Research Article Open Access

Yury Evtushenko and Mikhail Posypkin*

# Using necessary optimality conditions for acceleration of the nonuniform covering optimization method*

DOI 10.1515/eng-2016-0027
Received May 05, 2016; accepted Jun 22, 2016

**Abstract:** Paper deals with the non-uniform covering method that is aimed at deterministic global optimization. This method finds a feasible solution to the optimization problem numerically and proves that the obtained solution differs from the optimal by no more than a given accuracy. Numerical proof consists of constructing a set of covering sets – the coverage. The number of elements in the coverage can be very large and even exceed the total amount of available computer resources. Basic method of coverage construction is the comparison of upper and lower bounds on the value of the objective function. In this work we propose to use necessary optimality conditions of first and second order for reducing the search for box-constrained problems. We provide the algorithm description and prove its correctness. The efficiency of the proposed approach is studied on test problems.

**Keywords:** deterministic optimization; non-uniform covering method; optimality conditions; branch-and-bound method; search acceleration

## 1 Introduction

Today there are a great variety of methods for solving global optimization problems [1, 2]. The optimization algorithms can be roughly divided into two large groups: deterministic and non-deterministic methods. The deterministic methods search an approximate global minimum and guarantee its accuracy. Non-deterministic methods use local search techniques, heuristics or their combination to locate good approximations for the global minimum but have no means to estimate the accuracy of the obtained results. The main disadvantage of non-deterministic methods is the lack of certainty in the optimality of obtained solutions. Though for many problems the solution found by heuristic algorithms is satisfactory, there are plenty of areas where the knowledge of accuracy of the obtained minima is mandatory. In such fields heuristics can't replace deterministic methods.

Many deterministic methods were developed so far [3]. Methods for convex problems [4] rely on the following property: if the objective and constraints are convex functions and a local minimum exists then it is a global minimum. For such problems local optimization methods find global minima. In practice objective and constraints are often non-convex.

The most successful deterministic methods for global optimization are based on interval analysis [5, 6], convexification [7], and Lipschitzian approaches [8, 9]. Acceleration of Lipschitzian algorithms has been comprehensively studied in [10–12]. One of the first deterministic methods – the Non-uniform covering method was proposed in [13] and further elaborated in [14–16]. In this paper we strengthen the Non-uniform covering method by exploiting first and second order optimality conditions for box-constrained problems, *i.e.* mathematical programming problems with interval constraints on parameters and without functional constraints. The objective should be a twice differentiable function. Based on these conditions we developed an optimization algorithm, proved its correctness and studied its performance on test problems. Experimental results demonstrated that using first and second order optimality conditions can decrease the number of steps in an order of magnitude.

**Yury Evtushenko:** Dorodnicyn Computing Centre FRC CSC RAS Moscow, Russian Federation
***Corresponding Author: Mikhail Posypkin:** Dorodnicyn Computing Centre FRC CSC RAS Moscow, Russian Federation; Email: mposypkin@gmail.com
* Supported by Ministry of Science and Education of Republic of Kazakhstan, project 0115PK00554, Leading Scientific Schools project NSH-8860.2016.1

The paper is organized as follows. Section 2 introduces the basic theory. Section 3 outlines an optimization algorithm. Section 4 presents experimental results.

## 2 Theoretical background

A *box-constrained optimization problem* is formulated as follows

$$\begin{cases} f(x) \to \min, \\ x \in X, \end{cases} \quad (1)$$

where $a, b \in R^n$ are n-dimensional vectors and the *feasible set* $X = [a, b] = \{x \in R^n : a \le x \le b\}$ is an n-dimensional box. Any point from the feasible set is called a *feasible solution*. The goal is to find an *optimal solution*. An optimal solution is a point $x^* \in X$ such that $f\left(x^*\right) \le f(x)$ for all $x \in X$. For complex problems finding an exact optimal solution is usually impossible and optimization methods search approximate solutions.

A feasible solution $x^\varepsilon$ is called the *ε-solution* if $f(x^\varepsilon) \le f\left(x^*\right) + \varepsilon$. Non-uniform covering method [10–13] is able to find an *ε-solution* in a finite number of steps. Below we outline a basic idea of the method.

The Non-uniform covering method (NUC) generates the *trial sequence* $\mathrm{T} = \left\{x^{(1)}, \dots, x^{(k)}\right\}$ – a set of points where the objective function is evaluated. The key concept behind NUC is the concept of coverage. The coverage is a finite collection of sets $Cov = \bigcup_{i=1}^{m} C_i$, where $C_i$, $i = 1, \dots, m$ are subsets of $R^n$. Sets $C_i$ are called *covering sets*. The trial sequence and the coverage are constructed in a way to ensure the following *covering property*: $X \subseteq Cov$ implies that the point $x^R$ is an *ε-solution*, where $x^R = \arg\min f\left(x^{(i)}\right)$, $i = 1, \dots, k$.

In papers [10–13] several ways of coverage construction were proposed. All these approaches were based on the idea of bounding the objective function from below. The following theorem forms a theoretical basis for NUC method.

**Theorem 1.** *Let* $\mathrm{T} = \left\{x^{(1)}, \dots, x^{(k)}\right\}$ *be a trial sequence and* $Cov = \bigcup_{i=1}^{m} C_i$ *be such coverage that the covering property* $X \subseteq Cov$ *holds. Let for all* $i = 1, \dots, m$ *it holds*

$$\min_{x \in C_i} f(x) \ge f\left(x^{(l)}\right) - \varepsilon \ forsome\ l \in 1, \dots, k. \quad (2)$$

*Then* $x^R = \arg\min f\left(x^{(i)}\right)$, $i = 1, \dots, k$ *is an ε-solution.*

This obvious theorem is applied as follows. The method constructs the trial sequence and the coverage until the whole feasible set is totally covered. Then the minimum of the trial sequence becomes an *ε-solution*.

For objective functions with continuous first and second derivatives necessary optimality conditions can be used to construct additional covering sets. In the sequel we assume $f(\cdot)$ to be a differentiable function with continuous gradient and Hessian. Let's recall first and second order optimality conditions.

**Statement 1 (1st order optimality condition).** *If* $x^*$ *is an optimal solution of the problem (1) then the following properties hold for all* $i = 1, \dots, n$:

$$\text{if } a_i < x_i^* < b_i, \text{ then } \frac{\partial f(x^*)}{\partial x_i} = 0,$$

$$\text{if } x_i^* = b_i \text{ then } \frac{\partial f(x^*)}{\partial x_i} \le 0,$$

$$\text{if } x_i^* = a_i \text{ then } \frac{\partial f(x^*)}{\partial x_i} \ge 0.$$

**Statement 2 (2nd order optimality condition).** *Let* $x^*$ *be an optimal solution of the problem (1). Then for all* $i = 1, \dots, n$ *if* $a_i < x_i^* < b_i$ *then* $\frac{\partial^2 f(x^*)}{\partial x_i^2} \ge 0$.

Based on statements 1-2 we can formulate the following generalization of the Theorem 1.

**Theorem 2.** *Let* $\mathrm{T} = \left\{x^{(1)}, \dots, x^{(k)}\right\}$ *be a trial sequence and* $Cov = \bigcup_{i=1}^{m} C_i$ *be such coverage that the covering property* $X \subseteq Cov$ *holds. Let for all* $i = 1, \dots, m$ *the covering set* $C_i$ *satisfies one of the following properties:*

1. $\min_{x \in C_i} f(x) \ge f(x^{(l)}) - \varepsilon$ *for some* $l \in 1, \dots, k$,
2. $C_i \cap \left\{x \in X : x_j = a_j\right\} = \emptyset$ *and* $\min_{x \in C_i} \frac{\partial f(x)}{\partial x_j} > 0$ *for some* $j \in 1, \dots, n$,
3. $C_i \cap \left\{x \in X : x_j = b_j\right\} = \emptyset$ *and* $\max_{x \in C_i} \frac{\partial f(x)}{\partial x_j} < 0$ *for some* $j \in 1, \dots, n$,
4. $C_i \cap \left\{x \in X : x_j = a_j \text{ or } x_j = b_j\right\} = \emptyset$ *and* $\max_{x \in C_i} \frac{\partial^2 f(x)}{\partial x_j^2} < 0$ *for some* $j \in 1, \dots, n$.

*Then* $x^R = \arg\min f\left(x^{(i)}\right)$, $i = 1, \dots, k$ *is an ε-solution.*

**Proof.** Let $C_i$ be a covering set containing an optimal solution $x^*$. According to statements 1,2 none of the properties 2-4 holds for $C_i$. Therefore the property 1 holds, *i.e.* $\min_{x \in C_i} f(x) \ge f(x^{(l)})$ for some $l \in 1, \dots, k$. Since $x^* \in C_i$ we obtain $f(x^*) \ge f(x^{(l)}) - \varepsilon$. Thus $f(x^{(l)}) \le f(x^*) + \varepsilon$. By definition of $x^R$ we obtain $f(x^R) \le f(x^*) + \varepsilon$. □

# 3 Algorithm outline

To ensure the covering property one needs an efficient way to construct covering sets. According to Theorem 2, a covering set consists of feasible points satisfying one the following four inequalities:

$$f(x) \geq f(x^{(l)}) - \varepsilon, \quad \frac{\partial f(x)}{\partial x_j} < 0, \quad \frac{\partial f(x)}{\partial x_j} > 0, \quad \frac{\partial^2 f(x)}{\partial x_j^2} < 0.$$

For numerical stability reasons strict inequalities are substituted by non-strict inequalities:

$$f(x) \geq f(x^{(l)}) - \varepsilon, \quad \frac{\partial f(x)}{\partial x_j} \leq -\delta, \quad \frac{\partial f(x)}{\partial x_j} \geq \delta, \quad \frac{\partial^2 f(x)}{\partial x_j^2} \leq -\delta \tag{3}$$

where $\delta$ is a sufficiently small positive real number. If the numerical error in derivative calculation is less than $\delta$ this approach guarantees that the removed subsets do not contain feasible points.

Checking inequalities (3) directly is problematic for complex functions. The common way around is to consider more weak inequalities by substituting function by its under- or overestimations.

**Definition.** *A function $\mu(\cdot) : R^n \to R$ is called an underestimation for a function $f(\cdot)$ over a set $Z \subseteq R^n$ if $f(x) \geq \mu(x)$ for all $x \in Z$. A function $\nu(\cdot) : R^n \to R$ is called an overestimation for a function $f(\cdot)$ over a set $Z \subseteq R^n$ if $f(x) \leq \nu(x)$ for all $x \in Z$.*

Thus the inequality $f(x) \geq a$ is substituted by a weaker inequality $\mu(x) \geq a$and the inequality $f(x) \leq a$ by the inequality $\nu(x) \leq a$. Estimations are usually selected to ensure an easy resolution of such inequalities. Examples of such underestimations are interval bounds [3, 5, 6] or Lipschitzian bounds [8–13].

Assume that we are able to construct an underestimation $\phi(x)$ for the objective function $f(x)$, overestimations $\nu_i(x)$ and underestimations $\mu_i(x)$ for i-th component $\frac{\partial f(x)}{\partial x_i}$ of the gradient and overestimations $\xi_i(x)$ for i-th diagonal component $\frac{\partial^2 f(x)}{\partial x_i \partial x_i}$ of the Hessian. We also assume that minimum and maximum of these estimates on a box can be analytically found.

Let $Z = [c, d]$ be a sub-box of a box $X$. Define reduction rules for the box $Z$ based on the introduced estimates.

**Reduction rule 1.** If $\min_{x \in Z} \phi(x) \geq f(x^R) - \varepsilon$ the box $Z$ is discarded.

**Reduction rule 2.1.** Let $\min_{x \in Z} \mu_i(x) \geq \delta$. If $c_i = a_i$ the box $Z$is reduced to the box $[c_1, d_1] \times \ldots \times [c_i, c_i] \times \ldots \times [c_n, d_n]$ by eliminating its i-th dimension. Otherwise the box $Z$is discarded.

**Reduction rule 2.2.** Let$\min_{x \in Z} \nu_i(x) \leq -\delta$. If $d_i = b_i$ the box $Z$is reduced to the box $[c_1, d_1] \times \ldots \times [d_i, d_i] \times \ldots \times [c_n, d_n]$ by eliminating its i-th dimension. Otherwise the box $Z$is discarded.

**Reduction rule 3.1.** Let $\max_{x \in Z} \xi_i(x) \leq -\delta$. If $c_i = a_i, \ d_i = b_i$ the box $Z$ is substituted by a union of two boxes $[c_1, d_1] \times \ldots \times [c_i, c_i] \times \ldots \times [c_n, d_n]$ and $[c_1, d_1] \times \ldots \times [d_i, d_i] \times \ldots \times [c_n, d_n]$.

**Reduction rule 3.2.** Let $\max_{x \in Z} \xi_i(x) \leq -\delta$. If $c_i = a_i, \ d_i \neq b_i$ the box $Z$ is reduced to the box $[c_1, d_1] \times \ldots \times [c_i, c_i] \times \ldots \times [c_n, d_n]$.

**Reduction rule 3.3.** Let $\max_{x \in Z} \xi_i(x) \leq -\delta$. If $c_i \neq a_i, \ d_i = b_i$ the box $Z$ is reduced to the box $[c_1, d_1] \times \ldots \times [d_i, d_i] \times \ldots \times [c_n, d_n]$.

**Reduction rule 3.4.** Let $\max_{x \in Z} \xi_i(x) \leq -\delta$. If $c_i \neq a_i, \ d_i \neq b_i$ the box $Z$ is discarded.

The following statement is obvious.

**Statement 3.** *The set of points eliminated by any of the rules 1-3.4 satisfies one of the properties 1-4 from the Theorem 2.*

The proposed algorithm NUC (Non-uniform covering method) follows the standard Branch-and-Bound approach. It maintains a list of boxes $\Lambda$ throughout the search. On each step a box $Z$ is extracted from a list $\Lambda$ (step 4), then the incumbent solution is updated (step 5) and reduction rules are applied to the box $Z$thereby obtaining a new (probably empty) list of sub-boxes $\Lambda'$ (step 6). If the list $\Lambda'$ contains exactly one box, it is partitioned into two new boxes which substitute this box in the list $\Lambda'$ (step 7). Partitioning is done by splitting a box across its longest edge. This approach ensures a sufficient reduction of the box at each iteration. Then the list $\Lambda'$ is appended to the list $\Lambda$: $\Lambda = \Lambda \cup \Lambda'$ at step 8. The procedure is repeated until the list $\Lambda$contains at least one element (step 3). The algorithm evaluates the objective in a sequence of trial points $T$ and at each step compute the incumbent solution $x^R = \arg\min f(x), \ x \in T$.

**Algorithm NUC**

1. Initialize the list $\Lambda$with a box $X$: $\Lambda = \{X\}$.
2. Initialize the trial sequence by an empty set: $T = \emptyset$.
3. If $\Lambda = \emptyset$ then terminate the algorithm.
4. Extract a box $Z$ from the list $\Lambda$.
5. Compute function value in the center $z$ of the box $Z$ and update the trial sequence with this point, and compute the incumbent solution $x^R = \arg\min f(x)$, $x \in T$.
6. Apply reduction rules to the box $Z$, obtain a list of sub-boxes $\Lambda'$.
7. If $\Lambda'$contains only one box $B$ then partition it into two equal boxes $B_1$and $B_2$and assign $\Lambda' = B_1 \cup B_2$
8. Update the list $\Lambda$: $\Lambda = \Lambda \cup \Lambda'$ and go to the step 3.

**Theorem 3.** *The incumbent solution $x^R$obtained by the NUC algorithm is an an $\varepsilon$-solution of the problem (1).*

**Proof.** According to statement 3, all subsets eliminated by this algorithm satisfy the covering property. When the algorithm terminates, the list $\Lambda$is empty which means that the union of the discarded sets includes the feasible set $X$. Thus the covering property $X \subseteq Cov$ holds. According to the Theorem 2 the incumbent value $x^R = \arg\min f\left(x^{(i)}\right)$, $i = 1, \dots, k$ is an $\varepsilon$-*solution*. □

# 4 Implementation and experimental evaluation

The proposed algorithm was implemented as an open-source C++ library [17]. The modular design enables straightforward addition of new reduction rules. Reduction rules are represented as C++ classes inherited from the base class CutFactory. There are three separate factories implementing reduction rules 1, 2.1-2.2 and 3.1-3.4 respectively. The class CompositeClassFactory combines several factories by means of addFactory method. It provides a convenient way for introducing new factories to the search process.

Three test functions were selected for evaluating the proposed approach. The bounding cubic box $X = [-A, A] \times \dots \times [-A, A]$ was the same for all three problems with $A = 10$. The first test function is a classical global optimization test [18].

**Test function 1 (Zirilli or Alluffi-Pentini function)**

$$f(x) = 0.25\, x_1^4 - 0.5\, x_1^2 + 0.1\, x_1 + 0.5\, x_2^2$$

The global minimum is located at $x^* = (-1.0465, 0)$, $f(x^*) = -0.3523$.

Second and third test functions are specially designed to test the ability of finding minima located on a feasible set boundary.

**Test function 2 (Saddle function)**

$$f(x) = x_1^2 - x_2^2$$

The global minimum is located at points$x^* \in \{(0, A), (0, -A)\}$, $f(x^*) = -A^2$.

**Test function 3 (Cubic function)**

$$f(x) = \sum_{i=1}^{n} x_i^3$$

The global minimum is located at the point $x^* = (-A, -A, \dots, -A)$, $f(x^*) = -n\,A^3$.

The first two test functions have two variables while the third test function may have an arbitrary number of parameters. We used 2, 4, 8 and 16 parameters for testing (lines Cubic 2, Cubic 4, Cubic 8, Cubic 16 in Table 1). In all cases $\varepsilon = 10^{-4}$.

Three search strategies with different choice of reduction rules were evaluated and compared. The first strategy used reduction rule 1, second used reduction rules 1 and 2.1-2.2, third one used all reduction rules. Interval lower bounds for objective functions and Lipschitzian lower bounds for gradients and Hessians were applied. The Lipschitz constants were overestimated with techniques proposed in [15]. The strategies were compared by the number of iterations of 3-9 loop of the NUC method. Table 1 summarizes the results.

**Table 1:** Comparison of different search strategies.

| Test function | Number of steps | | |
|---|---|---|---|
| | Rule 1 | Rules 1, 2.* | Rules 1, 2.*, 3.* |
| Zirilli | 118075 | 319 | 319 |
| Saddle | 85 | 25 | 3 |
| Cubic 2 | 105 | 13 | 9 |
| Cubic 4 | 217 | 25 | 13 |
| Cubic 8 | 449 | 51 | 21 |
| Cubic 16 | 927 | 113 | 37 |

The first observation that can be made from the analysis of the Table 1 is that the efficiency of applying different reduction rules depend on a problem. For Zirilli problem gradient-based reductions (rule 2.*) gives 3-order of magnitude speedup. However Hessian-based reduction (rule 3.*) doesn't further decrease the number of steps. For Saddle and Cubic functions Hessian-based reductions have a remarkable effect. This can be explained by large regions of objective function's concavity efficiently eliminated by reduction rules 3.1-3.4 for problems 2 and 3. For Saddle function it decreases the number of steps in two times and for Cubic function – in 1.5-3 times and the effect grows with the problem dimension.

## 5 Conclusions

The paper studied an approach based on exploiting first and second order optimality conditions in Non-uniform covering method for box-constrained problems. Additional reduction rules based on optimality conditions were introduced. The correctness of the introduced rules was proven and the efficiency was experimentally evaluated. Experiments demonstrated a significant speedup due to optimality conditions.

In the future we are going to further elaborate the proposed techniques by combining these rules and efficient box-reduction techniques proposed in [15]. We also plan to apply the proposed approach to challenging global optimization problems arising in material science and use parallelization to increase the performance of the algorithm [19, 20].

## References

[1] Pardalos P., Romeijn E., Tuy H., Recent developments and trends in global optimization, J. of Computational and Applied Mathematics, 2000, 124 (1-2), 209–228

[2] Pardalos P.M., Resende M., (Eds), Handbook of Applied Optimization, Oxford University Press, 2002

[3] Scholz D., Deterministic global optimization: geometric branch-and-bound methods and their applications, Springer Science & Business Media, 2011.

[4] Nesterov Y., Introductory Lectures on Convex Optimization: A Basic Course (Applied Optimization), Springer Netherlands, 2003

[5] Kearfott R.B., Rigorous Global Search: Continuous Problems, Kluwer, Dordrecht, 1996

[6] Hansen E., Global Optimization Using Interval Analysis, Dekker, New York, 1992

[7] Tawarmalani M., Sahinidis N.V., Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications, Kluwer, Dordrecht, 2002

[8] Strongin R.G., Sergeyev Ya.D., Global optimization with non-convex constraints: Sequential and parallel algorithms, Kluwer Academic Publishers, Dordrecht, 2000

[9] Pinter J.D., Global Optimization in Action, Kluwer, Dordrecht, 1996

[10] Molinaro A., Pizzuti C., Sergeyev Ya.D., Acceleration tools for diagonal information global optimization algorithms, Computational Optimization and Applications, 2001, 18(1), 5–26

[11] Lera D., Sergeyev Ya.D., Acceleration of univariate global optimization algorithms working with Lipschitz functions and Lipschitz first derivatives, SIAM Journal on Optimization, 2013, 23(1), 508–529

[12] Kvasov D.E., Sergeyev Ya.D., Lipschitz gradients for global optimization in a one-point-based partitioning scheme, Journal of Computational and Applied Mathematics, 2012, 236(16), 4042–4054

[13] Evtushenko Yu., Numerical methods for finding global extreme (case of a non-uniform mesh), U.S.S.R. Comput. Maths. Math. Phys., Moscow, 1971, 11 (6), 38–54

[14] Evtushenko Yu., Potapov M., Deterministic global optimization, In Spedicato, E. (eds.) Algorithms for Continuous Optimization. The State of the Art, Kluwer, 1987, 481–500

[15] Evtushenko Y., Posypkin M., A deterministic approach to global box-constrained optimization, Optimization Letters, 2013, 7 (4), 819–829

[16] Evtushenko Y.G., Posypkin M.A., A deterministic algorithm for global multi-objective optimization, Optimization Methods and Software, 2014, 29 (5), 1005–1019

[17] Posypkin M.A., Non-uniform covering method library. https://github.com/mposypkin/NUC (viewed April 15, 2016)

[18] Jamil M., Yang X.S., A literature survey of benchmark functions for global optimisation problems, International Journal of Mathematical Modelling and Numerical Optimisation, 2013, 4. (2), 150–194

[19] Evtushenko Y.G., Malkova V.U., Stanevichyus A.A., Parallel global optimization of functions of several variables, Computational Mathematics and Mathematical Physics, 2009, 49 (2), 246–260

[20] Evtushenko Y., Posypkin M., Sigal I., A framework for parallel large-scale global optimization, Computer Science-Research and Development, 2009, 23(3-4), 211–215