Available online at www.sciencedirect.com
SCIENCE @ DIRECT®
Mathematical and Computer Modelling 40 (2004) 63–76
MATHEMATICAL AND COMPUTER MODELLING
www.elsevier.com/locate/mcm

Cubic Algorithm for Global Optimization with Box and Equality Constraints and Application to Optimal Allocation of Resources

E. A. GALPERIN†
Département de Mathématiques
Université du Québec à Montréal
C.P. 8888, Succ. Centre-Ville
Montréal, Québec, Canada H3C 3P8
galperin.efim@uqam.ca

P. YA. EKEL‡ AND J. G. PEREIRA, JR.‡
Programa de Pós-Graduação em Engenharia Elétrica
Pontifícia Universidade Católica de Minas Gerais
Av. Dom José Gaspar, 500
30.535-610, Belo Horizonte, MG, Brasil
ekel@pucminas.br joelgpj@hotmail.com

(Received and accepted December 2003)

Abstract—C++ code of the cubic algorithm [1] is proposed for nonconvex global optimization of continuous functions with box and equality constraints, and for the solution of nonlinear systems of equations, including underdetermined and overdetermined systems. The algorithm is upgraded for application to problems with linear and nonlinear equality constraints usually encountered in optimal resource allocation, investment planning, construction projects, and many engineering applications. It is further extended for visualization analysis of high dimensional sets of global optimizers by means of projections on linear subspaces and on arbitrary planes. C++ windows are presented for input, output, algorithmic control, and special research possibilities including graphics and screen display. The code does not create ill-conditioned situations.

Keywords—Nonconvex global optimization, Cubic algorithm, Numerical methods.

1. INTRODUCTION

In this paper, the cubic algorithm [1] is upgraded and extended to include effective separate treatment of equality constraints, and other noteworthy features for analysis of the set of global optimal solutions.

There exist codes for the cubic algorithm in MATLAB for dim = 1, see [2], and in MAPLE for arbitrary dimension n, see [3]. A code in (object oriented) C++ for this version of the cubic algorithm allows one to use vast libraries developed by Microsoft and provides convenient Window

†Research of this author was partially supported by the NSERC Grant No. RGPIN-3492-00.
‡Research of these authors was supported by the CNPq Grant No. 550408-2002/9 and the PUC-MG Grant No. FIP 2003/46-TLE.

0895-7177/04/$ - see front matter Typeset by $\mathcal{A}_{\mathcal{M}}\mathcal{S}$-TEX
doi:10.1016/j.mcm.2003.12.003

presentation at any iteration, see Section 6. New features of this code include linear subspace projection and intersection blocks and plane intersection block which provide the possibility for analysis and visualization according to the needs of the user in a practical application. The code can be further upgraded for nonrectangular inf or sup-compact sets in R^n of any nature (non-robust, disconnected) given by a system of inequalities using the beta-algorithm [1, pp. 79–134] that sifts out all sets of infeasible points without loss of global optimizers.

The code solves a problem of finding the global minimum value

$$s^0 = \inf f(x), \qquad x \in \bar{X} \subset R^n, \tag{1.1}$$

over the n-rectangle (box)

$$\bar{X} = \{x \in R^n : a_i \le x_i \le b_i,\ i = 1, \dots, n\}, \tag{1.2}$$

and the set of all global minimizers

$$X^0 = \{x \in \bar{X} : f(x) = s^0\}. \tag{1.3}$$

Here $a_i < b_i$, $\forall i$, the bar means closure, and the function $f(x)$ is supposed to be Lipschitz continuous over $\bar{X}$, that is

$$|f(x) - f(x')| \le A\,\|x - x'\|, \qquad \forall x, x' \in \bar{X}. \tag{1.4}$$

If the minimal Lipschitz constant over $\bar{X}$ is known (the symbol ∇ means gradient)

$$L = \max \|\nabla f(x)\|, \qquad x \in \bar{X}, \tag{1.5}$$

then one can take in (1.4) the value $A = L$ in which case (1.4) is nonimprovable over $\bar{X}$.

A function $f(x)$ is not supposed to be given as a formula, it is only assumed to be computable. In such cases, the constant L of (1.5) is unknown and difficult to determine. Moreover, a subset of $\bar{X}$ can have a smaller L than defined by (1.5) over the entire $\bar{X}$. For these reasons, we do not associate A in (1.4) with the value in (1.5) and treat it as a bound on the slopes of $f(x)$ within $\bar{X}$. This has transparent meaning if we consider the angle

$$\alpha = \arg\tan A, \qquad 0 < \alpha < \frac{\pi}{2}. \tag{1.6}$$

The reader can check that, if (1.4) is met, then in any plane section of $\bar{X}$ passing through x, x' the slope of $f(x)$ is not greater than α. Table 1 provides the visualization for the notion of a function of bounded slope.

Table 1.

α	0	30°	45°	60°	63°	71°	76°	78°	80°
A	0	$\frac{1}{\sqrt{3}}$	1	$\sqrt{3}$	2	3	4	5	6

For $A > 3$, slopes are very steep, and scaling should be applied to assure reasonable computer time for the procedure.

For universality of the code, it is expedient to transform a problem (1.1)–(1.3) into the unit cube $\bar{U}$, axes oriented, with the edge $c = 1$ and a vertex at the origin. It is formed by the unit vectors of the Cartesian coordinate system which is supposed to be the reference system in (1.2).

The linear transformation

$$x_i = a_i + (b_i - a_i) z_i, \qquad i = 1, \dots, n, \tag{1.7}$$