Expert Systems with Applications 42 (2015) 6777–6783

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Adaptive Penalty and Barrier function based on Fuzzy Logic

João Matias [a], Aldina Correia [a,b], Pedro Mestre [c,d,*], Carlos Serodio [c,d], Pedro Couto [c,d], Christophe Teixeira [e], Pedro Melo-Pinto [c,d]

[a] CM – Centre for Mathematics, Universidade de Trás-os-Montes e Alto Douro, UTAD, Escola de Ciências e Tecnologia, Quinta de Prados, 5000-801 Vila Real, Portugal[1]
[b] CIICESI – Center for Research and Innovation in Business Sciences and Information Systems, School of Technology and Management of Felgueiras, Porto Polytechnic, Portugal
[c] CITAB – Centre for the Research and Technology of Agro-Environment and Biological Sciences, Universidade de Trás-os-Montes e Alto Douro, UTAD, Escola de Ciências e Tecnologia, Quinta de Prados, 5000-801 Vila Real, Portugal[1]
[d] R&D Centre Algoritmi, University of Minho, Guimarães, Portugal
[e] Universidade de Trás-os-Montes e Alto Douro, UTAD, Quinta de Prados, 5000-801 Vila Real, Portugal[1]

A R T I C L E   I N F O

Article history:
Available online 5 May 2015

Keywords:
Applications
Fuzzy mathematical programming
Mathematics
Derivative free optimization
Direct search methods
Penalty and Barrier functions
Fuzzy Logic

A B S T R A C T

Optimization methods have been used in many areas of knowledge, such as Engineering, Statistics, Chemistry, among others, to solve optimization problems. In many cases it is not possible to use derivative methods, due to the characteristics of the problem to be solved and/or its constraints, for example if the involved functions are non-smooth and/or their derivatives are not know. To solve this type of problems a Java based API has been implemented, which includes only derivative-free optimization methods, and that can be used to solve both constrained and unconstrained problems. For solving constrained problems, the classic Penalty and Barrier functions were included in the API. In this paper a new approach to Penalty and Barrier functions, based on Fuzzy Logic, is proposed. Two penalty functions, that impose a progressive penalization to solutions that violate the constraints, are discussed. The implemented functions impose a low penalization when the violation of the constraints is low and a heavy penalty when the violation is high. Numerical results, obtained using twenty-eight test problems, comparing the proposed Fuzzy Logic based functions to six of the classic Penalty and Barrier functions are presented. Considering the achieved results, it can be concluded that the proposed penalty functions besides being very robust also have a very good performance.

1. Introduction

Optimization, also known as Mathematical Programming, is used in many decision making processes. In these processes the main objective is to determine the best use of available resources in order to obtain the best results for a given reality. So, optimization has been used in many scientific areas such as Engineering, Statistics and Chemistry, among others.

Problems are defined by models consisting of one or more functions (which need to be minimized or maximized), called objective function, and at least one variable, the decision variable(s). An unconstrained optimization problem can therefore be defined as in (1).

$$\min_{x \in \mathbb{R}^n} \quad f(x) \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is the objective function.

In some optimization problems the objective function variables may be subject to some conditions, defined by the problem constraint functions. These constrained problems can be defined as in (2):

$$\begin{aligned} &\min_{x \in \mathbb{R}^n} f(x) \\ \text{subject to } & c_i(x) = 0, \; i \in \mathcal{E} \\ & c_i(x) \leqslant 0, \; i \in \mathcal{I} \end{aligned} \tag{2}$$

where:

- $f : \mathbb{R}^n \to \mathbb{R}$ is the objective function;
- $c_i(x) = 0$, $i \in \mathcal{E}$, with $\mathcal{E} = \{1, 2, \ldots, t\}$, define the problem equality constraints;

[1] www.utad.pt.

* Corresponding author at: CITAB – Centre for the Research and Technology of Agro-Environment and Biological Sciences, Universidade de Trás-os-Montes e Alto Douro, UTAD, Escola de Ciências e Tecnologia, Quinta de Prados, 5000-801 Vila Real, Portugal.

E-mail addresses: j_matias@utad.pt (J. Matias), aic@estgf.ipp.pt (A. Correia), pmestre@utad.pt (P. Mestre), cserodio@utad.pt (C. Serodio), pcouto@utad.pt (P. Couto), christopheteixeira4@gmail.com (C. Teixeira), pmelo@utad.pt (P. Melo-Pinto).

http://dx.doi.org/10.1016/j.eswa.2015.04.070
0957-4174/

- $c_i(x) \leqslant 0, i \in \mathcal{I}$, with $\mathcal{I} = \{t+1, t+2, \ldots, m\}$, represent the inequality constraints;
- $\Omega = \{x \in \mathbb{R}^n : c_i = 0, i \in \mathcal{E} \wedge c_i(x) \leqslant 0, i \in \mathcal{I}\}$ is the set of all feasible points, i.e., the feasible region.

Unconstrained and constrained optimization problems can be found in many real life problems, and in many cases: the objective function values and/or its constraints are the result of complex and time consuming simulations; its values are obtained experimentally or by natural phenomena observation; the analytic functions might be too complex or even not be available; the samples have noise. There are also some cases where the objective function is non-smooth or non-differentiable, or even non-continuous. In such cases derivative based methods cannot be used to solve these problems, as presented in Conn, Scheinberg, and Vicente (2009).

Some of the possible solutions, whenever this type of problem must be solved, include the use of heuristic methods such as particle swarm (Wu et al., 2014), hybrid methods, e.g. particle swarm method followed by a global minimization method (Vaz & Vicente, 2007, 2009), tabu search and simulated annealing algorithms as presented in Hedar et al. (2002), Hedar and Fukushima (2006), or genetic algorithms (Boudjelaba, Ros, & Chikouche, 2014).

These methods can be used if the cost of the objective function evaluation is negligible, otherwise they must be avoided. Instead, other derivative free algorithms, namely deterministics algorithms, can be used. Such algorithms include direct search methods that do not use derivatives or approximations to them (Lewis, Torczon, & Trosset, 2000; Kolda, Lewis, & Torczon, 2003; Hooke & Jeeves, 1961).

In Correia, Matias, Mestre, and Serodio (2010), Mestre, Matias, Correia, and Serodio (2010) it was presented a Java API (Application Programming Interface), with remote access, which includes only Direct Search Optimization Methods for solving both unconstrained and constrained optimization problems. The objective of this API is to be included in other software packages, to solve problems where derivative based methods cannot be used. It was used in location estimation problems by the authors in Mestre et al. (2012, 2013) to tune the LEA (Location Estimation Algorithm) and adapt them to the mobile terminals. While in Mestre et al. (2012) a Fuzzy Logic based LEA was implemented and the API was used to tune the parameters/transitions of membership functions and adjust the weights of OWA (Ordered Weighted Averaging), in Mestre et al. (2013) the API was used to tune the internal parameters of the Weighted k-Nearest Neighbour algorithm and a scaling factor for the RSSI (Received Signal Strength) values. In both cases, because of the nature of the data, derivative based methods could not be used.

In this context, the most used techniques to solve constrained problems consist on transforming constrained problems into unconstrained problems that are easier to solve, which solution is related with the solution of the original problem. One of such techniques consists in using Penalty and Barrier methods, which are used in this work.

Penalty and Barrier functions have been widely used and studied in the last years, for example by Byrd, Nocedal, and Waltz (2008), Chen and Goldfarb (2006), Fletcher (1997), Gould, Orban, and Toint (2003), Leyffer, Calva, and Nocedal (2006), Klatte and Kummer (2002), Mongeau and Sartenaer (1995) and Zaslavski (2005), due to its ability to deal with Degenerated Problems.

Exact Penalty Methods have been successfully used to solve Mathematical Programs with Complementary Constraints, by Benson, Sen, Shanno, and Vanderbei (2006), Leyffer et al. (2006), Rodrigues and Monteiro (2006) and Rodrigues, Monteiro, and Vaz (2009). They were also used in Constrained NonLinear Programming to assure the admissibility of sub-problems and the iteration reliability by Byrd et al. (2008) and Chen and Goldfarb (2006).

Combination of Penalty Methods and Fuzzy concepts have been used by several authors such as Bustince, Jurio, Pradera, Mesiar, and Beliakov (2013), Chen, Pi, and Liu (2013), Gouicem, Benmahammed, Drai, Yahi, and Taleb-Ahmed (2012), Bogdana and Milan (2009) and Jamison and Lodwick (2001). In Bustince et al. (2013) the concept of penalty function is used to determine which aggregation function should be applied, i.e., the objective is to choose which aggregation function will output the best results. Chen et al. (2013) use Penalty concepts to construct a new objective function and avoid monotonicity and coincident clustering results. In Gouicem et al. (2012), Fuzzy is used for image reconstruction and Penalty is used in edge detection to penalise pixels for which it was not detected an edge. Penalty is used to transform Linear constrained problems into unconstrained problems by Bogdana and Milan (2009) and Jamison and Lodwick (2001).

These authors add the Penalty concept to Fuzzy, i.e., Penalty is used as an auxiliary tool used together with Fuzzy algorithms. The objective of the present work is different: use Fuzzy as a tool to generate Penalty functions that can be used together with Direct Search Optimization Methods. An improved version of a method presented by the authors in Matias et al. (2012), and a new Penalty function based on Fuzzy Logic, which adapts the penalty to apply based on its previous value, are presented. By using Fuzzy Logic to generate a Penalty Function it is possible to do a progressive penalization of the objective function when the problem constraints are violated.

With these two penalty functions it is expected to obtain a better performance of direct search methods, i.e., a lower number of objective function evaluations.

2. Penalty and Barrier functions

Let us consider the above presented constrained problem (2), Penalty and Barrier methods were developed to solve such problems by solving a sequence of unconstrained problems, i.e., a transformation of the original problem into a sequence of unconstrained problems is made.

Penalty and Barrier Methods comprise two processes (Fig. 1):

- External Process (EP) – where a succession of Unconstrained Optimization Problems is created from a constrained problem;
- Internal Process (IP) – where each of the previously generated problems, the Unconstrained Optimization Problems are solved.

When Penalty and Barrier functions are used a new objective function, Φ_k, based on information from the original problem is created. Therefore a succession of Unconstrained Optimization Problems are obtained. These new problems depend on a positive parameter, r_k (Penalty/Barrier parameter) which solutions $x^*(r_k)$ converge to the solution of the original problems x^* (*External Process*).

Direct Search methods are then used to solve the resulting Unconstrained Optimization Problems (*Internal Process*). At each iteration k the problem to be solved by the Internal Process is:

$$\min_{x_k \in \mathbb{R}^n} \Phi(x_k, r_k) = \min_{x_k \in \mathbb{R}^n} (f(x_k) + r_k p(x)) \tag{3}$$

where r is a penalty parameter and p is a Penalty/Barrier function that penalises/refuses points that lie outside the feasible region.

Barrier methods can be used only when we have an initial feasible point. These methods were widely presented in Doyle (2003).

According to Freund (2004) a barrier function can be defined as a function $b : \mathbb{R}^n \to \mathbb{R}$, that satisfies the following conditions: