INFORMATICA, 2016, Vol. 27, No. 2, 367–386

DOI: http://dx.doi.org/10.15388/Informatica.2016.90

On Solution of One Equation with d.c. Function

Alexander STREKALOVSKIY[1]*, Elena MUSATOVA[2]
[1] *Matrosov Institute for System Dynamics & Control Theory SB RAS*
Lermontov St., 134, Irkutsk, 664033, Russia
[2] *V.A. Trapeznikov Institute of Control Sciences of RAS*
Profsoyuznaya St., 65, Moscow, 117997, Russia
e-mail: strekal@icc.ru, nekolyap@mail.ru

Received: December 2015; accepted: May 2016

Abstract. In the paper we address the classical problem of solving one equation given by (d.c.) function represented by the difference of two convex functions. This problem is initiated by the optimization problems with constraints in the form of inequalities and/or equalities given by d.c. functions when one needs to descent from an unfeasible point to the boundary of a constraint improving, at the same time, the value of the objective function. We propose a new numerical procedure which allows to do this. Further, for the developed algorithm we provide the convergence results and numerical results of computational testing which look rather promising and competitive.

Key words: d.c. functions, nonlinear equation, numerical method, nonconvex optimization.

1. Introduction

In the second half of the 20-th century the computational mathematics became one of the most powerful and effective tools for the decision making procedures in various areas of the human civilization: from medicine (Martinkenas *et al.*, 2007) to economic and technical problems (Törn and Zilinskas, 2007).

Those achievements have been attained, in particular, due to the fantastic progress in computational sciences and developments in the field of computer technology.

On the other hand, the progress in the mathematical methods, specifically, in optimization, is not so obvious, for instance, it concerns nonconvex optimization and solution of the equation systems (Bakhvalov, 1977; Dennis and Schnabel, 1996; Horst and Tuy, 1993; Kelley, 1995; Ortega and Rheinboldt, 1970; Strekalovsky, 2003; Törn and Zilinskas, 2007; Zhigljavsky and Zilinskas, 2008).

However, observe that the latter case might be treated by the optimization approach, but nowadays nonconvex optimization problems are viewed as very difficult and often computationally intractable, because real-life nonconvex optimization problems may have a lot (often a huge number!) of local extrema and stationary (KKT-) points which are rather far from a global solution (Hiriart-Urruty, 1985; Nocedal and Wright, 2006; Izmailov and

*Corresponding author.

Solodov, 2014; Mathar and Zilinskas, 1994; Sergeyev and Kvasov, 2008; Strekalovsky, 2014; Horst and Tuy, 1993; Zhigljavsky and Zilinskas, 2008).

It is worth mentioning that in the case of the equation systems such a situation corresponds to the multitude of critical points (generated by Newton's methods) which are rather far from the root set. As a consequence, the classical optimization methods for nonconvex problem and numerous variants of Newton's schemes turn out to be, in general, inoperative and ineffective when it comes to finding a (global) solution or roots of the equation system, because they fail to escape a local pit in the case of arbitrary starting point (Bakhvalov, 1977; Dennis and Schnabel, 1996; Kelley, 1995; Ortega and Rheinboldt, 1970; Nocedal and Wright, 2006; Zhigljavsky and Zilinskas, 2008).

Therefore, all experts in the field agree that we need to search for new ways of constructing new numeric schemes designed for escaping local pits or improving of Newtonian critical points. On the other hand, the linear space $DC(\mathbb{R}^n)$ of d.c. functions, i.e. the functions represented by the difference of two convex functions, is large enough to test any approach to solving the problems mentioned above.

Recall that $C^2(\mathbb{R}^n) \subset DC(\mathbb{R}^n)$, and all power polynomials belong to $DC(\mathbb{R}^n)$; etc. Horst and Tuy (1993), Strekalovsky (2003), Hiriart-Urruty (1985).

Moreover, every continuous function on a compact can be approximated by a d.c. function at any accuracy. As a consequence, any system of equations with continuous functions can be replaced by an equivalent system of equations with a d.c. function at any desired accuracy. Finally, the convex cone of convex functions has been rather profoundly investigated in Hiriart-Urruty (1985).

Therefore, this paper addresses the simplest object in the equations theory – a single equation with a d.c. function. After the statement of the problem and its motivation in Section 2, we propose a numeric scheme for solving the equation. In Section 3, the convergence of the procedure is investigated. Finally, we present and analyse results of computational simulations.

2. Statement of the Problem and Motivation

Consider the equation

$$(\mathcal{E}): \qquad F(x) = 0, \quad x \in \mathbb{R}^n, \tag{2.1}$$

where $F(\cdot)$ is a (d.c.) function, which can be represented as a difference of two convex functions $g(\cdot)$ and $h(\cdot)$:

$$F(x) = g(x) - h(x),$$

where $g(\cdot)$ is continuously differentiable. Assume that there exist points u and v from $\mathbb{R}^n$ that satisfies the following relations

$$F(u) \triangleq g(u) - h(u) < 0 < F(v) \triangleq g(v) - h(v). \tag{2.2}$$

Remember, that the solution of the problem under consideration should be carried out within the general optimization problem with the d.c. inequality constraints for finding an admissible point, which is better than the current critical one provided by, say, a local search, in particular, in the problem (Strekalovsky, 2003; Horst and Tuy, 1993) as follows:

$$(\mathcal{P}_0): \qquad \left.\begin{array}{l} f(x) \downarrow \min, \quad x \in S, \\ F(x) \triangleq g(x) - h(x) \leqslant 0. \end{array}\right\} \tag{2.3}$$

Note, that in place of the point u we use the point that violated equation (2.1) and at which $F(u) < 0$, meanwhile $f(u) \leqslant f(z)$, where z is the current critical point. To find v it is sufficient to solve the relaxed convex problem without the inequality constraint:

$$(\mathcal{PR}): \qquad f(x) \downarrow \min, \quad x \in S. \tag{2.4}$$

In this case, one may have the following relations

$$F(v) > 0, \quad f(v) < f(u) \leqslant f(z). \tag{2.5}$$

Therefore, by virtue of (2.2), there exists a number $\lambda \in]0; 1[$ that satisfies the condition

$$x_\lambda = \lambda u + (1 - \lambda)v = v + \lambda(u - v) : F(x_\lambda) = 0, \tag{2.6}$$

or, which is the same,

$$g\big(x(\lambda)\big) = h\big(x(\lambda)\big). \tag{2.6$'$}$$

So, to find the solution to (2.6′), it is sufficient to perform a one-dimensional search along $\lambda \in]0; 1[$. However, the solution to (2.6′) might not be unique, and ideally we should choose the convex combination coefficient λ that corresponds to the smallest value of the goal function $f(x(\lambda))$. The explanation is that, in virtue, say, of convexity of $f(\cdot)$, the following inequalities hold:

$$f(x(\lambda)) \leqslant \lambda f(u) + (1 - \lambda) f(v) < f(z). \tag{2.7}$$

Taking into consideration (2.5), it is easy to see that the closer λ is to zero, the smaller the upper bound of the value $f(x(\lambda))$ is. Besides, if $f(\cdot)$ is convex, then the value of $f(x(\lambda))$ is also smaller.

Therefore, among the solutions to (2.6′), it is reasonable to search for the one that corresponds to the smallest λ.

3. Numerical Method

Further on, we propose a procedure which aims at the approximate computation of λ that satisfies (2.6′). To find μ_0, we do not use the equality

$$g(x(\mu_0)) = h\big(x(\mu_0)\big). \tag{3.1}$$

Instead, we employ the following relation:

$$g(v) + \mu_0\big\langle \nabla g(v), u - v\big\rangle = \mu_0 h(u) + (1 - \mu_0)h(v).$$

It means that we linearize the function $g(\cdot)$ at the point v, and in place of the function h we use the convex combination of its values at the points u and v. Then it immediately follows from the latter relation that

$$\mu_0 = \frac{F(v)}{h(u) - h(v) - \langle \nabla g(v), u - v\rangle}. \tag{3.2}$$

Hence, if we choose μ_0 as shown above, then we use an explicit form of $F(\cdot)$ as a difference of two convex functions. In this case, due to (2.2) and the convexity of $g(\cdot)$, the denominator in (3.2) turns out to be positive, since

$$\begin{aligned} \varphi(u, v) &\triangleq h(u) - h(v) - \big\langle \nabla g(v), u - v\big\rangle \\ &\geqslant h(u) - h(v) - g(u) + g(v) = F(v) - F(u) > 0. \end{aligned} \tag{3.3}$$

Thus, using (2.2), (3.2) and (3.3), we arrive at the chain of inequalities

$$0 < \mu_0 = \frac{F(v)}{\varphi(u, v)} \leqslant \frac{F(v)}{F(v) - F(u)} < 1. \tag{3.4}$$

Consequently, the number $\mu_0 \in\,]0, 1[$ can be a coefficient for the convex combination of the vectors u and v satisfying (2.2).

Further on, taking into consideration convexity of $g(\cdot)$ and $h(\cdot)$ and using (3.2), we obtain

$$\begin{aligned} F(x(\mu_0)) &= g\big(x(\mu_0)\big) - h\big(x(\mu_0)\big) = g\big(\mu_0 u + (1 - \mu_0)v\big) - h\big(\mu_0 u + (1 - \mu_0)v\big) \\ &\geqslant g\big(v + \mu_0(u - v)\big) - \mu_0 h(u) - (1 - \mu_0)h(v) \\ &\geqslant g(v) + \mu_0\big\langle \nabla g(v), u - v\big\rangle - \mu_0 h(u) - (1 - \mu_0)h(v) = 0, \end{aligned}$$

so that the following inequality is valid at the point $x(\mu_0)$:

$$F\big(x(\mu_0)\big) \geqslant 0. \tag{3.5}$$

Taking into account (3.2)–(3.5) and using the sequence $\{\mu_k\}$ of the convex combination coefficients, we can develop the procedure of convex combination (CoComba), which generates the sequence $\{x^s\}$, starting at $x^0 = v$ and all points of which belongs to the segment $[u; v]$.

Procedure "CoComba"

Step 0. Set $k := 0$, $x^k := v$.

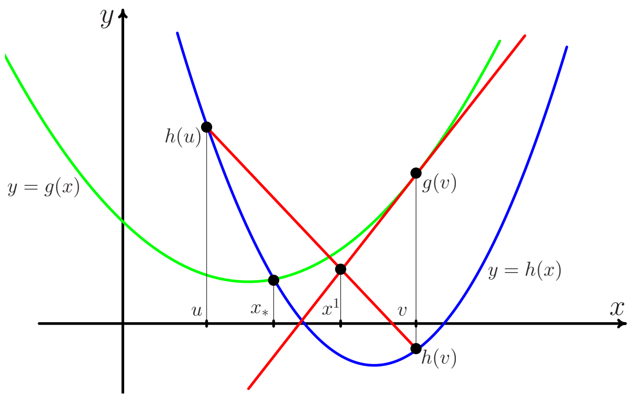


Fig. 1. The first iteration of procedure CoComba.

Step 1. Compute

$$\mu_k := \frac{F(x^k)}{h(u) - h(x^k) - \langle \nabla g(x^k), u - x^k \rangle}. \tag{3.6}$$

Step 2. Construct a convex combination

$$x := x^k + \mu_k \big(u - x^k\big). \tag{3.7}$$

Step 3. If $F(x) \leqslant \varepsilon$, then **Stop**, x is an ε-solution to the equation $F(x) = 0$.
Step 4. Set $k := k + 1$, $x^k := x$, move to **Step 1**.

Now let us give a geometric interpretation of the method proposed. More precisely Fig. 1 shows the first iteration of CoComba for the one-dimensional case ($x \in \mathbb{R}$). Besides, on the segment $[u, v]$ in place of $h(\cdot)$, i.e. we construct a segment that passes through the points $(u, h(u))$, $(v, h(v))$. After that, we linearize $g(\cdot)$, i.e. we construct the tangent to $y = g(x)$ at the point v. The point x^1 is the projection (into the space X, absciss) of the point of intersection of two lines, and all subsequent points x^k are located to the right of the root x_* of the equation.

Further, consider the relations of CoComba with the well-known methods (Bakhvalov, 1977; Ortega and Rheinboldt, 1970). In particular, let us investigate the cases, when equation (2.1) is defined by one convex function, and observe various changes in CoComba.

Case 1. Let $g(x) \equiv 0$. Then, on Step 1 the convex combination coefficient μ_k is found as

$$\mu_k = -\frac{h(x^k)}{h(u) - h(x^k)}. \tag{3.8}$$

By substituting μ_k into (3.7), we obtain

$$x^{k+1} = x^k - \frac{h(x^k)(u - x^k)}{h(u) - h(x^k)}, \quad k = 1, 2, \dots \tag{3.9}$$

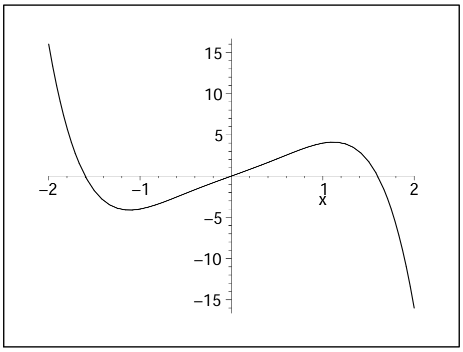


Fig. 2. The graph of function $F(x) = -x^5 + x^3 + 4x$.

which is the chord method (false position method) for solving the equation $h(x) = 0$ (Bakhvalov, 1977; Dennis and Schnabel, 1996; Ortega and Rheinboldt, 1970; Kelley, 1995).

Case 2. Let now in (2.1) the function $h(\cdot)$ be identically equal to zero: $h(x) \equiv 0$. Then, on Step 1 we obtain

$$\mu_k = -\frac{g(x^k)}{\langle \nabla g(x^k), u - x^k \rangle}, \tag{3.10}$$

which implies the iterative process

$$x^{k+1} = x^k - \frac{g(x^k)(u - x^k)}{\langle \nabla g(x^k), u - x^k \rangle}, \quad k = 1, 2, \dots \tag{3.11}$$

If $x \in \mathbb{R}^1$, this is nothing else but Newton's method for the equation $g(x) = 0$:

$$x^{k+1} = x^k - \frac{g(x^k)}{g'(x^k)}.$$

Note that the direct application of Newton's method (Bakhvalov, 1977; Izmailov and Solodov, 2014) or the chord method to the nonconvex function $F(x)$ does not always result in finding a root of the equation. The iterative process might diverge or converge to another root of the equation or even converge to a point which is not a root. The following example illustrates this.

Example 1. (See Nocedal and Wright, 2006.) Consider the equation $-x^5 + x^3 + 4x = 0$ (see Fig. 2).

The segment $[-1; 1]$ contains the root of the equation $x_* = 0$. If we choose $x_0 = 1$ or $x_0 = -1$ as a starting point for Newton's method, then the method loops endlessly generating either 1 or -1 for the next iteration.

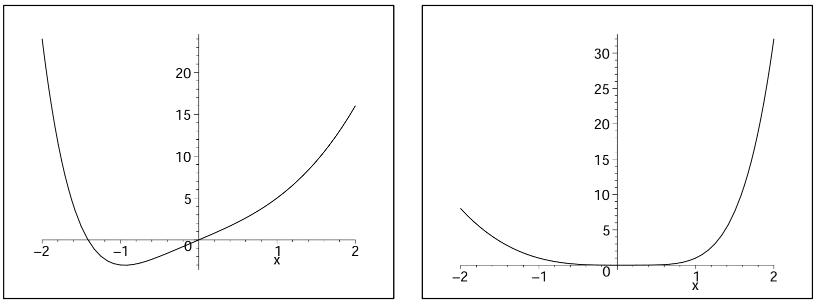


Fig. 3. The graphs of $g(\cdot)$ and $h(\cdot)$ from the d.c. representation of the function $F(x) = -x^5 + x^3 + 4x$.

If we apply formulae (3.9) to the segment $[u; v] = [-1.5; 1]$, the iterative process converges to the root $x \approx -1.6005$ that does not belong to the given segment.

Meanwhile, CoComba makes it possible to find the root $x_* = 0$ with an accuracy of 0.001 in 6 iterations in Case 1 and in 9 iterations in Case 2. Figure 3 shows graphs for the functions $g(x)$ and $h(x)$ from the d.c. representation of the function F defining the equation.

Here

$$g(x) = \begin{cases} -x^5 + 4x, & x < 0, \\ x^3 + 4x, & x \geqslant 0, \end{cases} \qquad h(x) = \begin{cases} -x^3, & x < 0, \\ x^5, & x \geqslant 0. \end{cases}$$

REMARK 1. Now let us have a closer look at the d.c. representation of the function defining the equation. As is well known (Hiriart-Urruty, 1985; Horst and Tuy, 1993), for any d.c. function $F(x)$ there exists an infinite number of the d.c. representations of d.c. function. The question that has to be answered is which d.c. representation is most suitable for the method in operation that uses the explicit representation into a difference of two convex functions.

Specifically, consider two d.c. representations for the same function $F(x)$. Let

$$F(x) = g_1(x) - h_1(x), \tag{3.12}$$

and, at the same time,

$$F(x) = g_2(x) - h_2(x), \tag{3.13}$$

where $g_2(x) = g_1(x) + r(x)$, $h_2(x) = h_1(x) + r(x)$, whereas $r(\cdot)$, $g_1(\cdot)$ and $h_1(\cdot)$ are convex functions. It means, we added the same convex function $r(x)$ to both convex terms $g_1(x)$ and $h_1(x)$ which are employed in the first d.c. representation.

In Case 1, the convex combination coefficient has the following form on Step 1:

$$\mu_{01} := \frac{F(x^0)}{h_1(u) - h_1(x^0) - \langle \nabla g_1(x^0), u - x^0 \rangle}. \tag{3.14}$$

On the other hand, in Case 2, on account of the convexity of $r(\cdot)$, we get

$$\begin{aligned}\mu_{02} &:= \frac{F(x^0)}{h_2(u) - h_2(x^0) - \langle \nabla g_2(x^0), u - x^0 \rangle} \\ &= \frac{F(x^0)}{h_1(u) - h_1(x^0) - \langle \nabla g_1(x^0), u - x^0 \rangle + (r(u) - r(x^0) - \langle \nabla r(x^0), u - x^0 \rangle)}.\end{aligned}$$

Further on, by employing the inequality of convexity for $r(\cdot)$

$$r(u) - r(x^0) - \langle \nabla r(x^0), u - x^0 \rangle \geqslant 0,$$

we arrive at the following relations

$$\mu_{02} \leqslant \frac{F(x^0)}{h_1(u) - h_1(x^0) - \langle \nabla g_1(x^0), u - x^0 \rangle} = \mu_{01}.$$

In particular, when $r(x) = \|x\|^2$, the coefficient μ_{02} reads as

$$\mu_{02} = \frac{F(x^0)}{\varphi(u, x^0) + (\|u\|^2 - \|x^0\|^2 - 2\langle x^0, u - x^0 \rangle)} = \frac{F(x^0)}{\varphi(u, x^0) + \|u - x^0\|^2}.$$

Thus, since $\|x^1 - x^0\| = \mu_0 \|u - x^0\|$, the strongly convex function $\|x\|^2$ added to $g_1(\cdot)$ and $h_1(\cdot)$ decreases the convex combination coefficient and therefore the method's step size. This fact should be taken into consideration when choosing the d.c. representation of the function defining the equation. For example, if possible, we can take the minimal d.c. representation (Hiriart-Urruty, 1985).

4. Convergence Proof for the CoComba Procedure

Let us now investigate the properties of the sequence $\{x^k\}$ generated by the CoComba procedure when $\varepsilon = 0$, i.e. when the iterative process is infinite.

Since $x^0 := v$, $F(x^0) > 0$, then we obtain from (3.5)

$$F(x^k) \geqslant 0, \quad k = 0, 1, 2, \ldots \tag{4.1}$$

On the other hand, by construction $x^0 := v$, and one can see that

$$\begin{aligned}x^1 - x^0 &= \mu_0(u - x^0), \quad 0 < \mu_0 < 1, \\ x^2 - x^1 &= \mu_1(u - x^1) = \mu_1[u - x^0 - \mu_0(u - x^0)] = \mu_1(1 - \mu_0)(u - x^0), \\ x^3 - x^2 &= \mu_2(u - x^2) = \mu_2[u - x^1 - \mu_1(u - x^1)] = \mu_2(1 - \mu_1)(u - x^1) \\ &= \mu_2(1 - \mu_1)[u - x^0\mu_0(u - x^0)] = \mu_2(1 - \mu_1)(1 - \mu_0)(u - x^0) \\ &= \mu_2(1 - \mu_1)(1 - \mu_0)(u - v).\end{aligned} \tag{4.2}$$

Therefore, it follows immediately that

$$x^{k+1} - x^k = \mu_k(1-\mu_{k-1})(1-\mu_{k-2})\cdots(1-\mu_0)(u-v). \tag{4.3}$$

Besides, similarly, we get

$$x^{k+1} - u = x^k + \mu_k\big(u - x^k\big) - u = (1-\mu_k)(1-\mu_{k-1})\cdots(1-\mu_0)(v-u). \tag{4.4}$$

Hence, the behaviour of the sequence $\{x^k\}$ depends on the behaviour of $\{\mu_k\}$ and $\{1-\mu_k\}$. For this reason, first we investigate properties of the sequence $\{\mu_k\}$.

Theorem 1. *The numerical sequence* $\{\mu_k\}$ *constructed by the rule* (3.6) *converges to zero*: $\lim_{k\to\infty} \mu_k = 0$.

Proof. Suppose the contrary and let $\{\mu_k\}$ satisfy the condition

$$\mu_k \geqslant \gamma > 0 \quad \forall k = 0, 1, 2, \ldots$$

Then, due to the inequalities

$$0 < \gamma \leqslant \mu_k < 1,$$

we have

$$1 - \mu_k \leqslant 1 - \gamma = q < 1 \quad \forall k = 0, 1, 2, \ldots$$

Therefore, it follows from (4.4)

$$\big\|x^{k+1} - u\big\| = (1-\mu_k)(1-\mu_{k-1})\cdots(1-\mu_0)\|v-u\| \leqslant q^{k+1}\|v-u\|.$$

Consequently, we derive

$$\lim_{s\to\infty} \big\|x^{k+1} - u\big\| \leqslant \lim_{k\to\infty} q^{k+1}\|v-u\| = 0.$$

This means that $x^k \to u$, which is impossible in virtue of continuity of the function $F(\cdot)$, because due to (4.1) we have

$$F(x^k) \geqslant 0 > F(u), \quad k = 0, 1, 2, \ldots \tag{4.5}$$

□

Corollary 1. *The sequence* $\{x^k\}$ *generated by CoComba converges*:

$$\lim_{k\to\infty} x^k = x_*, \qquad F(x_*) \geqslant 0.$$

Proof. It follows from (4.3) that

$$\|x^{k+1} - x^k\| = \mu_k(1-\mu_{k-1})\cdots(1-\mu_0)\|u-v\| \leqslant \mu_k\|u-v\|,$$

since

$$(1-\mu_i) \leqslant 1, \quad i = 0, 1, 2, \ldots, k-1.$$

Hence, taking into consideration that $\mu_k \downarrow 0$, we arrive at the equality

$$\lim_{k\to\infty} \|x^{k+1} - x^k\| = 0.$$

Therefore, there exists a point $x_* \in \mathbb{R}^n : x_* = \lim_{k\to\infty} x^k$.

So, the sequence $\{x^k\}$ converges to some limit x_*. Due to the continuity of $F(\cdot)$, we obtain that

$$\lim_{k\to\infty} F(x^k) = F(x_*) \geqslant 0.$$

□

Let now introduce a set of numbers λ, which are the roots of equation (2.6′):

$$\Lambda_* = \{\lambda \in [0,1] \mid F(\lambda u + (1-\lambda)v) = 0\},$$

and a set of corresponding vectors $x(\lambda)$:

$$X_* = \{x \in \mathbb{R}^n \mid \exists \lambda \in \Lambda_* : x = \lambda u + (1-\lambda)v,\ F(x) = 0\}.$$

Further on, let us demonstrate that the sequence $\{x^k\}$ converges to the solution of equation (2.1), meanwhile the algorithm finds the closest to v root of the equation from the set X_*.

Theorem 2. *The limit x_* of the sequence $\{x^k\}$ generated by the CoComba procedure*

(i) *is a root of the equation $F(x) = 0$;*
(ii) *satisfies the relation*

$$\|x_* - v\| = \min_x \{\|x - v\| \mid x \in X_*\}. \tag{4.6}$$

Proof. (i) Since $F(x^k) \geqslant 0$ $\forall k = 0, 1, \ldots,$ then, due to continuity of $F(\cdot)$, $F(x_*) \geqslant 0$. Suppose that the limit of the sequence $\{x^k\}$ is not the root of the equation, i.e. there exists $\eta > 0$ such that

$$\lim_{s\to\infty} F(x^k) = F(x_*) = \eta > 0.$$

Then, by the construction of the convex combination coefficient μ_k on Step 1 of the algorithm from (3.6), we obtain

$$h(u) - h(x^k) - \langle \nabla g(x^k), u - x^k \rangle \triangleq \varphi(u, x^k) = \frac{F(x^k)}{\mu_k} \geqslant \frac{\eta}{\mu_k},$$

whence, due to Theorem 1, it follows that

$$\lim_{k\to\infty} \left[h(u) - h(x^k) - \langle \nabla g(x^k), u - x^k \rangle\right] \triangleq \lim_{k\to\infty} \varphi(u, x^k) = +\infty. \tag{4.7}$$

On the other hand, since $x^k \to x_*$, then, due to continuity of $h(\cdot)$, the mapping $\nabla g(\cdot)$ and the scalar product, we have

$$\begin{aligned}\lim_{k\to\infty} \varphi(u, x^k) &= \lim_{k\to\infty} \left[h(u) - h(x^k) - \langle \nabla g(x^k), u - x^k \rangle\right] \\ &= h(u) - h(x_*) - \langle \nabla g(x_*), u - x_* \rangle \in \mathbb{R},\end{aligned}$$

which does not coincide with (4.7). Therefore, the assumption that $F(x_*) > 0$ is false, and thereby the first statement of the theorem is proved.

(ii) Let us now show the validity of (4.6). Suppose $\bar{x} \in X_*$ is a solution to the equation for which

$$\|\bar{x} - v\| = \min_x \left\{ \|x - v\| \,\middle|\, x \in X_* \right\}.$$

Further, let us prove that x_* coincides with $\bar{x}$. To this end, along with the sequence $\{\mu_k\}$, we consider the sequence $\{\gamma_k\}$ such that

$$\bar{x} = \gamma_k u + (1 - \gamma_k)x^k, \quad k = 0, 1, 2, \ldots$$

Let us verify that at each step of the algorithm $\mu_k \leqslant \gamma_k$ and that the point x^k happens to be closer to v than $\bar{x}$, i.e. the following inequality holds:

$$\|x^k - v\| \leqslant \|\bar{x} - v\| \quad \forall k = 0, 1, 2, \ldots \tag{4.8}$$

First let us demonstrate that $\mu_0 \leqslant \gamma_0$ as $k = 0$.

Due to convexity of the functions $g(\cdot)$ and $h(\cdot)$, the following chain of inequalities holds:

$$\begin{aligned}&g(v) + \gamma_0 \langle \nabla g(v), u - v \rangle \leqslant g(\gamma_0 u + (1 - \gamma_0)v) \\ &\quad = h(\gamma_0 u + (1 - \gamma_0)v) \leqslant \gamma_0 h(u) + (1 - \gamma_0)h(v).\end{aligned} \tag{4.9}$$

Now, consider an affine function of one variable

$$\Psi_0(\gamma) = g(v) + \gamma \langle \nabla g(v), u - v \rangle - \gamma h(u) - (1 - \gamma)h(v),$$

which represents the difference between the right-hand and the left-hand sides of (4.9) where γ_0 is replaced by γ. Then, μ_0 is a unique root of the equation $\Psi_0(\lambda) = 0$. Furthermore, in virtue of (4.9), $\Psi_0(\gamma_0) \leqslant 0$.

On the other hand,

$$\Psi_0(0) = g(v) - h(v) > 0,$$

whence it follows that $\mu_0 \leqslant \gamma_0$. Therefore, we obtain

$$\left\| x^1 - v \right\| = \mu_0 \|u - v\| \leqslant \gamma_0 \|u - v\| = \|\bar{x} - v\|.$$

Consequently, the root $\bar{x}$ can be represented as a convex combination of the points u and x_1, so that $\gamma_1 \in [0; 1[$.

Further, let the following condition hold for $k = m$

$$\mu_{m-1} \leqslant \gamma_{m-1}, \quad \left\| x^m - v \right\| \leqslant \|\bar{x} - v\|, \quad \gamma_m \in [0; 1[. \tag{4.10}$$

Let us move to the $(m+1)$-th iteration. Similarly, for the function

$$\Psi_m(\gamma) = g\big(x^m\big) + \gamma \big\langle \nabla g\big(x^m\big), u - x^m \big\rangle - \gamma h(u) - (1-\gamma) h\big(x^m\big),$$

we get $\Psi_m(\mu_m) = 0$, $\Psi_m(0) = g(x^m) - h(x^m) \geqslant 0$, $\Psi_m(\gamma_m) \leqslant 0$, whence it follows that

$$\mu_m \leqslant \gamma_m. \tag{4.11}$$

Now let us prove that $\|x^{m+1} - v\| \leqslant \|\bar{x} - v\|$. Indeed, from (4.4) we obtain

$$\begin{aligned} \left\| x^{m+1} - v \right\| &= \left\| \big(x^{m+1} - u\big) + (u - v) \right\| \\ &= \left\| (1-\mu_m)(1-\mu_{m-1}) \cdots (1-\mu_0)(v - u) + (u - v) \right\| \\ &= \big(1 - (1-\mu_m)(1-\mu_{m-1}) \cdots (1-\mu_0)\big) \|u - v\|. \end{aligned} \tag{4.12}$$

On the other hand, it is easy to see that

$$\begin{aligned} \|\bar{x} - v\| &= \left\| (\gamma_m u + (1-\gamma_m) v - u) + (u - v) \right\| = \left\| (1-\gamma_m)(x_m - u) + (u - v) \right\| \\ &= \big(1 - (1-\gamma_m)(1-\mu_{m-1})(1-\mu_{m-2}) \cdots (1-\mu_0)\big) \|u - v\|. \end{aligned} \tag{4.13}$$

Furthermore, taking into consideration (4.11), we derive from (4.12) and (4.13) that $\|x^{m+1} - v\| \leqslant \|\bar{x} - v\|$.

Hence, we conclude that (4.8) is valid.

Finally, passing to the limit in (4.8) as $k \to \infty$, we obtain

$$\|x_* - v\| \leqslant \|\bar{x} - v\| = \min_x \big\{ \|x - v\| \bigm| x \in X_* \big\},$$

whence the assertion (4.6) of Theorem 2 follows. □

As has been mention in the introduction, in the one-dimensional case, the problem of finding a special solution, satisfying (4.6), can be motivated by a problem of finding the smallest root of the equation. Consider an example of such a problem arising in practice (Sergeyev and Kvasov, 2008).

Suppose it is required to determine how long a device can work properly within the time interval $[t_0, t_1]$. Herewith, the function $f(t)$ describes the proper work of the device, $f(t_0) > 0$. Besides, the device works properly at time t if $f(t) > 0$. We are asked to find such a moment of time t^* that

$$f(t^*) = 0, \quad f(t) > 0, \quad t \in \left[t_0, t^*\right[, \quad t^* \in]t_0, t_1].$$

It is worth noting that in Sergeyev and Kvasov (2008), Molinaro and Sergeyev (2001), the solution of this problem is based on construction of auxiliary functions to approximate f with various techniques for estimating the Lipschitz constant.

In addition, in Khamisov (2015) the method for solving a d.c. equation involves construction of concave support functions, which allows finding the closest to v root. At each iteration of this method, we linearize the function $g(\cdot)$ and search for the root x^{k+1} of the equation

$$\varphi\left(x, x^k\right) \triangleq g\left(x^k\right) + p^k\left(x - x^k\right) - h(x) = 0, \tag{4.14}$$

where $p^k \in \partial g(x^k)$, x^k is a point obtained at the previous iteration.

It is clear, that if $h(\cdot) \equiv 0$, then this method as well as CoComba coincides with Newton's method. Consider the relationship between the two methods in the general case.

Notice that if we use the method from Khamisov (2015) and, instead of finding the exact solution of (4.14), we carry out just a single iteration of the chord method, we obtain:

$$\mu\varphi\left(u, x^k\right) + (1 - \mu)\varphi\left(x^k, x^k\right) = 0$$

or

$$g\left(x^k\right) + \mu p^k\left(u - x^k\right) = \mu h(u) + (1 - \mu)h\left(x^k\right),$$

whence, as $p^k = \nabla g(x^k)$, we find the coefficient μ to compute the next approximation in CoComba.

Therefore, instead of solving (4.14) by some numerical method, CoComba executes only one step of the chord method, which, nevertheless, is enough for convergence to the root of the equation. In this case, we do not need to solve (4.14) at a high accuracy. Otherwise, we can conclude that in CoComba the computational complexity of each iteration is decreased in comparison to the method from Khamisov (2015).

5. Computational Simulations

The computational experiment aimed at numerical testing of the CoComba procedure for solving equations with d.c. functions and comparing it with some other methods that guarantee finding of a solution within a given interval. To carry out the comparison, we chose the modified chord method (MCM) and the bisection method.

The modified chord method allows us to stay within the given interval all the time by redefining the boundaries of the interval that contains the root. Below we give the description of the MCM "step-by-step".

Modified Chord Method

Step 0. Set $s := 0$, $v^s := v$, $u^s := u$.
Step 1. Compute

$$\mu_s := \frac{F(v^s)}{F(v^s) - F(u^s)}. \tag{5.1}$$

Step 2. Make a convex combination of the points u^s, v^s:

$$x^s := \mu_s u^s + (1 - \mu_s)v^s = v^s + \mu_s\big(u^s - v^s\big). \tag{5.2}$$

Step 3. If $|F(x^s)| \leqslant \varepsilon$, then x^s is an ε- approximate solution to $F(x) = 0$.
Step 4. If $F(x^s) < 0$, then set

$$u^{s+1} := x^s, \qquad v^{s+1} := v^s, \tag{5.3}$$

$s := s + 1$, move to Step 1.
Step 5. If $F(x^s) > 0$, then set

$$u^{s+1} := u^s, \qquad v^{s+1} := x^s, \tag{5.4}$$

$s := s + 1$, move to Step 1.

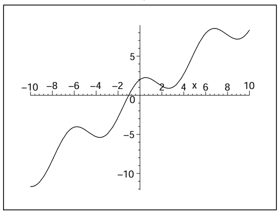
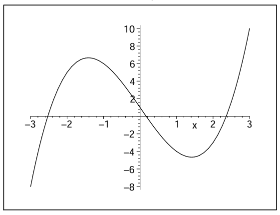
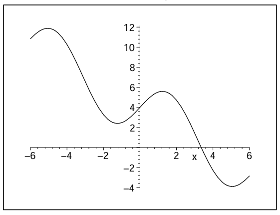
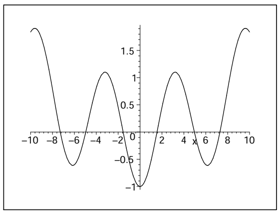
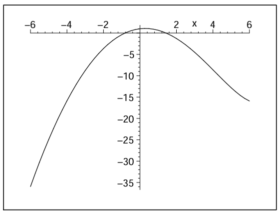
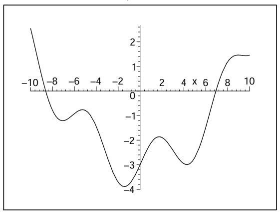
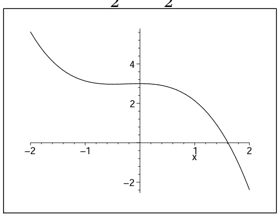
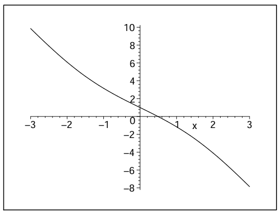
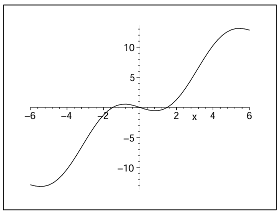
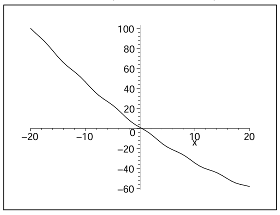
Here, as well as in the CoComba procedure, at each iteration we construct a convex combination of the points at which the function F has the opposite signs. However, we cannot predict (like it happened to be possible in the case of CoComba) which sign F has at the next iteration. Therefore, in contrast to CoComba, where the sequence $\{x^s\}$ approaches the solution from one side, here we construct two sequences $\{u^s\}$ and $\{v^s\}$.

At first, we tested the methods on the d.c. equations of one variable. The data on the test equations is given in Table 1.

Further, Table 2 comprises the results of the second stage of testing. Here, N is the equation number followed by the segment that contains the roots of the equations and the number of iterations required by CoComba, the modified chord method and the bisection method. Besides, to carry out the comparison, we used MatLab r2009a. Note that the bold font indicates the smallest number of iterations required to find a root by a respective method. Here the chosen accuracy ε was 10^{-7}.

It can be readily seen from Table 2 that CoComba is most effective with respect to the number of iterations in all cases.

Table 1
Test equations.

No.	Equation	No.	Equation
1	$2\cos x + x = 0$ 	6	$x^3 - 6x + 1 = 0$ 
2	$3\sin x - x + 4 = 0$ 	7	$0.01x^2 - \cos x = 0$ 
3	$e^{\frac{x}{2}} - x^2 = 0$ 	8	$0.05x^2 + \sin x = 0$ 
4	$3\cos\frac{x}{2} - \frac{x^3}{2} = 0$ 	9	$\sin x - 3x + 1 = 0$ 
5	$2x - 3\sin x = 0$ 	10	$\sin x - 4x + 0.05x^2 + 1 = 0$ 

However, it should be taken into account that the computational cost of a single iteration of the CoComba procedure is higher because it needs to compute the derivative of g. Nevertheless, all methods demonstrated comparable effectiveness with respect to the solution time.

Table 2
Solution of one-dimensional equations (number of iterations).

N	Interval	CoComba	MCM	Bisection M.
1	[−5; 0]	**4**	8	26
2	[0; 5]	**5**	**5**	27
3	[1; 2]	**7**	10	16
	[−5; 0]	**8**	43	19
4	[1; 2]	**4**	11	23
5	[1; 2]	**4**	14	23
	[0.3; 1.8]	**3**	14	24
6	[2; 3]	**5**	16	25
	[−1; 1]	**3**	4	25
7	[0; 2]	**3**	4	23
	[3; 6]	**4**	5	24
	[6; 10]	**4**	6	24
8	[6; 8]	**6**	7	27
9	[0; 1]	**3**	6	23
10	[0; 2]	**5**	9	21

Table 3
Increase of the interval containing the root (number of iterations).

N	Interval	CoComba	MCM	Bisection M.
1	[−10; 10]	33	**9**	28
2	[10; −10]	30	**6**	29
3	[0; 6]	**12**	25	19
	[−10; 0]	**9**	86	20
4	[−10; 10]	170	222	**25**
5	[1; 10]	**7**	15	26
6	[1; 10]	**8**	221	29
7	[−6; 10]	39	**6**	25
8	[−5; 15]	32	**11**	26
9	[−10; 10]	27	**15**	26
10	[−10; 10]	23	**17**	27

If we increase the interval containing the root (see Table 3), then the situation becomes quite different. As we could have foreseen, in this case the bisection method is most reliable, meanwhile the chord method as well as CoComba, being effective in certain cases, requires a large number of iterations to perform in some other situations (see, for example, Problem 4).

The similar phenomenon can be observed when solving multi-dimensional problems. The bisection method happened to be the fastest, however, recall that it does not additionally find the closest to v root, which is important when we search for the admissible point in the problem $(\mathcal{P}_0)$–(2.3) with the d.c. constraint. However, we can give examples when CoComba is most effective even for solving problems of large dimensions.

Tables 4 shows to results of solving the equation $\sin(x_1 + x_2 + \cdots + x_n) = 0$. In Table 4, we use the following denotations: n is a number of variables, u_i, v_i are components of the vectors u and v, respectively, It is a number of iterations, T_{1000} is time required to solve 1000 identical problems (solving one problem takes too little time).

Table 4
Solution of the equation $f(x) \triangleq \sin(x_1 + x_2 + \cdots + x_n) = 0$.

n	u_i	v_i	ε	CoComba		MCM		Bisection M.	
				It	T_{1000}	It	T_{1000}	It	T_{1000}
10	$\frac{\pi}{2n}$	1	0.001	3	0.09	**4**	**0.06**	12	0.14
			0.00001	5	0.16	**5**	**0.07**	19	0.19
100	$\frac{\pi}{2n}$	1.1	0.001	**2**	**0.09**	8	0.11	14	0.18
			0.00001	4	0.16	9	0.13	23	0.21
500	$\frac{\pi}{2n}$	1	0.001	**2**	**0.13**	11	0.24	18	0.39
			0.00001	**3**	**0.18**	12	0.27	24	0.51
1000	1	$-\frac{\pi}{2n}$	0.001	**3**	**0.23**	13	0.43	17	0.52
			0.00001	**4**	**0.31**	14	0.45	25	0.78

Here, independently of the growth in dimension, CoComba steadily passes 4 iterations outperforming other methods with respect to the number of iterations and running time.

In conclusion, consider an example from Khamisov (2015), which was used to compare CoComba with the method of concave support functions.

Find the root of the equation $f(x) \triangleq -\sin(x) - \sin(3x+1) + 1.5 = 0$. Assume [5; 8.5] as an interval containing the root.

The method of concave support functions required 13 iterations to find the root with accuracy of 10^{-7}. The CoComba took 48 iterations. However, the computational cost of these methods differs: in the first case, in addition to computing values of $h(\cdot)$ and $\nabla g(\cdot)$ or $g'(x) \in \partial g(x)$ at each iteration, one needs to somehow solve the equation (4.14) (for example, by the bisection method). Here CoComba has the obvious advantages. Since the dimension of example from Khamisov (2015) is equal to 1, counting time is negligibly small in both cases.

So, the test examples demonstrated effectiveness of CoComba. Since CoComba takes acceptable amount of time and also takes into consideration specific properties of the optimization problem, it is reasonable to use it in solving optimization problems with the d.c. constraints to find admissible points.

6. Conclusions

In the paper, we proposed a new numerical method (the CoComba procedure) for solving one equation with d.c. function of multi-dimensional variable and gave the motivation for this problem with some applications.

In addition, we investigated the convergence of the advanced algorithm and the properties of sequence generated by the method.

Furthermore, the computational testing has been carried out on the test examples from the literature. Finally, the results of computational experiments have been analysed and compared with the results of other methods such as the modified chord method and the bisection method.

To sum up, the developed new method shows itself comparable with the effectiveness of other methods, and the results of computational testing of the CoComba procedure look rather promising, in particular, with respect to the solution time and number of iterations.

References

Bakhvalov, N.S. (1977). *Numerical Methods: Analysis, Algebra, Ordinary Differential Equations*. MIR Publishers, Moscow.

Dennis, J.E., Schnabel, R.B. (1996). *Numerical Methods for Unconstrained Optimization and Nonlinear Equation*. SIAM, Philadelphia.

Hiriart-Urruty, J.-B. (1985). Generalized differentiability, duality and optimization for problem dealing with differences of convex functions. *Lecture Notes in Economics and Mathematical Systems*, Vol. 256, Springer-Verlag, Berlin, pp. 37–69.

Horst, R., Tuy, H. (1993). *Global Optimization. Deterministic Approaches*. Springer-Verlag, Berlin.

Izmailov, A.F., Solodov, M.V. (2014). *Newton-Type Methods for Optimization and Variational Problems*. Springer, New York.

Kelley, C.T. (1995). *Iterative Methods for Linear and Nonlinear Equations*. SIAM, Philadelphia.

Khamisov, O.V. (2015). Finding roots of nonlinear equations using the method of concave support functions. *Mathematical Notes*, 98(3), 484–491.

Martinkenas, A., Kaminskas, V., Varoneckas, G. (2007). Forecast model of impact of meteorological factors on coronary artery disease patients. *Informatica*, 18(3), 407–418.

Mathar, R., Zilinskas, A. (1994). A class of test functions for global optimization. *Journal of Global Optimization*, 5(2), 195–199.

Molinaro, A., Sergeyev, Y.D. (2001). Finding the minimal root of an equation with the multiextremal and non-differentiable left-hand part. *Numer. Algorithms*, 28(1–4), 255–272.

Nocedal, J., Wright, S.J. (2006). *Numerical Optimization*. Springer-Verlag, New York, Heidelberg.

Ortega, J.M., Rheinboldt, W.C. (1970). *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, New York, London.

Sergeyev Y.D., Kvasov, D.E. (2008). *Diagonal Global Optimization Methods*. Fizmatlit, Moscow (in Russian).

Strekalovsky, A.S. (2003). *Elements of Nonconvex Optimization*. Nauka, Novosibirsk (in Russian).

Strekalovsky, A.S. (2014). On solving optimization problems with hidden nonconvex structures. In: Rassias, T.M., Floudas, C.A., Butenko, S. (Eds.), *Optimization in Science and Engineering*. Springer, New York, pp. 465–502.

Törn, A., Zilinskas, J. (Eds.). (2007). Models and Algorithms for Global Optomization. *Essays Dedicated to Antanas Zilinskas on the Occasion of His 60th Birthday*. Springer, New York, 362 pp.

Zhigljavsky, A., Zilinskas, J. (2008). *Stochastic Global Optomization*. Springer, New York, 262 pp.

A. Strekalovskiy was awarded a degree of Doctor of Sciences in physics and mathematics at the Department of Computational Mathematics and Cybernetics at Lomonosov State University of Moscow in 1993. He is a doctor-professor from 1995. A. Strekalovskiy is a Honored scientist of Russian Federation. His main research interests include nonconvex optimization problems, bilevel programming, polymatrix games, optimal control and etc. At present, he is the head of laboratory of Nonconvex Optimization of Matrosov Institute for System Dynamics and Control Theory of the Siberian Branch of Russian Academy of Sciences.

E. Musatova was awarded her PhD degree in physics and mathematics at the Institute for System Dynamics and Control Theory of the Siberian Branch of Russian Academy of Sciences in 2011. At present, she is a senior researcher of V.A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences.

Apie vienos lygties su D.C. funkcija sprendimą

Alexander STREKALOVSKIY, Elena MUSATOVA

Šiame straipsnyje nagrinėjame klasikinį uždavinį su viena lygtimi, aprašyta D.C. funkcija, pateikta per dviejų iškilų funkcijų skirtumą. Šis uždavinys yra inicijuotas optimizavimo uždavinių su nelygybiniais ir arba lygybiniais apribojimais, aprašytais D.C. funkcijomis, kai reikia grįžti iš neleistino taško prie apribojimo krašto ir tuo pačiu pagerinti tikslo funkcijos reikšmę. Siūlome tai leidžiančią naują skaitinę procedūrą. Sukurtam algoritmui pateikiame konvergavimo rezultatus ir skaitinio testavimo rezultatus, kurie yra gana perspektyvūs ir konkurencingi.