$. v_2 is generated randomly in $[-10, 6]$ in the first Step of the algorithm. v_3 is built by the secant method performed in Step 2.2. v_3 (the best initial condition) is returned as the output in Step 3, since $h_1(v_3)$ is lower than $h_1(v_1)$ and $h_1(v_2)$.

Thus, in order to solve problem (4), we introduce the algorithm $A_1 : V \to V$ which, for any $v_1 \in V$, returns $A_1(v_1) \in V$ after the following steps:

Step 1- Choose v_2 randomly in V.
Step 2- Find $v \in \text{argmin}_{w\in\mathcal{O}(v_1,v_2)} h_1(w)$, where $\mathcal{O}(v_1, v_2) = \{v_1 + t(v_2 - v_1), t \in \mathbb{R}\} \cap V$, using a line search method.
Step 3- Return v.

The user may choose of the line search minimization algorithm in A_1.

This construction can be pursued looking for an optimal initialization for A_1. This can be done adding an external layer to algorithm A_1 and introducing $h_2 : V \to \mathbb{R}$ defined by

$$h_2(v) = h_1(A_1(v)) \quad (5)$$

and considering the following problem:

$$\text{Find } v \in V \text{ such that } v \in \text{argmin}_{w\in V} h_2(w). \quad (6)$$

To solve problem (6), we use the two-layers algorithm $A_2 : V \to V$ that, for each $v_1 \in V$, returns $A_2(v_1) \in V$ given by

Step 1- Choose v_2 randomly in V.
Step 2- Find $v \in \text{argmin}_{w\in\mathcal{O}(v_1,v_2)} h_2(w)$ using a line search method.
Step 3- Return v.

As previously, the user may choose the line search minimization algorithm in A_2. Due to the fact that the line search direction $\mathcal{O}(v_1, v_2)$ in A_1 is constructed randomly, the algorithm A_2 performs a multi-directional search of the solution of problem (2).

This construction can be pursued recursively defining

$$h_i(v) = h_{i-1}(A_{i-1}(v)), \quad \text{for} \quad i \in \mathbb{N}, \quad (7)$$

and considering the problem

$$\text{Find } v \in V \text{ such that } v \in \text{argmin}_{w\in V} h_i(w). \quad (8)$$