

An always convergent algorithm for global minimization of multivariable continuous functions

J. Abaffy, A. Galántai

Óbuda University

John von Neumann Faculty of Informatics

1034 Budapest, Bécsi u. 96/b, Hungary

abaffy.jozsef@nik.uni-obuda.hu

galantai.aurel@nik.uni-obuda.hu

Abstract: We develop and test a Bolzano or bisection type global optimization algorithm for continuous real functions over a rectangle. The suggested method combines the branch and bound technique with an always convergent solver of underdetermined nonlinear equations. The numerical testing of the algorithm is discussed in detail.

Keywords: global optimum, nonlinear equation, always convergent method, Newton method, branch and bound algorithms, Lipschitz functions

1 Introduction

In this paper we study the minimization problem

$$f(x) \rightarrow \min \quad (f : \mathbb{R}^n \rightarrow \mathbb{R}, x \in X = \times_{i=1}^n [l_i, u_i]) \quad (1)$$

with $f \in C(X)$, and develop a method to find its global minimum. Assume that

$$[x_{sol}, iflag] = \text{equation_solve}(f, c) \quad (2)$$

denotes a solution algorithm for the single multivariate equation

$$f(x) = c \quad (x \in X) \quad (3)$$

such that $iflag = 1$, if a true solution $x_{sol} \in X$ exists (that is $f(x_{sol}) = c$), and $iflag = -1$, otherwise.

Let $f_{\min} = \min \{f(x) \mid x \in X\}$ be the global minimum of f , and let $b_1 \in \mathbb{R}$ any lower bound of f such that $f_{\min} \geq b_1$. Let $z_0 \in D_f$ be any initial approximation to the global minimum point ($f(z_0) \geq b_1$). The suggested algorithm then takes the form:

```

Data:  $a_1 = f(z_1), b_1, i = 1$ 
1 while  $a_i - b_i > tol$  do
2    $c_i = (a_i + b_i) / 2$ 
3    $[\xi, iflag] = \text{equation\_solve}(f, c_i);$ 
4   if  $iflag = 1$  then
5      $z_{i+1} = \xi, a_{i+1} = f(\xi), b_{i+1} = b_i;$ 
6   else
7      $z_{i+1} = z_i, a_{i+1} = a_i, b_{i+1} = c_i;$ 
8   end
9    $i = i + 1$ 
10 end

```

Algorithm 1.

Using the idea of Algorithm 1 we can also determine a lower bound of f , if such a bound is not known a priori (see later or [1]). Algorithm 1 has certain conceptual similarities with the bisection algorithms of Shary [30], [31] and Wood [40], [41].

Theorem 1. *Assume that $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is continuous and bounded from below by b_1 . Then Algorithm 1 is globally convergent in the sense that $f(z_i) \rightarrow f_{\min}$.*

Proof. At the start we have z_1 and the lower bound b_1 such that $f(z_1) \geq b_1$. Then we take the midpoint of this interval, i.e. $c_1 = (f(z_1) + b_1) / 2$. If a solution ξ exists such that $f(\xi) = c_1$ ($iflag = 1$), then $c_1 = f(z_2) \geq f_{\min} \geq b_1$ holds by the initial assumptions. If there is no solution of $f(\xi) = c_1$ (i.e. $iflag = -1$), then $c_1 < f_{\min}$. By continuing this way we always halve the inclusion interval $(b_i, f(z_i))$ for f_{\min} . Hence the method is convergent in the sense that $f(z_i) \rightarrow f_{\min}$. ■

Note that sequence $\{z_i\}$ is not necessarily convergent.

The performance of Algorithm 1 clearly depends on the equation solver, which for $n > 1$, has to solve a sequence of underdetermined equations of the form (3).

In paper [1] we tested a version of Algorithm 1 that used a locally convergent nonlinear Kaczmarz method [38], [23], [24], [22] and a local minimizer for acceleration as well. The algorithm showed fast convergence in most of the test problems, but in some cases it also showed numerical instability, when $\|\nabla f(z_k)\|$ was close to zero. This and later experiments indicated that only "globally convergent" and gradient free solvers are useful in the above scheme at the price of losing speed.

Hence in [2], for one dimensional Lipschitz functions, we developed and successfully tested a version of Algorithm 1 that is based on an always convergent iteration method of Szabó [36], [37].

Here we investigate two versions of Algorithm 1 that use an always convergent iteration method (Galántai [14]) for solving equations of the form (3). This solver is based on continuous space-filling curves lying in the rectangle X and it has a kind of monotone convergence to the nearest zero on the given curve, if it exists, or the iterations leave the region in a finite number of steps.

Definition 1. *Let $r : [0, 1] \rightarrow [0, 1]^n$ ($n \geq 2$) be a continuous mapping. The curve*

$r = r(t)$ ($t \in [0, 1]$) is space-filling if r is surjective.

Given a space-filling curve $r : [0, 1] \rightarrow [0, 1]^n$ and the rectangle $X = \times_{i=1}^n [l_i, u_i]$, the mapping

$$h_i(t) = (u_i - l_i) r_i(t) + l_i, \quad i = 1, \dots, n$$

clearly fills up the whole rectangle X .

The use of space-filling curves in optimization was first suggested by Butz [5], [6], and later by Strongin and others (see, e.g. [34], [35], [32]).

These methods reduce problem (1) to the one dimensional problem

$$f(h(t)) \rightarrow \min \quad (t \in [0, 1])$$

using mainly the Hilbert space filling function and one dimensional global minimizers. We note that Butz [8] suggested the use of Hilbert's space-filling functions for solving nonlinear systems as well (see also [14]). However these dimension reduction type minimization methods are criticized by various authors pointing out the limited use, speed and other matters (see, e.g. Törn and Zilinskas [39] or Pintér [28]). Using complexity results of Nemirovsky and Yudin [27] Goertzel [16] argues in favour of such methods if f is Lipschitz. For the global minimization of Lipschitz functions, see, e.g. Hansen, Jaumard, Lu [18], [19], [20], [21] and Pintér [28].

Our aim here is only to assess the feasibility and reliability of Algorithm 1 using space-filling based equation solvers, which seems to be a new approach.

Instead of space-filling curves we can also use α -dense curves introduced by Cheruault and Guillez (see, e.g. [9], [17] or [10]).

Definition 2. Let $I = [a, b] \subset \mathbb{R}$ be an interval and $X = \times_{i=1}^n [l_i, u_i] \subset \mathbb{R}^n$ be a rectangle. The map $x : I \rightarrow X$ is an α -dense curve, if for every $x \in X$, there exists a $t \in I$ such that $\|x(t) - x\| \leq \alpha$.

The α -dense curves are not space-filling functions. Note that the practical approximations of space-filling curves are also α -dense curves for some α . For 2D, the k th approximating polygon of the Hilbert curve is α -dense with $\alpha \leq \sqrt{2}/2^{2k}$ (see, e.g. Sagan [29]). Recently Mora [25] characterized the connection of space-filling and α -dense curves.

In the rest of the paper we define the class of always convergent methods for solving nonlinear equations in Section 2. Details and the results of numerical testing will be given in Section 3. The numerical testing was performed on a set of 2D Lipschitz continuous problems.

We close the paper with conclusions and the appendix of test problems.

2 Always convergent methods for nonlinear equations

Consider nonlinear equations of the form

$$f(x) = 0 \quad (f : \mathbb{R}^n \rightarrow \mathbb{R}^m, x \in X = \times_{i=1}^n [l_i, u_i]), \quad (4)$$

where f is continuous on the rectangle X .

Assume that a continuous curve $\Gamma = \{r(t) : 0 \leq t \leq 1\} \subset X$ is given. We seek for the solution of $f(x) = 0$ on the curve Γ , that is the solution of equation

$$f(r(t)) = 0 \quad (t \in [0, 1]), \quad (5)$$

which is equivalent to the real equation

$$\|f(r(t))\| = 0 \quad (t \in [0, 1]). \quad (6)$$

Theorem 2. (Galántai [14]). Assume that $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is continuous on the rectangle $X = \times_{i=1}^n [l_i, u_i]$ and $\Gamma = \{r(t) : 0 \leq t \leq 1\} \subset X$ is a continuous curve. Let ω_f and ω_r be the modulus of continuity of f on X and Γ on $[0, 1]$, respectively. Assume that $\rho_f, \rho_r : [0, \infty) \rightarrow [0, \infty)$ are continuous and strictly monotone increasing functions so that

$$\rho_f(0) = 0, \quad \rho_f(\delta) \geq \omega_f(\delta) \quad (\delta \in [0, \text{diam}(X)]), \quad \lim_{\delta \rightarrow \infty} \rho_f(\delta) = \infty \quad (7)$$

and

$$\rho_r(0) = 0, \quad \rho_r(\delta) \geq \omega_r(\delta) \quad (\delta \in [0, \tau]), \quad \lim_{\delta \rightarrow \infty} \rho_r(\delta) = \infty \quad (8)$$

hold, respectively. Furthermore assume that

(a) $F(x, y)$ is continuous in $[0, 1] \times [0, \infty)$;

(b) $x \geq 0, F(x, y) = x \Leftrightarrow y = 0$;

(c) $F(x, y) < x$ ($x \in [0, 1], y > 0$);

(d) For $x > \xi$ ($x, \xi \in [0, 1]$) and $0 \leq y \leq x - \xi, F(x, y) \geq \xi$.

(e) $F(x, y)$ is strictly monotone increasing in x , and strictly monotone decreasing in y ;

Define $\varphi(t) = \rho_r^{-1}(\rho_f^{-1}(\|f(r(t))\|))$ ($t \in [0, 1]$). Let $t_0 = 1$ and assume that $\varphi(1) > 0$. Define

$$t_{i+1} = F(t_i, \varphi(t_i)) \quad (i = 0, 1, 2, \dots). \quad (9)$$

Then $\{t_i\}$ is a strictly monotone decreasing sequence that converges to ξ_{\max} if a root ξ of $\|f(r(t))\| = 0$ exists in $[0, 1]$. If no root exists, then the sequence $\{t_i\}$ leaves the interval $[0, 1]$ in a finite number of steps.

For the proof of theorem, see [14] or [15]. If Γ is a space-filling curve, then the method clearly always convergent in the sense that it either converges to a solution (if exists) or it leaves the region in a finite number of iterations (if no solution exist). If one selects a curve Γ that is not space-filling, the algorithm may fail to find a zero. Note however that the space-filling functions used in practice are only approximations to the true ones.

A function f is said to be Lipschitz β ($0 < \beta \leq 1$) with the Lipschitz constant L , that is $f \in \text{Lip}_L \beta$, if

$$\|f(x) - f(y)\| \leq L \|x - y\|^\beta \quad (x, y \in D_f). \quad (10)$$

Assume that $f \in \text{Lip}_{L_f} \beta$ ($0 < \beta \leq 1$). Then $\omega_f(\delta) \leq L_f \delta^\beta$ and we can select $\rho_f(\delta) = L_f \delta^\beta$ and $\rho_f^{-1}(\delta) = \left(\frac{\delta}{L_f}\right)^{1/\beta}$. Similarly, if curve Γ is $\text{Lip}_{L_\Gamma} \mu$ ($\mu \in (0, 1]$), that is

$$\|r(s) - r(t)\| \leq L_\Gamma |s - t|^\mu \quad (t, s \in [0, \tau]), \quad (11)$$

then $\omega_r(\delta) \leq L_\Gamma \delta^\mu$ and so we can take $\rho_r(\delta) = L_\Gamma \delta^\mu$ and $\rho_r^{-1}(\delta) = \left(\frac{\delta}{L_\Gamma}\right)^{1/\mu}$. Thus

$$\varphi(t) = \rho_r^{-1} \rho_f^{-1} (\|f(r(t))\|) = \frac{1}{L_\Gamma^{1/\mu}} \left(\frac{\|f(r(t))\|}{L_f} \right)^{\frac{1}{\mu\beta}}. \quad (12)$$

Based upon the numerical testing [14] we select $F(x, y) = x - y$, and the method

$$t_{i+1} = t_i - \varphi(t_i) \quad (i = 0, 1, \dots). \quad (13)$$

Here we use the Hilbert space filling curve (see, e.g. [33], Butz [5], [7], [29], [3], [35], [32]).

Lemma 1. *The Hilbert mapping $r_H : [0, 1] \rightarrow [0, 1]^n$ is space-filling, nowhere differentiable and $\text{Lip}_K \mu$ with $L_\Gamma = 2\sqrt{n+3}$ and $\mu = 1/n$:*

$$\|r_H(s) - r_H(t)\| \leq L_\Gamma |s - t|^{1/n} \quad (s, t \in [0, 1]). \quad (14)$$

For a proof, see, e.g. [42]. For $n = 2$, the Lipschitz constant $L_\Gamma = 2\sqrt{5}$ can be replaced by the sharper value $L_\Gamma = \sqrt{6}$ (Bauman [4]). The following figure shows the recursive k th approximation of the Hilbert curve for $k = 6$.

Similarly to space-filling functions there are many α -dense curves (see, e.g. [10]). Here we use the α -dense curve of Cherruault [10] given by

$$x_i(t) = \frac{1}{2} (1 - \cos(\omega_i 2\pi t)), \quad i = 1, \dots, n \quad (15)$$

with $\omega_i = \sigma^i$ (for reasons, see [14]). For $n = 2$ and $\sigma = 1000$, $\alpha \approx 0.0044$. This curve is smooth ($\mu = 1$) unlike the Hilbert curve, but it has a huge Lipschitz constant (see, e.g. [14]). The following figure shows the Cherruault curve for $\sigma = 100$ in 777 points.

3 The numerical experiments

We tested two algorithms. Namely, Algorithm 1 with given lower estimates for the global minimum and the following modification of Algorithm 1 that constructs a lower bound for f_{\min} .

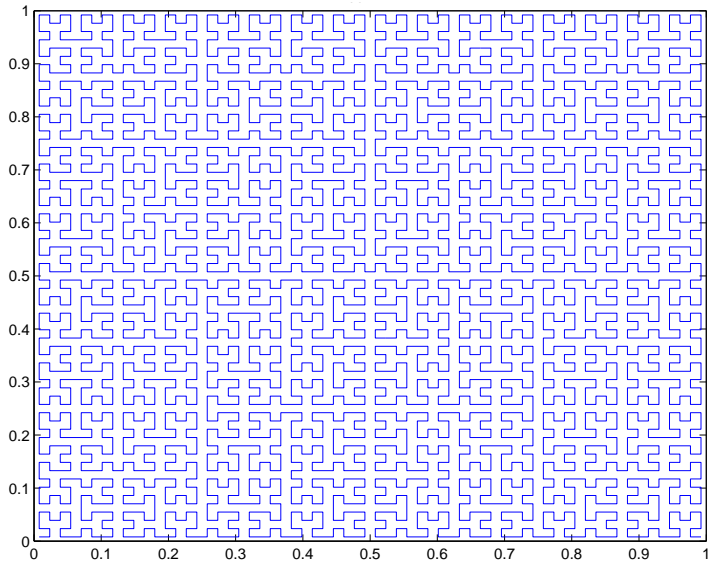


Figure 1
Hilbert curve approximation for $k = 6$.

```

Data:  $a_1 = f(z_1)$ ,  $iflag = 1$ ,  $d = 1$ ,  $c = a_1 - d$ 
1 while  $iflag = 1$  do
2    $[\xi, iflag] = \text{equation\_solve}(f, c)$ ;
3   if  $iflag = 1$  then
4      $a_1 = f(\xi)$ ,  $z_1 = \xi$ ,  $d = 2d$ ,  $c = a_1 - d$ ;
5   else
6      $b_1 = c$ ;
7   end
8 end

Data:  $i = 1$ 
9 while  $a_i - b_i > tol$  do
10   $c_i = (a_i + b_i) / 2$ 
11   $[\xi, iflag] = \text{equation\_solve}(f, c_i)$ ;
12  if  $iflag = 1$  then
13     $z_{i+1} = \xi$ ,  $a_{i+1} = f(\xi)$ ,  $b_{i+1} = b_i$ ;
14  else
15     $z_{i+1} = z_i$ ,  $a_{i+1} = a_i$ ,  $b_{i+1} = c_i$ ;
16  end
17   $i = i + 1$ 
18 end

```

Algorithm 2.

We used the numerical solver (13) with the exit condition

$$\|f(r(t_i))\| \leq tol \forall i = itmax. \quad (16)$$

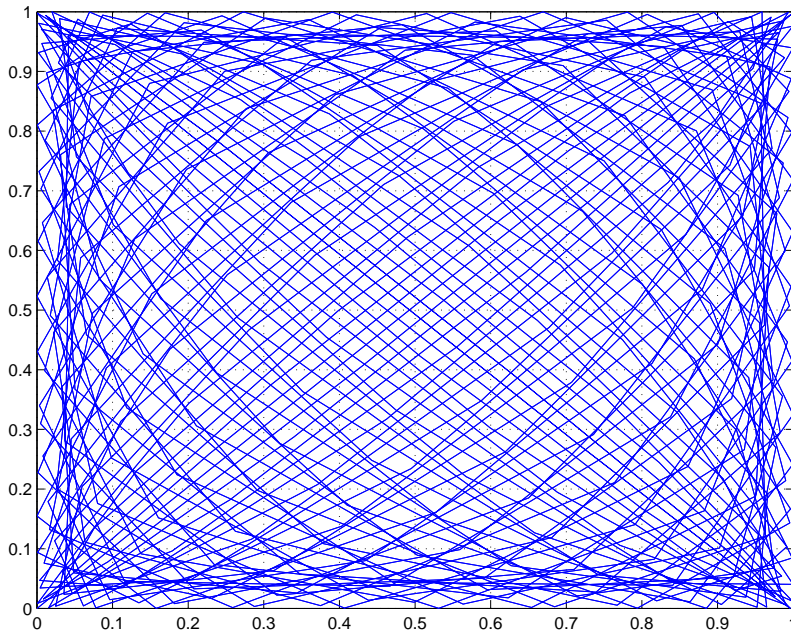


Figure 2
Cherruault 2D curve for $\sigma = 100$.

We selected a set of two dimensional Lipschitz 1 test problems whose Lipschitz constants were numerically estimated using standard techniques (see, e.g. [19], [28]).

We used two versions of equation solver (13): one that is based on Hilbert's space-filling curve and a second one that is based on Cherruault's α -dense curve (15).

For the computation of the 2D Hilbert curve we used the algorithm of page 52 of Bader [3] with $depth = 54$, that computes the points of the curve with an error proportional to $2^{-54} = 5.5511 \times 10^{-17}$.

Since the stepsize $\varphi(t_i)$ can be arbitrarily small, it is reasonable to impose the lower bound $\varphi(t_i) \geq \varepsilon_{machine}$ on the iterates t_i . For $f \in \text{Lip}_{L_f} 1$ and $r \in \text{Lip}_{L_r} \frac{1}{2}$, this holds if and only if $\|f(r(t_i))\| \geq L_f L_r \varepsilon_{machine}^{1/2} \approx 6.67 \times 10^{-8} L_f$ and we have the lower bound $tol \geq 6.67 \times 10^{-8} L_f$ for the tol parameter. The computer experiments of [14] and also of Butz [8] indicate that tol can not be too small. Here we selected $tol = 1e-3$ and $itmax = 1e+6$ for the Hilbert's curve based solver and $itmax = 1e+5$ for the α -dense based solver.

The computations were carried out in Matlab R2011b (64 bit) on a PC with Windows 7 operating system and Intel I7 processor.

The CPU times and absolute errors of Algorithms 1 and 2 using Hilbert's curve based solver (Bolzano-v1H, Bolzano-v2H) are shown on the following two figures.

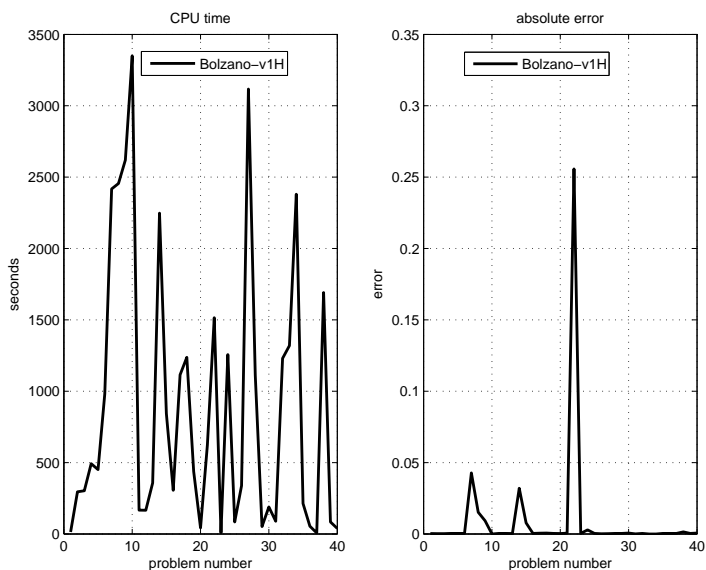


Figure 3

CPU time and absolute error of Algorithm 1 using Hilbert's curve based solver.

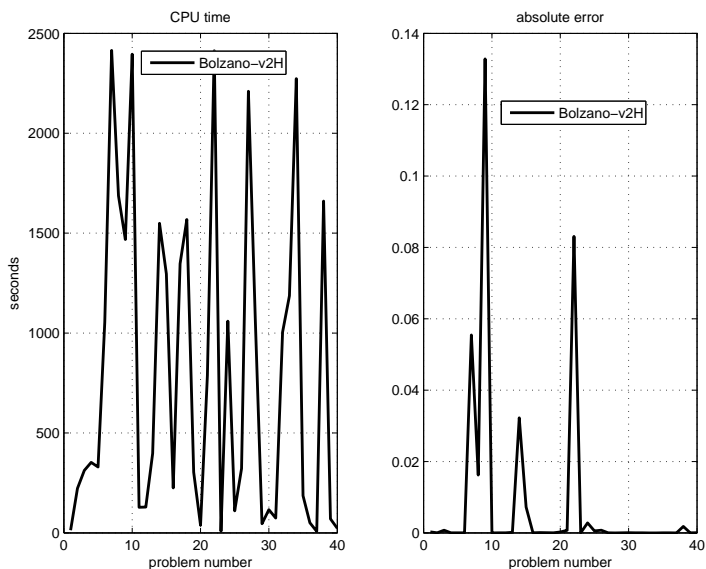


Figure 4

CPU time and absolute error of Algorithm 2 using Hilbert's curve based solver.

Here we can observe extremely big computational times for both algorithms (as expected) and only a few absolute errors greater than $10 \times tol$. The computational times of Algorithm 2 are somewhat less than in the case of Algorithm 1 (the average

CPU time of Algorithm 2 is 797.89 sec. in opposition to the average CPU time of Algorithm 1, which is 892.49 sec.). The absolute errors for Algorithm 1 exceed $10 \times tol = 1e - 2$ for the test problems number 7, 8, 14 and 22 while for Algorithm 2 the corresponding cases are the test problems number 7,8,9,14 and 22. A close inspection of these cases reveals that the stepsize $\varphi(t_i)$ of algorithm (13) become less than $\varepsilon_{machine}$, while t_i was much bigger (only for cases $c \approx f_{min}$). Hence $t_{i+1} = t_i$ was repeated due to the floating point arithmetic and it was stopped only by $itmax$. This problem can be overcome using multiple precision arithmetic.

The CPU times and absolute errors of Algorithms 1 and 2 using α -dense curve based solver (Bolzano-v1C, Bolzano-v2C) are shown on Figures 5 and 6.

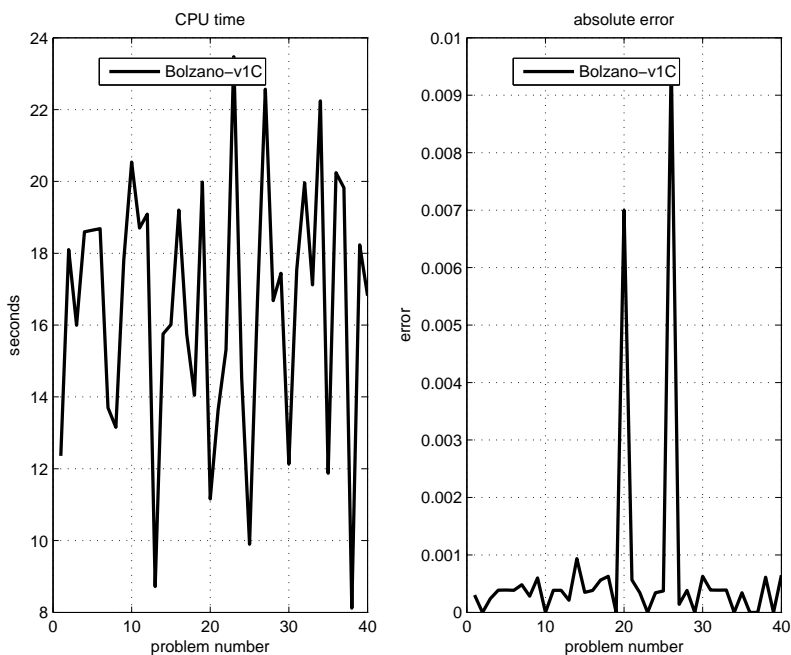


Figure 5
CPU time and absolute error of Algorithm 1 using α -dense curve based solver

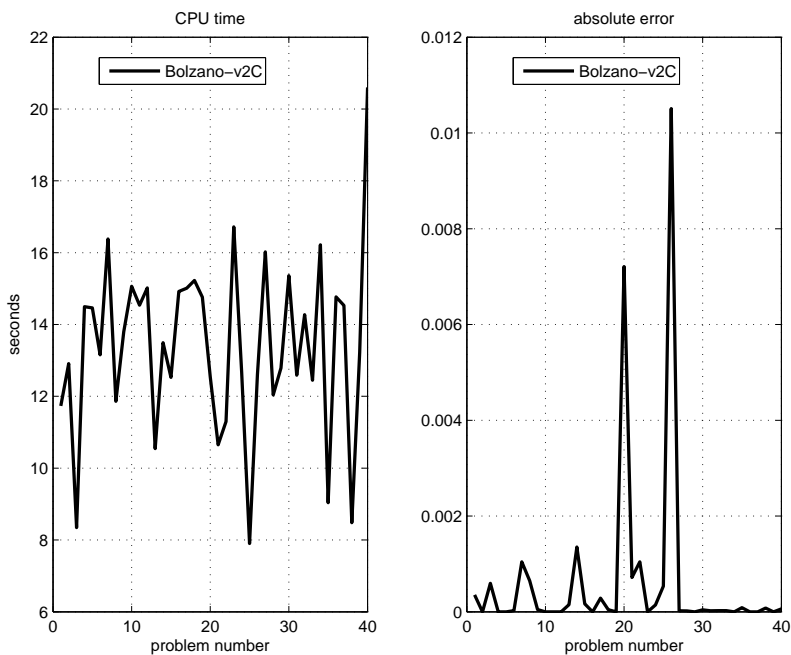


Figure 6

CPU time and absolute error of Algorithm 2 using α -dense curve based solver.

For these versions of Algorithms 1 and 2, the computational times are significant less, while the achieved precision is also better. For Algorithm 1, none of the absolute errors exceeds $10 \times tol = 1e-2$, while for Algorithm 2, there is only one case, test number 26, when the error exceed $1e-2$. It is, in fact, 0.010507.

A comparison of the four versions using the performance profile of Moré et al. [13], [26] clearly shows the ranking of the Algorithms (see Figure 7).

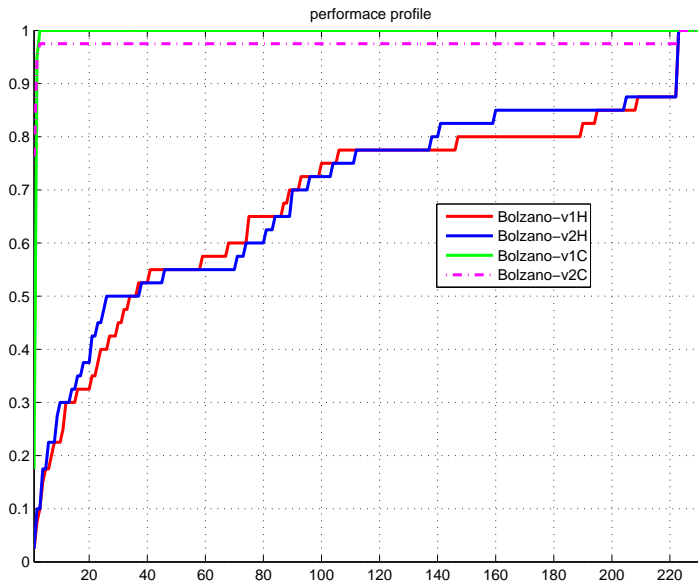


Figure 7
performance profile

4 Conclusions

In the paper we described two general algorithms to find a global minimum of a continuous function in an n -dimensional rectangle. The key point of our algorithms is to solve underdetermined nonlinear equations of the form $f(x) = c$ with such a method that gives unambiguously if the solution exists or not. For this purpose we used a method that exploits the Hilbert space filling function and Cherruault's α dense function. We tested the algorithms for 40 two dimensional well-known test problems. The experimental results clearly indicate that the solutions obtained by the α -dense function based algorithms are much more accurate and require much less execution time than the corresponding algorithms using the Hilbert space filling function. The obtained results show also the reliability of the algorithms in the numerical implementations too. Finally we have to mention that we still need to analyze the cases when $n > 2$.

5 Appendix

Here we enlist the test problems.

1. Adjiman function

$$f(x) = \cos(x_1) \sin(x_2) - \frac{x_1}{x_2^2 + 1} \quad (x \in [-1, 2] \times [-1, 1]).$$

2. Alpine 1 function

$$f(x) = \sum_{i=1}^n |x_i \sin(x_i) + 0.1x_i| \quad (x \in [-10, 10]^n).$$

3. Alpine 2 function

$$f(x) = \prod_{i=1}^n \sqrt{x_i} \sin(x_i), \quad (x \in [0, 10]^n).$$

4. Bohachevsky 1 function

$$f(x) = x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1) - 0.4 \cos(4\pi x_2) + 0.7 \quad (x \in [-1, 1]^2).$$

5. Bohachevsky 2 function

$$f(x) = x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1) \cos(4\pi x_2) + 0.3 \quad (x \in [-1, 1]^2).$$

6. Bohachevsky 3 function

$$f(x) = x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1 + 4\pi x_2) + 0.3 \quad (x \in [-1, 1]^2).$$

7. Booth function

$$f(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2 \quad (x \in [-10, 10]^2).$$

8. Branin function

$$f(x) = \left(x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6 \right)^2 + 10 \left(1 - \frac{1}{8\pi} \right) \cos(x_1) + 10,$$

where $x \in [-5, 10] \times [0, 15]$.

9. Brown almost linear function

$$f(x) = \sum_{i=1}^n f_i^2(x), \quad (x \in [-1, 2]^n),$$

$$f_i(x) = x_i + \sum_{j=1}^n x_j - (n+1), \quad 1 \leq i \leq n-1,$$

$$f_n(x) = \left(\prod_{j=1}^n x_j \right) - 1.$$

10. Bukin 12 function

$$f(x) = 1000(|x_1 + 5 - \rho \cos(\rho)| + |x_2 + 5 - \rho \sin(\rho)|) + \rho,$$

$$\rho = \sqrt{(x_1 + 5)^2 + (x_2 + 5)^2}, \quad x \in [-10, 0]^2.$$

11. Chained crescent function 1

$$f(x) = \max \left\{ \sum_{i=1}^{n-1} \left(x_i^2 + (x_{i+1} - 1)^2 + x_{i+1} - 1 \right), \right. \\ \left. \sum_{i=1}^{n-1} \left(-x_i^2 - (x_{i+1} - 1)^2 + x_{i+1} + 1 \right) \right\},$$

where $x \in [-1, 1]^n$.

12. Chained crescent function 2

$$f(x) = \sum_{i=1}^{n-1} \max \left\{ x_i^2 + (x_{i+1} - 1)^2 + x_{i+1} - 1, -x_i^2 - (x_{i+1} - 1)^2 + x_{i+1} + 1 \right\} \\ x \in [-1, 1]^n.$$

13. Chained LQ function

$$f(x) = \sum_{i=1}^{n-1} \max \left\{ -x_i - x_{i+1}, -x_i - x_{i+1} + (x_i^2 + x_{i+1}^2 - 1) \right\} \quad (x \in [-1, 1]^n).$$

14. Chained Mifflin function

$$f(x) = \sum_{i=1}^{n-1} \left(-x_i + 2(x_i^2 + x_{i+1}^2 - 1) + 1.75 |x_i^2 + x_{i+1}^2 - 1| \right) \quad (x \in [-1, 4]^n).$$

15. Chichinadze function

$$f(x) = x_1^2 - 12x_1 + 11 + 10 \cos\left(\frac{\pi}{2}x_1\right) + 8 \sin(\pi x_1) - \frac{1}{\sqrt{2\pi}} e^{-\frac{(x_2 - 0.5)^2}{2}} \\ x \in [0, 10] \times [0, 5].$$

16. Cosine mixture function

$$f(x) = -0.1 \sum_{i=1}^n \cos(5\pi x_i) + \sum_{i=1}^n x_i^2 \quad (x \in [-1, 1]^n, -0.1 < 0 < 5\pi).$$

17. Cross in tray function

$$f(x) = -0.0001 \left(\left| \sin(x_1) \sin(x_2) e^{\left| 100 - \frac{\sqrt{x_1^2 + x_2^2}}{\pi} \right|} \right| + 1 \right)^{0.1} \quad (x \in [-10, 10]^2).$$

18. Deb function

$$f(x) = -\frac{1}{n} \sum_{i=1}^n \sin^6(5\pi x_i) \quad (x \in [-1, 1]^n).$$

19. Egg crate function

$$f(x) = x_1^2 + x_2^2 + 25 (\sin^2(x_1) + \sin^2(x_2)) \quad (x \in [-5, 5]^2).$$

20. El-Attar-Vidyasagar-Dutta function

$$f(x) = |x_1^2 + x_2 - 10| + |x_1 + x_2^2 - 7| + |x_1^2 - x_2^3 - 1| \quad (x \in [-5, 5]^2).$$

21. Hosaki function

$$f(x) = \left(1 - 8x_1 + 7x_1^2 - \frac{7}{3}x_1^3 + \frac{1}{4}x_1^4\right) x_2^2 e^{-x_2} \quad (x \in [0, 5] \times [0, 6]).$$

22. Levy function

$$f(x) = \sin^2(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 (1 + 10 \sin^2(\pi y_i + 1)) + (y_n - 1)^2 (1 + 10 \sin^2(2\pi y_n)),$$

where

$$y_i = 1 + \frac{x_i - 1}{4} \quad (i = 1, \dots, n), \quad x \in [-10, 10]^n.$$

23. MAXHILB function

$$f(x) = \max_{1 \leq i \leq n} \left| \sum_{j=1}^n \frac{x_j}{i + j - 1} \right| \quad (x \in [-1, 1]^n).$$

24. McCormick function

$$f(x) = \sin(x_1 + x_2) + (x_1 - x_2)^2 - \frac{3}{2}x_1 + \frac{5}{2}x_2 + 1 \quad (x \in [-1.5, 4] \times [-3, 3])$$

25. Michalewicz function

$$f(x) = - \sum_{i=1}^n \sin(x_i) \sin^{2m} \left(\frac{i x_i^2}{\pi} \right) \quad (x \in [0, \pi]^n, m = 10).$$

26. Mishra 2 function

$$f(x) = \left(1 + n - \frac{1}{2} \sum_{i=1}^{n-1} (x_i + x_{i+1})\right)^{n - \frac{1}{2} \sum_{i=1}^{n-1} (x_i + x_{i+1})} \quad (x \in [0, 1]^n).$$

27. Multimod function

$$f(x) = \sum_{i=1}^n |x_i| \prod_{j=1}^n |x_j| \quad (x \in [-10, 10]^n).$$

28. Nesterov 2 function

$$f(x) = \frac{1}{4} (x_1 - 1)^2 + \sum_{i=1}^{n-1} |x_{i+1} - 2x_i^2 + 1| \quad (x \in [0, 2]^n).$$

29. Nesterov 3 function

$$f(x) = \frac{1}{4} |x_1 - 1| + \sum_{i=1}^{n-1} |x_{i+1} - 2|x_i| + 1| \quad (x \in [0, 2]^n).$$

30. Parsopoulos function

$$f(x) = \cos(x_1)^2 + \sin(x_2)^2 \quad (x \in [-5, 5]^2).$$

31. Pathological function

$$f(x) = \sum_{i=1}^{n-1} \left(0.5 + \frac{\sin\left(\sqrt{100x_i^2 + x_{i+1}^2}\right)^2 - 0.5}{1 + 0.001(x_i^2 - 2x_i x_{i+1} + x_{i+1}^2)^2} \right) \quad (x \in [-100, 100]^n).$$

32. Pintér's function

$$f(x) = \sum_{i=1}^n ix_i^2 + \sum_{i=1}^n i \sin^2(x_{i-1} \sin x_i - x_i + \sin x_{i+1}) \\ + \sum_{i=1}^n i \ln \left(1 + i(x_{i-1}^2 - 2x_i + 3x_{i+1} - \cos x_i + 1)^2 \right), \\ x_0 = x_n, x_{n+1} = x_1, x \in [-1, 1]^n.$$

33. Powell sum function

$$f(x) = \sum_{i=1}^n |x_i|^{i+1} \quad (x \in [-1, 1]^n).$$

34. Trigonometric function

$$f(x) = \sum_{i=1}^n f_i^2(x), \quad (x \in [-1, 1]^n), \\ f_i(x) = n - \sum_{j=1}^n \cos(x_j) + i(1 - \cos(x_i)) - \sin(x_i).$$

35. Ursem F1 function

$$f(x) = -\sin(2x_1 - 0.5\pi) - 3\cos(x_2) - 0.5x_1 \quad (x \in [-2.5, 3] \times [-2, 2]).$$

36. Ursem F3 function

$$f(x) = -\sin(2.2\pi x_1 + 0.5\pi) \frac{(3 - |x_1|)(2 - |x_2|)}{4} - \sin(0.5\pi x_2^2 + 0.5\pi) \frac{(2 - |x_1|)(2 - |x_2|)}{4},$$

where $x \in [-2, 2] \times [-1.5, 1.5]$.

37. Ursem F4 function

$$f(x) = -3 \sin(0.5\pi x_1 + 0.5\pi) \frac{2 - \sqrt{x_1^2 + x_2^2}}{4} \quad (x \in [-2, 2]^2).$$

38. Vincent function

$$f(x) = -\sum_{i=1}^n \sin(10 \log(x_i)) \quad (x \in [0.25, 10]^n).$$

39. W function

$$f(x) = 1 - \frac{1}{n} \sum_{i=1}^n \cos(kx_i) e^{-\frac{x_i^2}{2}} \quad (x \in [-\pi, \pi]^n),$$

where k is a parameter. $k = 10$ for $n = 2$.

40. Yang function 1

$$f(x) = \left(e^{-\sum_{i=1}^n \left(\frac{x_i}{\beta}\right)^{2m}} - 2e^{-\sum_{i=1}^n (x_i - \pi)^2} \right) \prod_{i=1}^n \cos^2(x_i),$$

$$m = 5, \beta = 15, x \in [-1, 4]^n$$

References

- [1] Abaffy J., Galántai A.: A globally convergent branch and bound algorithm for global minimization, in LINDI 2011 3rd IEEE International Symposium on Logistics and Industrial Informatics, August 25–27, 2011, Budapest, Hungary, IEEE, 2011, pp. 205-207, ISBN: 978-1-4577-1842-7, DOI: 10.1109/LINDI.2011.6031148
- [2] Abaffy J., Galántai A.: An always convergent algorithm for global minimization of univariate Lipschitz functions, Acta Polytechnica Hungarica, vol. 10, No. 7, 2013, 21–39
- [3] Bader, M.: Space-Filling Curves An Introduction with Applications in Scientific Computing, Springer, 2013
- [4] Bauman, K.E.: The Dilation Factor of the Peano-Hilbert curve, Mathematical Notes, 2006, 80, 5, 609-620

- [5] Butz, A.R.: Space Filling Curves and Mathematical Programming, *Information and Control*, 1968, 12, 314–330
- [6] Butz, A.R.: Convergence with Hilbert's Space Filling Curve, *Journal of Computer and System Sciences*, 1969, 3, 128–146
- [7] Butz, A.R.: Alternative algorithms for Hilbert's space-filling curve, *IEEE Transactions on Computers*, April, 1971, 424–426
- [8] Butz, A.R.: Solutions of Nonlinear Equations with Space Filling Curves, *Journal of Mathematical Analysis and Applications*, 1972, 37, 351–383
- [9] Cherruault, Y.: *Mathematical Modelling in Biomedicine*, D. Reidel Publishing Company, Dordrecht, Holland, 1986
- [10] Cherruault, Y., Mora, G.: *Optimisation Globale. Théorie des courbes α -denses*, Economica, Paris, 2005, ISBN 2-7178-5065-1
- [11] Csendes T.: Nonlinear parameter estimation by global optimization - efficiency and reliability, *Acta Cybernetica* 8, 1988, 361–370
- [12] Csendes T., Pál L., Sendín, J.-Ó. H., Banga, J.R.: The GLOBAL optimization method revisited, *Optimization Letters*, 2, 2008, 445–454
- [13] Dolan, E.D., Moré, J.J.: Benchmarking optimizations software with performance profiles, *Mathematical Programming, Series A* 91, 2002, 201–213
- [14] Galántai, A.: Always convergent methods for solving nonlinear equations, *Journal of Computational and Applied Mechanics*, Vol. 10., No. 2., 2015, pp. 183-208
- [15] Galántai, A.: Always convergent methods for solving nonlinear equations of several variables, *Numerical Algorithms*, DOI 10.1007/s11075-017-0392-z
- [16] Goertzel, B.: Global Optimizations with Space-Filling Curves, *Applied Mathematics Letters*, 12, 1999, 133-135
- [17] Guillez, A.: Alienor, fractal algorithm for multivariable problems, *Mathl Comput. Modelling*, 1990, 14, 245–247
- [18] Hansen, P., Jaumard, B., Lu, S.H.: On the number of iterations of Piyavskii's global optimization algorithm, *Mathematics of Operations Research*, 16, 1991, 334–350
- [19] Hansen, P., Jaumard, B., Lu, S.H.: On using estimates of Lipschitz constants in global optimization, *JOTA*, 75, 1, 1992, 195–200
- [20] Hansen, P., Jaumard, B., Lu, S.H.: Global optimization of univariate Lipschitz functions: I. Survey and properties, *Mathematical Programming*, 55, 1992, 251–272
- [21] Hansen, P., Jaumard, B., Lu, S.H.: Global optimization of univariate Lipschitz functions: II. New algorithms and computational comparison, *Mathematical Programming*, 55, 1992, 273–292

- [22] Levin, Y., Ben-Israel, A.: Directional Newton method in n variables, *Mathematics of Computation*, 71, 2001, 251–262
- [23] McCormick, S.: An iterative procedure for the solution of constrained nonlinear equations with application to optimization problems, *Numerische Mathematik*, 23, 1975, 371–385
- [24] Meyn, K.-H.: Solution of underdetermined nonlinear equations by stationary iteration methods, *Numerische Mathematik*, 42, 1983, 161–172
- [25] Mora, G.: The Peano curves as limit of α -dense curves, *Rev. R. Acad. Cien. Serie A. Mat.* 2005, 99, 1, 23–28
- [26] Moré, J.J, Wild, S.M.: Benchmarking derivative-free optimization algorithms, *SIAM J. Optimization*, 20,1, 2009, 172–191
- [27] Nemirovsky, S., Yudin, V.: *Problem Complexity and Method Efficiency in Optimization*, Wiley, 1983
- [28] Pintér, J.D.: *Global Optimization in Action*, Kluwer, 1996
- [29] Sagan, H.: *Space-filling Curves*, Springer, 1994
- [30] Shary, S.P.: A surprising approach in interval global optimization, *Reliable Computing*, 7, 2001, 497–505
- [31] Shary, S.P.: Graph subdivision methods in interval global optimization, in M. Ceberio, V. Kreinovich (eds.) *Constraint Programming and Decision Making, Studies in Computational Intelligence 539*, Springer, 2014, 153–170
- [32] Sergeyev, Y.D., Strongin, R.G., Lera, D.: *Introduction to Global Optimization Exploiting Space-Filling Curves*, Springer, 2013
- [33] Singh, A.N.: *The Theory and Construction of Non-Differentiable Functions*, Lucknow University Studies, No. I., Lucknow, India, 1935
- [34] Strongin, R.G.: On the convergence of an algorithm for finding a global extremum, *Engineering Cybernetics*, 1973, 11, 4, 549–555
- [35] Strongin, R.G., Sergeyev, Y.D.: *Global Optimization with Non-Convex Constraints*, Springer, 2000
- [36] Szabó Z.: Über gleichungslösende Iterationen ohne Divergenzpunkt I-III, *Publ. Math. Debrecen*, 20 (1973) 222-233, 21 (1974) 285–293, 27 (1980) 185-200
- [37] Szabó Z.: Ein Erweiterungsversuch des divergenzpunktfreien Verfahrens der Berührungsprabeln zur Lösung nichtlinearer Gleichungen in normierten Vektorverbänden, *Rostock. Math. Kolloq.*, 22, 1983, 89–107
- [38] Tompkins, C.: Projection methods in calculation, in: H. Antosiewicz (ed.): *Proc. Second Symposium on Linear Programming*, Washington, D.C., 1955, 425–448

- [39] Törn, A., Zilinskas, A.: Global Optimization, Lecture Notes in Computer Science 350, Springer, 1987
- [40] Wood, G.R.: The bisection method in higher dimensions, *Mathematical Programming*, 55, 1992, 319–337
- [41] Wood, G.: Bisection global optimization methods, in: C.A. Floudas, P.M. Pardalos (eds.): *Encyclopedia of Optimization*, 2nd ed., Springer, 2009, pp. 294–297
- [42] Zumbusch, G.: *Parallel Multilevel Methods: Adaptive Mesh Refinement and Loadbalancing*, B.G. Teubner, Stuttgart-Leipzig-Wiesbaden, 2003